Ahmednagar Jilha Maratha Vidya Prasarak Samaj's

New Arts, Commerce and Science College, Ahmednagar (Autonomous)

(Affiliated to Savitribai Phule Pune University, Pune)

Choice Based Credit System (CBCS)

Bachelor of Business Administration (Computer Application)

B.B.A.(C.A.)

Syllabus of

T.Y.B.B.A.(C.A.)

SEMESTER IV AND V

Implemented from

Academic Year 2023 - 24

Ahmednagar Jilha Maratha Vidya Prasarak Samaj's

New Arts, Commerce and Science College, Ahmednagar (Autonomous)

Board of Studies in BBA(CA)

Sr. No.	Name	Designation
1.	Mrs. Nimbalkar Sangita S.	Chairman
2.	Mr. Talule Sonyabapu S.	Member
3.	Mr. Gobare Manohar B.	Member
4.	Miss. Danave Bharati M.	Member
5.	Mr. Pachpande Suhas D.	Academic Council Nominee
6.	Dr. Patil Chandrashekhar Himmatrao	Academic Council Nominee
7.	Prof. (Mrs.) Siddavatam A. Shakilabanu	Vice Chancellor Nominee
8.	Mrs. Mohite-Patil Amruta Rahul	Alumni
9.	Mr. Dawbhat Arun Rangnath	Industry Expert
10.	Mrs. Kulkarni Aparna A.	Member(co-opt)

3. Programme Structure and Course Titles: (All academic years)

Sr. No.	Class	Semester	Course Code	Course Title	Credits
1.	FY BBA(CA)	I	BBACA-101 T	FUNDAMENTALS OF INFORMATION TECHNOLOGY	04
2.	FY BBA(CA)	I	BBACA-102 T	‘C’ PROGRAMMING	04
3.	FY BBA(CA)	I	BBACA-103 T	FINANCIAL ACCOUNTING	04
4.	FY BBA(CA)	I	BBACA-104 T	BUSINESS STATISTICS	04
5.	FY BBA(CA)	I	BBACA-105 P	PRACTICAL- (BASED ON BBACA-101 T)	02
6.	FY BBA(CA)	I	BBACA-106 P	PRACTICAL –(BASED ON BBACA-102 T)	02
7	FY BBA(CA)	I	BBACA-107 T(A)	PRINCIPLES OF MANAGEMENT	04
8	FY BBA(CA)	I	BBACA-107 T(B)	PRINCIPLES OF MARKETING	04
9	FY BBA(CA)	II	BBACA-201 T	DATABASE MANAGEMENT SYSTEM	04
10	FY BBA(CA)	II	BBACA-202 T	WEB TECHNOLOGY(HTML,CSS,JS)	04
11	FY BBA(CA)	II	BBACA-203 T	BUSINESS MATHEMATICS	04
12	FY BBA(CA)	II	BBACA-204 T	ORGANIZATIONAL BEHAVIOUR & HUMAN RESOURCE MANAGEMENT	04
13	FY BBA(CA)	II	BBACA-205 P	PRACTICAL (BASED ON - BBACA-201 T)	02
14	FY BBA(CA)	II	BBACA-206 P	PRACTICAL (BASED ON - BBACA-202 T)	02
15	FY BBA(CA)	II	BBACA-207 T(A)	DIGITAL MARKETING CONCEPTS	04
16	FY BBA(CA)	II	BBACA-207 T(B)	E-COMMERCE CONCEPTS	04
17	SY BBA(CA)	III	BBACA-301 T	RELATIONAL DATABASE MANAGEMENT SYSTEM	04
18	SY BBA(CA)	III	BBACA-302 T	DATA STRUCTURE USING C	04
19	SY BBA(CA)	III	BBACA-303 T	WEB DEVELOPMENT WITH PHP	04
20	SY BBA(CA)	III	BBACA-304 P	PRACTICAL (BASED ON - BBACA-301 T)	02

21	SY BBA(CA)	III	BBACA-305 P	PRACTICAL (BASED ON - BBACA-302 T & BBACA-303 T)	02
22	SY BBA(CA)	III	BBACA-306 T	SOFTWARE ENGINEERING	02
23	SY BBA(CA)	III	BBACA-307 T	BUSINESS COMMUNICATION	02
24	SY BBA(CA)	III	BBACA-308 T	KNOWLEDGE MANAGEMENT	02
25	SY BBA(CA)	IV	BBACA-401 T	DATA COMMUNICATION AND COMPUTER NETWORKING	04
26	SY BBA(CA)	IV	BBACA-402 T	OBJECT ORIENTED CONCEPTS THROUGH C++	04
27	SY BBA(CA)	IV	BBACA-403 T	ADVANCED WEB DEVELOPMENT	04
28	SY BBA(CA)	IV	BBACA-404 P	PRACTICAL – BASED ON BBACA-402 T	02
29	SY BBA(CA)	IV	BBACA-405 P	PRACTICAL – BASED ON BBACA-403 T	02
30	SY BBA(CA)	IV	BBACA-406 T	OPERATING SYSTEM CONCEPTS	02
31	SY BBA(CA)	IV	BBACA-407 T	ENVIRONMENTAL STUDIES	02
32	SY BBA(CA)	IV	BBACA-408 T	BLOGGING TOOLS	02
33	TY BBA(CA)	V	BBACA-501 T	PROGRAMMING IN CORE JAVA	04
34	TY BBA(CA)	V	BBACA-502 P	PRACTICAL (BASED ON - BBACA-501 T)	02
35	TY BBA(CA)	V	BBACA-503 T(A)	BIG DATA	04
36	TY BBA(CA)	V	BBACA-503 T(B)	BLOCK CHAIN	04
37	TY BBA(CA)	V	BBACA-504 T(A)	PYTHON	04
38	TY BBA(CA)	V	BBACA-504 T(B)	NO-SQL	04
39	TY BBA(CA)	V	BBACA-505 P	PRACTICAL (BASED ON - BBACA-503 & BBACA-504)	02
40	TY BBA(CA)	V	BBACA-506 T	OBJECT ORIENTED SOFTWARE ENGINEERING	04
41	TY BBA(CA)	V	BBACA-507 T	CYBER SECURITY	02
42	TY BBA(CA)	V	BBACA-508 PR	SOFTWARE PROJECT	02
43	TY BBA(CA)	VI	BBACA-601 T	ADVANCED JAVA	04
44	TY BBA(CA)	VI	BBACA-602 P	PRACTICAL (BASED ON - BBACA-601 T)	02
45	TY BBA(CA)	VI	BBACA-603 T(A)	SOFTWARE TESTING	04
46	TY BBA(CA)	VI	BBACA-603 T(B)	ARTIFICIAL INTELLIGENCE CONCEPTS	04
47	TY BBA(CA)	VI	BBACA-604 T(A)	DOT NET PROGRAMMING	04
48	TY BBA(CA)	VI	BBACA-604 T(B)	ANDROID PROGRAMMING	04
49	TY BBA(CA)	VI	BBACA-605 P	PRACTICAL (BASED ON - BBACA-603 T & BBACA-604 T)	02

50	TY BBA(CA)	VI	BBACA-606 T	RECENT TRENDS IN INFORMATION TECHNOLOGY	04
51	TY BBA(CA)	VI	BBACA-607 T	INTERPERSONEL SKILLS AND PROFESSIONAL ETHICS	02
52	TY BBA(CA)	VI	BBACA-608 PR	SOFTWARE PROJECT	02

Semester -V	**Paper -I**
Course Code: BBACA-501 T	**Core Java**
Credits: 04	**Total Lectures: 60**

Prerequisite:
- Student should know basics of object oriented programming.

Course Objectives:
- To introduce the object oriented programming concepts.
- To understand object oriented programming concepts, and apply them in solving problems.
- To introduce the principles of inheritance and polymorphism; and demonstrate how they relate to the design of abstract classes
- To introduce the implementation of packages and interfaces
- To introduce the concepts of exception handling and multithreading.
- To introduce the design of Graphical User Interface using applets and swing controls.

Course Outcomes:
- Able to solve real world problems using OOP techniques.
- Able to understand the use of abstract classes.
- Able to solve problems using java collection framework and I/o classes.
- Able to develop multithreaded applications with synchronization.
- Able to develop applets for web applications.
- Able to design GUI based applications

UNT 1. Java Fundamentals **12**

1.1 Introduction to Java.

1. Features of Java
2. Basics of Java: - Data types, variable, expression, operators, constant.
3. Structure of Java Program.
4. Execution Process of java Program.
5. JDK Tools.
6. Command Line Arguments.
7. Array and String:
   1. Single Array & Multidimensional Array
   2. String, String Buffer
8. Built In Packages and Classes :
   1. java.util:- Scanner, Date, Math etc.

2. java.lang

UNIT 2. Classes, Objects and Methods 12

1. Class and Object
2. Object reference
3. Constructor: Constructor Overloading
4. Method: Method Overloading, Recursion, Passing and

   Returning object form Method

5. new operator, this and static keyword, finalize() method
6. Nested class, Inner class, and Anonymous inner class

UNIT 3. Inheritance, Package and Collection 12

1. Overview of Inheritance
2. inheritance in constructor
3. Inheriting Data members and Methods,
4. Multilevel Inheritance – method overriding Handle multilevel constructors
5. Use of super and final keyword
6. Interface:
7. Creation and Implementation of an interface, Interface reference
8. Interface inheritance
9. Dynamic method dispatch
10. Abstract class
11. Comparison between Abstract Class and interface
12. Access control

13. Packages

1. Packages Concept
2. Creating user defined packages
3. Java Built inpackages
4. Import statement, Static import

13. Collection

1. CollectionFramework.
2. Interfaces: Collection, List, Set
3. Navigation: Enumeration, Iterator, ListIterator
4. Classes: LinkedList, ArrayList, Vector, HashSet

UNIT 4. File and Exception Handling 12

Exception

1. Exception and Error
2. Use of try, catch, throw, throws and finally
3. Built in Exception
4. Custom exception
5. Throwable Class.

File Handling

6. Overview of Different Stream (Byte Stream, Character stream)
7. Readers and Writers class
8. File Class
9. File Input Stream , File Output Stream

- Input Stream Reader and Output Stream Writer class

- FileReader and FileWriter class

10. Buffered Reader class.

UNIT 5. Applet, AWT, Event and Swing **12**

Applet Programming

1. Introduction
2. Typesapplet
3. Applet Lifecycle
1. Creatingapplet
2. Applet tag

Reference Books:

1. Programming with JAVA - EBalgurusamy
2. The Complete Reference – JAVA HerbertSchildt
3. Programming in Java, S. Malhotra, S. Chudhary, 2nd edition, Oxford Univ. Press.
4. Java Programming and Object-oriented Application Development, R. A. Johnson, Ceng

Semester -V	**Paper -II**
Course Code: BBACA-502 P	**PRACTICALS(BASED ON BBACA-501-T)**
Credits: 02	**Total Practicals: 30**

Ass. No.	**Week**	**Assignment**
1.	First	1. Write a java Program to check whether a given String is Palindrome or not. 2. Write a Java program which accepts three integer values and prints the maximum and minimum. 3. Write a Java program to accept a number from the command prompt and generate a multiplication table of a number. 4. Write a Java program to display Fibonacci series. 5. Write a Java program to calculate the sum of digits of a number. 6. Write a Java program to accept a year and check if it is a leap year or not. 7. Write a Java program to display characters from A to Z using a loop. 8. Write a Java program to accept two numbers using command line argument and calculate addition, subtraction, multiplication and division. 9. Write a java Program to calculate the sum of the first and last digit of a number. 10. Write a java program to calculate the sum of even numbers from an array.
2.	Second	1. Write a java Program to check whether a given number is Prime or Not. 2. Write a java Program to display all the perfect numbers between 1 to n.

		3. Write a java Program to accept employee names from a user and display it in reverse order. 4. Write a java program to display all the even numbers from an array. (Use Command Line arguments) 5. Write a java program to display the vowels from a given string.
3.	Third	1. Write a java program to accept n city names and display them in ascending order. 2. Write a java program to accept n numbers from a user store only Armstrong numbers in an array and display it. 3. Write a java program to search given name into the array, if it is found then display its index otherwise display appropriate message. 4. Write a java program to display following pattern: 5 4 5 3 4 5 2 3 4 5 1 2 3 4 5
4.	Fourth	1. Write a java program to count the frequency of each character in a given string. 2. Write a java program to display each word in reverse order from a string array. 3. Write a java program for union of two integer arrays. 4. Write a java program to display the transpose of a given matrix. 5. Write a java program to display alternate characters from a given string.
5.	Fifth	1. Write a Java program for the implementation of reference variables. 2. Write a Java program to keep the count of objects created of a class. Display the count each time when the object is created. 3. Write a Java program to convert integer primitive data. (use toString()). 4. Write a Java program to calculate the sum of digits of a number using Recursion. 5. Write a Java program for the implementation of this keyword.
6.	Sixth	1. Write a Java program to calculate power of a number using recursion. 2. Write a Java program to display Fibonacci series using functions. 3. Write a Java program to calculate the area of Circle, Triangle & Rectangle.(Use Method Overloading) 4. Write a Java program to Copy data of one object to another Object. 5. Write a Java program to calculate factorial of a number using recursion.
7.	Seventh	1. Define a class person(pid,pname,age,gender). Define Default and parameterised constructor. Overload the constructor. Accept the 5 person details and display it.(use this keyword). 2. Define a class product(pid,pname,price). Write a function to accept the product details, to display product details and to calculate total amount. (use array of Objects) 3. Define a class Student(rollno,name,per). Create n objects of the student class and Display it using toString().(Use parameterized constructor) 4. Define a class MyNumber having one private integer data member. Write a default constructor to initialize it to 0 and another constructor to initialize it to a value. Write methods isNegative, isPositive. Use command line argument to pass a value to the object and perform the above tests.
8.	Eighth	1. Define class Student(rno, name, mark1, mark2). Define Result class(total, percentage) inside the student class. Accept the student details & display the mark sheet with rno, name, mark1, mark2, total, percentage. (Use inner class concept)

		2. Write a java program to accept n employee names from users. Sort them in ascending order and Display them.(Use array of object and Static keyword) 3. Write a java program to accept details of ‘n’ cricket players(pid, pname, totalRuns, InningsPlayed, NotOuttimes). Calculate the average of all the players. Display the details of players having maximum average. 4. Write a java program to accept details of ‘n’ books. And Display the quantity of the given book.
9.	Ninth	1. Create abstract class Shape with abstract method area().Write a Java program to calculate area of Rectangle and Triangle.(Inherit Shape class in classes Rectangle and Triangle) 2. Create a class Teacher(Tid, Tname, Designation, Salary, Subject). Write a Java program to accept the details of ‘n’ teachers and display the details of teacher who is teaching Java Subject.(Use array of Object) 3. Create a class Doctor(Dno, Dname, Qualification, Specialization). Write a Java program to accept the details of ‘n’ doctors and display the details of doctors in ascending order by doctor name. 4. Write a Java program to accept ‘n’ employee names through command line, Store them in vector. Display the name of employees starting with the character ‘S’. 5. Create a package Mathematics with two classes Maximum and Power. Write a java program to accept two numbers from the user and perform the following operations on it: a. Find Maximum of two numbers. b. Calculate the power(XY);
10.	Tenth	1. Write a java program to calculate area of Cylinder and Circle.(Use super keyword) 2. Define an Interface Shape with abstract method area(). Write a java program to calculate an area of Circle and Sphere.(use final keyword) 3. Define an Interface “Integer” with an abstract method check().Write a Java program to check whether a given number is Positive or Negative. 4. Define a class Student with attributes rollno and name. Define default and parameterized constructor. Override the toString() method. Keep the count of Objects created. Create objects using parameterized constructor and Display the object count after each object is created. 5. Write a java program to accept ‘n’ integers from the user & store them in an ArrayList collection. Display the elements of the ArrayList collection in reverse order.
11.	Eleven	1. Create an abstract class Shape with methods calc_area() & calc_volume(). Derive two classes Sphere(radius)& Cone(radius, height) from it. Calculate area and volume of both. (Use Method Overriding) 2. Define a class Employee having private members-id, name, department, salary. Define default & parameterized constructors. Create a subclass called Manager with a private member bonus. Define methods accept & display in both the classes. Create n objects of the manager class & display the details of the manager having the maximum total salary(salary+bonus). 3. Construct a Linked List contains names: CPP, Java, Python and PHP. Then extend your program to do the following: i. Display the contents of the List using an iterator ii. Display the contents of the List in reverse order using a ListIterator. 4. Create a hashtable containing employee name & salary. Display the details of the hashtable. Also search for a specific Employee and display the salary of that employee. 5. Write a package game which will have 2 classes Indoor & Outdoor. Use a function display() to generate the list of players for the specific game. Use default & parameterized constructor.

12.	Twelve	1. Write a java program to accept the data from a user and write it into the file. 2. Write a java program to display ASCII values of the characters from a file. 3. Write a java program to count the number of digits, spaces and characters from a file. 4. Write a java program to accept a number from a user if it is non-zero then check whether it is Armstrong or not, otherwise throws user-defined Exception “Number is Invalid”. 5. Write a java program to check whether a given file is hidden or not. 6. Write a java program to display the name and size of the given files.
13.	Thirteen	1. Write a java program to count the number of integers from a given list.(Use command line arguments). 2. Write a java program to check whether a given candidate is eligible for voting or not. Handle user defined as well as system defined Exception. 3. Write a java program to calculate the size of a file. 4. Write a java program to accept a number from a user, if it is zero then throw user defined Exception “Number is Zero”. If it is non-numeric then generate an error “Number is Invalid” otherwise check whether it is palindrome or not. 5. Write a java program to accept a number from a user, If it is greater than 100 then throw a user defined exception “Number is out of Range” otherwise do the addition of digits of that number. (Use static keyword)
14.	Fourteen	1. Write a java program to copy the data from one file into another file, while copying change the case of characters in target file and replaces all digits by ‘*’ symbol. 2. Write a java program to accept strings from a user. Write ASCII values of the characters from a string into the file. 3. Write a java program to accept a number from a user, if it is less than 5 then throw user defined Exception “Number is small”, if it is greater than 10 then throw user defined exception “Number is Greater”, otherwise calculate its factorial. 4. Write a java program to display contents of a file in reverse order. 5. Write a java program to display each word from a file in reverse order.
15	Fifteen	1. Develop an applet that draws a circle. The dimension of the applet should be 500 * 300 pixels the circle should be centered on the applet and have a radius of 100 pixels. Display your name centered in a circle(using drawOval() method) 2. Write a program to create a frame using AWT. Implement mouseClicked(), mouseEntered() and mouseExited() events. Frame should become visible when the mouse enters it. 3. Write a program to display a string in a frame window with pink color as background. 4. Write a program to create two buttons named “Red” and “Blue”. When a button is pressed the background color should be set to the color named by the button’s label. 5. Write a program which responds to the KEY_TYPED event and updates the status window with a message (“Typed character is: X”). Use adapter class for other two events. 6. Write a program to create two buttons labeled ‘GetInfo’ and ‘GetCGPA’. When button ‘GetInfo’ is pressed, it displays your personal information (Name, Course, Roll No, College) and when button ‘GetCGPA’ is pressed, it displays your CGPA in previous semester.
16	Sixteen	1.Write a java program to implement a simple arithmetic calculator. Perform appropriate validations. 2. Write an applet application to draw Temple. 3. Write an applet application to display Table lamps. The color of the lamp should change in random color.

		4. Write a java program to design email registration form.(Use maximum Swing component in form).
17	Seventeen	1. Write a java program to accept the details of employee eno,ename, sal and display it on the next frame using appropriate events . 2. Write a java program to display at least five records of employees in JTable.(Eno, Ename ,Sal). 3. Write a java Program to change the color of the frame. If the user clicks on the close button then the position of the frame should change.

Semester -V	**Paper -III**
Course Code: BBACA-503 T (A)	**Big Data**
Credits: 04	**Total Lectures: 60 Hrs.**

Objectives:

1. Understand the Big Data Platform and its Use cases
2. Provide an overview of Apache Hadoop
3. Provide HDFS Concepts and Interfacing with HDFS ,Understand Map Reduce Jobs
4. Provide hands on Hadoop EcoSystem ,Apply analytics on Structured, Unstructured Data.
5. Exposure to Data Analytics with R.

COURSE OUTCOMES:

The students will be able to:

• Identify Big Data and its Business Implications.
• List the components of Hadoop and Hadoop Ecosystem
• Access and Process Data on Distributed File System
• Manage Job Execution in Hadoop Environment

• Develop Big Data Solutions using Hadoop EcoSystem
• Apply Machine Learning Techniques using R.

1.	**INTRODUCTION TO BIG DATA**	**18**
	1.1 Introduction to Big Data and its characteristics	
	1.2 Types of Digital Data	
	1.3 Big Data Analytics	
	1.4 Application of Big data	
2.	**Introduction to Big Data Platforms**	**12**
	2.1Introduction to Big Data Storage Platforms for Large Scale Data Storage	
	2.2. Introduction to Big Data Streaming Platforms for Fast Data	
3.	**INTRODUCTION TO DATA SCIENCE**	**16**
	3.1 Basics of Data Analytics	
	3.2 Types of Analytics –	
	3.2.1 Descriptive,	
	3.2.2 Predictive,	
	3.2.3 Prescriptive	
	3.2.4 Statistical Inference	
	3.3 Populations and samples	
	3.3.1 Statistical modeling,	
	3.3.2 Probability	
	3.3.3 Distribution	
	3.3.4 Correlation	
	3.3.5 Regression	
4.	**INTRODUCTION TO MACHINE LEARNING**	**14**
	4.1 Basics of Machine Learning	
	4.2 Supervised Machine Learning	
	4.2.1 K- Nearest-Neighbors,	
	4.2.2 Naïve Bayes	
	4.2.3 Decision tree	
	4.2.4 Support Vector Machines	
	4.3 Unsupervised Machine Learning	
	4.3.1 Cluster analysis	
	4.3.2 K means	
	4.3.3 EM Algorithm	

Suggested Readings:

1. Seema Acharya, Subhasini Chellappan, "Big Data Analytics" Wiley 2015.

2. Jay Liebowitz, “Big Data and Business Analytics” Auerbach Publications, CRC press (2013)

3. ArvindSathi, “BigDataAnalytics: Disruptive Technologies for Changing the Game”, MC Press, 2012.

Semester -V	**Paper -III**
Course Code: BBACA-503 T (B)	**Block Chain**
Credits: 04	**Total Lectures: 60 Hrs.**

PREREQUISITES:

This course is highly technical in nature and would require the student to be comfortable with coding. To prepare for the class all students MUST:

- Understanding of basic programming languages like Java, or Javascript.
- Understanding of PKI and Docker.

WHAT YOU'LL LEARN

- Understand what and why of Blockchain
- Explore the major components of Blockchain
- Learn about Bitcoin, Cryptocurrency, Ethereum
- Deploy and exercise example smart contracts
- Identify a use case for a Blockchain application
- Create your own Blockchain network application

COURSE OBJECTIVES

By the end of the course, students will be able to

1. Understand how blockchain systems (mainly Bitcoin and Ethereum) work,
2. To securely interact with them,
3. Design, build, and deploy smart contracts and distributed applications,
4. Integrate ideas from blockchain technology into their own projects.

UNIT 1 Introduction To Blockchain **14**

1.1 Digital Trust
1.2 Asset
1.3 Transactions
1.4 Distributed Ledger Technology
1.5 Types of network
1.6 Components of blockchain or DLT
1.7 Ledger
    1.7.1. Blocks
    1.7.2. Blockchain
1.8 PKI and Cryptography
    1.8.1. Private keys
    1.8.2. Public keys
    1.8.3. Hashing
    1.8.4. Digital Signature
1.9. Consensus
    1.9.1. Byzantine Fault
    1.9.2. Proof of Work
    1.9.3. Poof of Stake
1.10. Security
    1.10.1.DDos
    1.11 Cryptocurrency
    1.12.Digital Token

Unit 2. How Blockchain Works **14**

2.1 How Blockchain Works
2.2. Structure of Blockchain
2.3.Block
2.4. Hash
2.5. Blockchain
2.6. Distributed
2.7. Lifecycle of Blockchain

2.8. Smart Contract
2.9. Consensus Algorithm
2.10 Proof of Work
2.11 Proof of Stake
2.12 Practical Byzantine
2.13 Fault Tolerance
2.14 Actors of Blockchain
2.15 Blockchain developer
2.16 Blockchain operator
2.17 Blockchain regulator
2.18 Blockchain user
2.19 Membership service provider
2.20 Building A Small Blockchain Application

Unit 3. Introduction to Bitcoin 10

3.1 Currency
3.2 Double Spending
3.3 Cryptocurrency
3.4 P2P Payment Gateway
3.5 Wallet
3.6 Mining

Unit 4. Ethereum 10

4.1. Ethereum network
4.2. EVM
4.3.Transaction fee
4.4.Mist
4.5.Ether, gas
4.6.Solidity - Smart contracts
4.7.Truffle
4.8.Web3
4.9.Design and issue Cryptocurrency
4.10. Mining
4.11. DApps
4.12. DAO

Unit 5 Introduction To Hyperledger Fabric V1.1 12

5.1. Introduction to Hyperledger
5.2 What is Hyperledger
5.3 Why Hyperledger
5.4 Where can Hyperledger be used
5.5 Hyperledger Architecture
5.6 Membership
5.7 Blockchain
5.8 Transaction
5.9 Chaincode
5.10 Hyperledger Fabric
5.11 Features of Hyperledger

References: Text Book 1.

1. Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller and Steven Goldfeder,
2. Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction, Princeton University Press (July 19, 2016).

Reference Books

1. Antonopoulos, Mastering Bitcoin: Unlocking Digital Cryptocurrencies
2. Satoshi Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System
3. DR. Gavin Wood, “ETHEREUM: A Secure Decentralized Transaction Ledger,”Yellow paper.2014.
4. Nicola Atzei, Massimo Bartoletti, and Tiziana Cimoli, A survey of attacks on Ethereum smart contracts

Semester -V	**Paper -IV**
Course Code: BBACA-504 T (A)	**Python**
Credits: 04	**Total Lectures: 60**

Prerequisites:
1. Experience with a high level language (C/C++, Java, MATLAB) is suggested.
2. Prior knowledge of a scriptinglanguage (Perl, UNIX/Linux shells) and Object-Oriented concepts is helpful but not mandatory.

Course Objectives:
1. To learn and understand Python programming basics and paradigm.
2. To learn and understand python looping, control statements and string manipulations.
3. Students should be made familiar with the concepts of GUI controls and designing GUI applications.
4. To learn and know the concepts of file handling, exception handling.

Course Outcomes: On completion of the course, student will be able
1. Define and demonstrate the use of built-in data structures "lists" and "dictionary".
2. Design and implement a program to solve a real world problem.
3. Design and implement GUI application and how to handle exceptions and files.

Unit 1 Introduction to Python 16

1.1 History, feature of Python, setting up path, working with python Interpreter, basic syntax, variable and data types, operators
1.2 Conditional statements-If, If-Else, nested if-else, Examples.
1.3 Looping-For,While,Nested loops, Examples
1.4 Control Statements-Break, Continue, Pass.
1.5 String Manipulation-Accessing String, Basic Operations, String Slices, Function and Methods, Examples.
1.6 Lists-Introduction, accessing list, operations, working with lists, function & methods.
1.7 Tuple-Introduction, Accessing tuples, operations working, function & methods, Examples.
1.8 Dictionaries-Introduction, Accessing values in dictionaries, working with dictionaries, properties, function, Examples.
1.9 Functions-Defining a function, calling a function, types of function, function arguments, anonymous function, global & local variable, Examples.

Unit 2 Modules and Packages 6

2.1Built in Modules
2.1.1 Importing modules in python program
2.1.2 Working with Random Modules.
2.1.3 E.g. - built-ins, time, date time, calendar, sys, etc
2.2 User Defined functions
2.2.1Structure of Python Modules
2.3 Packages
2.3.1 Predefined Packages
2.3.2User defined Packages

Unit 3 Classes ,Objects and Inheritance 12

3.1 Classes and Objects
3.1.1 Classes as User Defined Data Type
3.1.2 Objects as Instances of Classes
3.1.3 Creating Class and Objects
3.1.4 Creating Objects By Passing Values
3.1.5 Variables & Methods in a Class
3.2 Inheritance
3.2.1 Single Inheritance
3.2.2 Multilevel Inheritance
3.2.3 Multiple Inheritance
3.2.4 Hybrid Inheritance
3.2.5 Hierarchical Inheritance
3.2.6 IS-A Relationship and HAS-A Relationship

Unit 4 Exception Handling **8**

4.1 Python Exception
4.2 Common Exception
4.3 Exception handling in Python (try-except-else)
4.4 The except statement with no exception
4.5 Multiple Exception
4.6 The try-finally clause
4.7 Custom Exception and assert statement

Unit 5 GUI Programming **8**

5.1 Introduction
5.2 Tkinter programming
5.4 Tkinter widgets
5.5 Frame
5.6 Button
5.7 Label
5.8 Entry

Unit 6 Python Libraries **8**

6.1 Statistical Analysis- NumPy, SciPy, Pandas, StatsModels
6.2 Data Visualization- Matplotlib, Seaborn, Plotly
6.3 Data Modelling and Machine Learning- Scikit-learn, XGBoost, Eli5
6.4 Deep Learning- TensorFlow, Pytorch, Keras
6.5 Natural Language Processing (NLP)- NLTK, SpaCy, Gensim

Suggested Readings:

1.Mark Lutz, Programming Python, O`Reilly, 4th Edition, 2010
2.Dive into Python, Mike
3. Learning Python, 4th Edition by Mark Lutz
4. Programming Python, 4th Edition by Mark Lutz
5.Python Programming:An introduction to computer,John Zelle,3rd Edition.

Semester -V	**Paper -IV**
Course Code: BBACA-504 T (B)	**NO SQL**
Credits: 04	**Total Lectures: 60 Hrs.**

Course Objectives:

1. Explore the origins of NoSQL databases and the characteristics that distinguish them from traditional relational database management systems.
2. Understand the architectures and common features of the main types of NoSQL databases (key-value stores, document databases, column-family stores, graph databases)
3. Discuss the criteria that decision makers should consider when choosing between relational and non-relational databases and techniques for selecting the NoSQL database that best addresses specific use cases.

UNIT-I Overview and History of NoSQL 12
Databases. Definition of the Four Types of NoSQL Database, The Value of Relational Databases, Getting at Persistent Data, Concurrency, Integration, Impedance
Mismatch, Application and Integration Databases, Attack of the Clusters, The Emergence of NoSQL,
Key Points.

UNIT-II Comparison of relational databases to new NoSQL stores 12
MongoDB, Cassandra, HBASE, Neo4j use and deployment, Application, RDBMS approach, Challenges NoSQL approach, Key-Value and Document Data Models, Column-Family Stores, Aggregate-Oriented Databases. Replication and sharding, MapReduce on databases. Distribution Models, Single Server, Sharding, Master-Slave Replication, Peer-to-Peer Replication, Combining Sharding and Replication.

UNIT-III NoSQL Key/Value databases using MongoDB, 12
NoSQL Key/Value databases using MongoDB, Document Databases, Document oriented Database
Features, Consistency, Transactions, Availability, Query Features, Scaling, Suitable Use Cases, Event
Logging, Content Management Systems, Blogging Platforms, Web Analytics or Real-Time Analytics,
E-Commerce Applications, Complex Transactions Spanning Different Operations, Queries against
Varying Aggregate Structure.

UNIT-IV Column- oriented NoSQL databases using Apache 12

HBASE, Column-oriented NoSQL databases using Apache Cassandra, Architecture of HBASE, Column-Family Data Store Features, Consistency,
Transactions, Availability, Query Features, Scaling, Suitable Use Cases, Event Logging, Content Management Systems, Blogging Platforms, Counters, Expiring Usage.

UNIT-V NoSQL Key/Value databases using Riak 12

NoSQL Key/Value databases using Riak, Key-Value Databases,Key-Value Store, Key-Value Store Features, Consistency, Transactions, Query Features, Structure of Data, Scaling, Suitable Use Cases, Storing Session Information, User Profiles, Preferences, Shopping Cart Data,Relationships among Data, Multi operation Transactions, Query by Data, Operations by Sets. Graph NoSQL databases using Neo4,NoSQL database development tools and programming languages, Graph Databases, Graph Database. Features, Consistency, Transactions, Availability, Query Features, Scaling, Suitable Use Cases.

Suggested Readings:

1. Sadalage, P. & Fowler, NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence, Wiley Publications,1st Edition ,2019.

Semester -V	**Paper -V**
Course Code: BBACA-505 P	**PRACTICALS(BASED ON BBACA-503 AND 504)**
Credits: 04	**Total Lectures: 60 Hrs.**

PYTHON

Ass. No.	**Week**	**Assignment**
1.	First	1. Write python script to calculate sum of digits of a given input number. 2. Write python script to check whether a input number is Armstrong number or not. 3. Write python script to check whether a input number is perfect number of not. 4. Write a program to calculate XY 5. Write a program to check whether a input number is palindrome or not. 6. Write a program to calculate sum of first and last digit of a number.
2.	Second	1. Write a program to accept a number and count number of even, odd, zero digits within that number. 2. Write a program to accept a binary number and convert it into decimal number. 3. Write a program which accepts an integer value as command line and print “Ok” if value is between 1 to 50 (both inclusive) otherwise it prints ”Out of range” 4. Write a program which accept an integer value ‘n’ and display all prime numbers till ‘n’. 5. Write python script to accept two numbers as range and display multiplication table of all numbers within that range.

3.	Third	1. Write a python script which accepts 5 integer values and prints “DUPLICATES” if any of the values entered are duplicates otherwise it prints “ALL UNIQUE”. Example: Let 5 integers are (32, 45, 90, 45, 6) then output “DUPLICATES” to be printed. 2. Write a python script to count the number of characters (character frequency) in a string. Sample String : google.com'. Expected Result : {'o': 3, 'g': 2, '.': 1, 'e': 1, 'l': 1, 'm': 1, 'c': 1} 3. Write a Python program to remove the characters which have odd index values of a given string. 4. Write a program to implement the concept of stack using list 5. Write a Python program to get a string from a given string where all occurrences of its first char have been changed to '$', except the first char itself. Sample String: 'restart' Expected Result : 'resta$t'
4.	Fourth	1. Write a Python program to get a string made of the first 2 and the last 2 chars from a given a string. If the string length is less than 2, return instead of the empty string. Sample String : 'General12' Expected Result : 'Ge12' Sample String : 'Ka' Expected Result : 'KaKa' Sample String : ' K' 18 Expected Result : Empty String 2. Write a Python program to get a single string from two given strings, separated by a space and swap the first two characters of each string. Sample String : 'abc', 'xyz' Expected Result : 'xycabz' 3. Write a Python program to count the occurrences of each word in a given sentence. 4. Write a program to implement the concept of queue using list
5.	Fifth	1. Write a Python program to find maximum and the minimum value in a set. 2. Write a Python program to add an item in a tuple. 3. Write a Python program to convert a tuple to a string. 4. Write a Python program to create an intersection of sets. 5. Write a Python program to create a union of sets. 6. Write a Python script to check if a given key already exists in a dictionary. 7. Write a Python script to sort (ascending and descending) a dictionary by value.
6.	Sixth	1. Write a Python program to create set difference and a symmetric difference. 2. Write a Python program to create a list of tuples with the first element as the number and second element as the square of the number. 3. Write a Python program to unpack a tuple in several variables. 4. Write a Python program to get the 4th element from front and 4th element from last of a tuple. 5. Write a Python program to find the repeated items of a tuple. 6. Write a Python program to check whether an element exists within a tuple. 7. Write a Python script to concatenate following dictionaries to create a new one. Sample Dictionary : dic1={1:10, 2:20} dic2={3:30, 4:40} dic3={5:50,6:60} Expected Result : {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
7.	Seventh	1. Write a recursive function which print string in reverse order. 2. Write a python script using function to calculate XY 3. Define a function that accept two strings as input and find union and intersection of them. 4. Write a recursive function to calculate sum of digits of a given input number. 5. Write generator function which generate even numbers up to n
8.	Eighth	1. Write a python script to generate Fibonacci terms using generator function. 2. Write python script using package to calculate area and volume of cylinder and cuboids. 3. Write a python script to accept decimal number and convert it to binary and octal number using function. 4. Write a function which print a dictionary where the keys are numbers between 1 and 20 5. (both included) and the values are square of keys

		6. Write a generator function which generates prime numbers up to n.
9.	Ninth	1) Write a Python Program to Accept, Delete and Display students details such as Roll.No, Name, Marks in three subject, using Classes. Also display percentage of each student. 2) Write a Python program that defines a class named circle with attributes radius and center, where center is a point object and radius is number. Accept center and radius from user. Instantiate a circle object that represents a circle with its center and radius as accepted input. 3) Write a Python class which has two methods get_String and print_String. get_String accept a string from the user and print_String print the string in upper case. Further modify the program to reverse a string word by word and print it in lower case. 4) Write Python class to perform addition of two complex numbers using binary + operator overloading.
10.	Tenth	1) Define a class named Rectangle which can be constructed by a length and width. The Rectangle class has a method which can compute the area and volume. 2) Write a function named pt_in_circle that takes a circle and a point and returns true if point lies on the boundry of circle. 3) Write a Python Program to Create a Class Set and Get All Possible Subsets from a Set of Distinct Integers. 4) Write a python class to accept a string and number n from user and display n repetition of strings using by overloading * operator.
11.	Eleven	1) Write a python program to demonstrate multilevel inheritance by using Base class name as “Team” which inherits Derived class name as “Dev”. 2) Write a python program by considering Baseclass as TeamMember and Derived class as TeamLeader use multiple inheritance concept to demonstrate the code. 3) Write a python program to make use of issubclass () or isinstance() functions to check the relationships of two classes and instances.
12.	Twelve	1) Write a python program to inherit (Derived class) “course“ from (base class) “University” Using hybrid inheritance concept. 2) Write a python program to show the Hierarchical inheritance of two or more classes named as “Square “ & “ Triangle” inherit from a single Base class as “Area “ . 3) Define a class named Shape and its subclass (Square/Circle). The subclass has an init function which takes an argument (length/radius). Both classes have an area and volume function which can print the area and volume of the shape where Shape's area is 0 by default. 4) Python Program to Create a Class in which One Method Accepts a String from the User and Another method Prints it. Define a class named Country which has a method called print Nationality. Define subclass named state from Country which has a method called print State . Write a method to print state, country and nationality.
13.	Thirteen	1) Define a custom exception class which takes a string message as attribute. 2) Write a function called oops that explicitly raises a IndexError exception when called. Then write another function that calls oops inside a try/except statement to catch the error. 3) Change the oops function you just wrote to raise an exception you define yourself, called MyError, and pass an extra data item along with the exception. Then, extend the try statement in the catcher function to catch this exception and its data in addition to IndexError, and print the extra data item.
14.	Fourteen	1) Define a class Date(Day, Month, Year) with functions to accept and display it. Accept date from user. Throw user defined exception “invalidDateException” if the date is invalid. 2) Write text file named test.txt that contains integers, characters and float numbers. Write a Python program to read the test.txt file. And print appropriate message using exception

15	Fifteen	1. Write Python GUI program to display an alert message when a button is pressed. 2. Write Python GUI program to Create background with changing colors 3. Write a Python GUI program to create a label and change the label font style (font name, bold, size) using tkinter module. 4. Write a Python GUI program to create a Text widget using tkinter module. Insert a string at the beginning then insert a string into the current text. Delete the first and last character of the text. 5. Write a Python GUI program to accept dimensions of a cylinder and display the surface area and volume of cylinder. 6. Write Python GUI program that takes input string and change letter to upper case when a button is pressed.
16	Sixteen	1. Write Python GUI program to take input of your date of birth and output your age when a button is pressed. 2. Write Python GUI program which accepts a sentence from the user and alters it when a button is pressed. Every space should be replaced by *, case of all alphabets should be reversed, digits are replaced by ?. 3. Write Python GUI A program to create a digital clock with Tkinter to display the time. 4. Create a program to generate a random password with upper and lower case letters. 5. Write Python GUI program which accepts a number n to displays each digit of number in words. 6. Write Python GUI program to accept a decimal number and convert and display it to binary, octal and hexadecimal number. 7. Write Python GUI program to add items in listbox widget to print and delete the selected items from listbox on button click. Provide two separate button for print and delete.
17	Seventeen	

MongoDB Practical Assignment

1. Create a database with the name ‘Movie’.

A ‘Film’ is a collection of documents with the following fields: a. Film Id b. Title of the film c. Year of release d. Genre / Category (like adventure, action, sci-fi, romantic etc.) A film can belong to more than one genre. e. Actors (First name and Last name) A film can have more than one actor. f. Director (First name and Last name) A film can have more than one director. g. Release details (It consists of places of release, dates of release and rating of the film.) 3. An ‘Actor’ is a collection of documents with the following fields: a. Actor Id b. First name c. Last Name d. Address (Street, City, State, Country, Pin-code) e. Contact Details (Email Id and Phone No) f. Age of an actor.

Queries:

1. Insert at least 10 documents in the collection Film – a. Insert at least one document with film belonging to two genres. b. Insert at least one document with film that is released at more than one place and on two different dates. c. Insert at least three documents with the films released in the same year. d. Insert at least two documents with the films directed by one director. e. Insert at least two documents with films those are acted by a pair ‘Madhuri Dixit’ and ‘Shahrukh Khan’.
2. Insert at least 10 documents in the collection Actor. Make sure, you are inserting the names of actors who have acted in films, given in the ‘Film’ collection.
3. Display all the documents inserted in both the collections.
4. Add a value to the rating of the film whose title starts with ‘T’.

5. Add an actor named " " in the 'Actor' collection. Also add the details of the film in 'Film' collection in which this actor has acted in.
6. Delete the film " ".
7. Delete an actor named " ".
8. Delete all actors from an 'Actor' collection who have age greater than " "
9. Update the actor's address where Actor Id is " ".
10.Update the genre of the film directed by " ".

Assignment 2

1.Create a database with name 'Company'.
2.2. An 'Employee' is a collection of documents with the following fields: a. Employee ID b. First Name c. Last Name d. Email e. Phone No. f. Address (House No, Street, City, State, Country, Pin-code) g. Salary h. Designation i. Experience j. Date of Joining k. Birthdate
3. A 'Transaction' is a collection of documents with the following fields: a. Transaction Id, b. Transaction Date c. Name (First Name of employee who processed the transaction) d. Transaction Details (Item Id, Item Name, Quantity, Price) e. Payment (Type of Payment (Debit/Credit/Cash), Total amount paid, Payment Successful) f. Remark (Remark field can be empty.)
   Queries:
   1. Insert at least 5 documents in 'Employee' collection.
   2. Insert multiple documents (at least 10) into the 'Transaction' collection by passing an array of documents to the db.collection.insert () method.
   3. Display all the documents of both the collections in a formatted manner.
   4. Update salary of all employees by giving an increment of Rs. 4000.
   5. Update the remark for transaction id 201.
   6. Update designation of an employee named "_ " from supervisor to manager.
   7. Update designation of an employee having Employee Id as .
   8. Change the address of an employee having Employee Id as .
   9. Delete transaction made by " " employee on the given date. 10.Delete all the employees whose first name starts with 'K'.

Assignment 3 This assignment is based on 'Movie' database having collections 'Film' and 'Actor'. Prerequisite: Read MongoDB Aggregate framework before executing the following assignments.
Note: It is expected that student should fill in the data relevant to the queries given in the assignment. The result set should not be empty.
1. Find the titles of all the films starting with the letter 'R' released during the year 2009 and 2011.
2. Find the list of films acted by an actor " ".
3. Find all the films released in 90s.
4. Find all films belonging to "Adventure" and "Thriller" genre.
5. Find all the films having 'A' rating.
6. Arrange the film names in ascending order and release year should be in descending order.
7. Sort the actors in ascending order according to their age.
8. Find movies that are comedies or dramas and are released after 2013.
9. Show the latest 2 films acted by an actor " ".
10.List the titles of films acted by actors " " and " ".
11.Retrieve films with an actor living in Spain.

12.Retrieve films with actor details. Note: Similarly, additional queries can be executed based on these collections for practice.

Assignment 4 This assignment is based on ‘Company’ database having collections ‘Employee’ and ‘Transaction’. Prerequisite: Read MongoDB Aggregate framework before executing the following assignments. Note: It is expected that student should fill in the data relevant to the queries given in the assignment. The result set should not be empty.
1. Find employees having designation as either ‘manager’ or ‘floor supervisor’.
2. Find an employee whose name ends with " " and print the output in json format.
3. Display the name of an employee whose salary is greater than using a MongoDB cursor.
4. Sort the employees in the descending order of their designation.
5. Count the total number of employees in a collection.
6. Calculate the sum of total amount paid for all the transaction documents.
7. Calculate the sum of total amount paid for each payment type.
8. Find the transaction id of the latest transaction.
9. Find designation of employees who have made transaction of amount greater than Rs. 500. 10. Find the total quantity of a particular item sold using Map Reduce.

Semester -V	**Paper –VI**
Course Code: BBACA-506 T	**Object Oriented Software Engineering**
Credits: 04	**Total Lectures: 60**

Pre Requisite: Students shall have the Basic Knowledge of Software Engineering

OBJECTIVES:

1. To understand the fundamentals of object modeling
2. To understand and differentiate Unified Process from other approaches.
3. To design with static UML diagrams.
4. To design with the UML dynamic and implementation diagrams.
5. To improve the software design with design patterns.
6. To test the software against its requirements specification.

Outcomes:

1. Students will be able to give Design Specifications for Project.
2. Students will acquire Knowledge in Basic Modeling.
3. Students will acquire Project Management Skills.

UNIT 1 Introduction and basics of Software Modelling 4

1.1. Software Life Cycle Models (Revision of SE)
1.2 System Concepts
1.3 Project Organization
1.4 Communication in Project Management
1.5 Risk management in Project Management

UNIT 2 SRS Documentation 4

2.1SRS Specification
2.2. Requirement Elicitation
2.3 Business Engineering

UNIT 3 Introduction to UML 4

3.1 Concept of UML
3.2 Advantages of UML

UNIT 4 Object Oriented Concepts and Principles **5**

4.1 What is Object Orientation? - Introduction , Object , Classes and Instance , Polymorphism, Inheritance
4.2 Object Oriented System Development- Introduction, Function/Data Methods (With Visibility), Object Oriented Analysis, Object Oriented Construction
4.3 Identifying the Elements of an Object Model
4.4. Identifying Classes and Objects
4.5. Specifying the Attributes (With Visibility)
4.6 Defining Operations
4.7 Finalizing the Object Definition

UNIT 5 Structural Modeling 15

5.1 Classes
5.2.Relationship
5.3. Common Mechanism
5.4. Class Diagram (Minimum three examples should be covered)
5.5. Advanced Classes
5.6. Advanced Relationship
5.7. Interface
5.8. Types and Roles
5.9. Packages
5.10. Object Diagram (Minimum three examples should be covered)

UNIT 6 Basic Behavioural Modeling 12

6.1. Interactions
6.2. Use Cases and Use Case Diagram with stereo types (Minimum three examples should be covered)
6.3. Interaction Diagram (Minimum two examples should be covered)
6.4. Sequence Diagram (Minimum two examples should be covered)
6.5. Activity Diagram (Minimum two examples should be covered)
6.6. State Chart Diagram (Minimum two examples should be covered)

UNIT 7 Architectural Modelling 8

7.1. Component
7.2. Components Diagram (Minimum two examples should be covered)
7.3. Deployment Diagram (Minimum two examples should be covered)
7.4. Collaboration Diagram (Minimum two examples should be covered)

UNIT 8 Object Oriented Analysis 4

8.1. Iterative Development and the Rational Unified Process
8.2. Inception
8.3. Understanding Requirements
8.4. Use Case Model From Inception to Elaboration
8.5. Elaboration

UNIT 9 Object Oriented Design 4

9.1. The Booch Method, The Coad and Yourdon Method and Jacobson Method and Raumbaugh Method
9.2. The Generic Components of the OO Design Model

Reading Reference:

1. The Unified Modeling Language User/Reference Guide, Grady Booch, James Rambaugh Pearson Education Inc
2. The Unified software development Process Ivar Jacobson, Grady Booch, James Rambaugh Pearson Education
3. Agile Software development Alistair Cockbair Pearson Educatio

Semester -V	**Paper –IV**
Course Code: BBACA-507 T	**Cyber Security**
Credits: 02	**Total Lectures: 30**

Prerequisites: -

- A course on Computer Networks.

Course Objectives:

- To understand the fundamentals of cyber security.
- To understand various categories of Cybercrime, Cyber-attacks on mobile, tools and techniques used in Cybercrime and case studies.
- To have an overview of the Cyber laws and concepts of Cyber forensics.

Course Outcome:-

- Have a good understanding of Cyber Security and the Tools.
- Identify the different types of Cyber Crimes.
- Have a good understanding of Cyber laws
- To develop Cyber forensics awareness.
- Identify attacks, security policies and credit card frauds in mobile and Wireless Computing Era.

Chapter 1:- Introduction to Cyber Crime and Cyber Security 10

1. Introduction
2. Cybercrime: Definition and Origin of the Word
3. Cybercrime and Information Security
4. Who are Cybercriminals?
5. Classifications of Cybercrimes:
   E-Mail Spoofing, Spamming, Cyber defamation, Internet Time Theft, Salami Attack/Salami Technique, Data Diddling,Forgery, Web Jacking, Newsgroup, Spam/Crimes Emanating from Usenet Newsgroup, Industrial

Spying/Industrial Espionage, Hacking,OnlineFrauds,Computer Sabotage, Email Bombing/Mail Bombs, Computer Network Intrusions, Password Sniffing, Credit Card Frauds, Identity Theft

6. Definition of Cyber Security
7. Vulnerability, Threats and Harmful acts
8. CIA Triad
9. Cyber Security Policy and Domains of Cyber Security Policy

Chapter 2 :- Cyber offenses and Cyberstalking **15**

1. Criminals Plan: Categories of Cybercrime Cyber Attacks: Reconnaissance, Passive Attack, Active Attacks, Scanning/Scrutinizing gathered Information, Attack (Gaining and Maintaining the System Access), Social Engineering, and Classification of Social Engineering.
2. Cyberstalking: Types of Stalkers, Cases Reported on Cyberstalking, Working of Stalking
3. Real-Life Incident of Cyber stalking
4. Cybercafe and Cybercrime
5. Botnets: The Fuel for Cybercrime, Botnet, Attack Vector
6. Cybercrime: Mobile and Wireless Devices – Proliferation - Trends in Mobility
7. Credit Card Frauds in Mobile and Wireless Computing Era
8. Security Challenges Posed by Mobile Devices
9. Authentication Service Security
10. Attacks on Mobile/Cell Phones

Chapter 3:- Tools and Methods Used in Cybercrime **10**

1. Introduction
2. Proxy Servers and Anonymizers
3. Phishing
4. Password Cracking
5. Keyloggers and Spywares
6. Virus and Worms
7. Trojan Horses and Backdoors
8. Steganography
9. DoS and DDoS Attacks
10. SQL Injection

Chapter 4 :- Cybercrimes and Cyber security: The Legal Perspectives **10**

1. Introduction
2. Cybercrime and the Legal Landscape around the World
3. Why Do We Need Cyberlaws: The Indian Context
4. The Indian IT Act
5. Challenges to Indian Law and Cybercrime Scenario in India
6. Consequences of not Addressing the Weakness in Information Technology Act
7. Digital Signatures and the Indian IT Act
8. Amendments to the Indian IT Act
9. Cybercrime and Punishment
10. Cyberlaw, Technology and Students: Indian Scenario

Chapter 5:- Cyber Forensics **05**

1. Introduction
2. Historical background of Cyber forensics
3. Digital Forensics Science

4. The Need for Computer Forensics
5. Cyber Forensics and Digital evidence
6. Forensics Analysis of Email
7. Digital Forensics Lifecycle
8. Challenges in Computer Forensics

Chapter 6:- Cybersecurity: Organizational Implications 10

1. Organizational Implications: Cost of cybercrimes and IPR issues
2. Web threats for organizations
3. Security and Privacy Implications from Cloud Computing
4. Social media marketing
5. Social computing and the associated challenges for organizations, Protecting people's privacy in the organization
6. Organizational guidelines for Internet usage and safe computing guidelines and computer usage policy
7. Incident handling

Chapter 7:- Cybercrime: Illustrations, Examples and Mini-Cases 05

1. Real-Life Examples
2. Mini-Cases
3. Illustrations of Financial Frauds in Cyber Domain
4. Digital Signature-Related Crime Scenarios
5. Digital Forensics Case Illustrations
6. Online Scams

References Books:

1. Cyber Security Understanding Cyber Crimes, Computer Forensics and Legal Perspectives – Nina Godbole, SunitBelapure, Wiley: April 2011 India Publications Released.
2. Principles of Information Security,-Michael E Whitman, Herbert J Mattord, 3rd Edition, 2011.
3. Computer Security: Principles and Practice -William Stallings and Lawrie Brown, 3rd edition, Pearson, 2015.
4. Cyber Security Essentials- James Graham Richard Howard Ryan Olson

Semester -VI	**Paper -IV**
Course Code: BBACA-604	**Advance JAVA**
Credits: 04	**Total Lectures: 60**

Prerequisite: Students should know basic programming concepts.

Objectives -:

1. To know the concept of Java Programming.
2. To understand how to use programming in day to day applications.

Outcomes:

3. To develop programming logic.

1. Students will know the concepts of JDBC Programming.
2. Students will know the concepts of Multithreading and Socket Programming.
3. Students will know the concepts of Spring and Hibernate.
4. Students will develop the project by using JSP and JDBC.
5. Students will develop applications in Spring and hibernate.

1.		**JDBC**	**8**
	1.	Introduction	
	2.	JDBC Architecture.	
	3.	JDBC Process	
	4.	**Working with ResultSet Interface.**	
2		**Multithreading:**	**12**
	1.	Introduction to Multithreading.	
	2.	Thread creation: Thread Class, Runnable Interface.	
	3.	Life cycle of Thread.	
	4.	Thread Priority.	
	5.	Execution of Thread Application.	
	6.	Synchronization and Interthread communication.	
3		**Networking:**	**5**
	1.	Overview of Networking.	
	2.	Networking Basics: Port Number, Protocols and classes.	
	3.	Sockets, Reading from and Writing to a Socket.	
4		**Servlet and JSP**	**12**
	1.	Introduction to Servlet	
	2.	Types of Servlet: Generic Servlet and Http Servlet	
	3.	Life cycle of servlet	
	4.	Session Tracking.	
	5.	Servlet with database.	
		JSP	
		6. Introduction to JSP.	
		7. JSP Life Cycle.	
		8. Components of JSP.	
		9. JSP with Database.	
5		**Spring & Hibernate Spring:**	**11**
	1.	Introduction	
	2.	Applications and Benefits of spring	
	3.	Architecture and Environment Setup	
	4.	Hello World Example	
	5.	Core Spring- IoC Containers, Spring Bean Definition, Scope, Lifecycle	
		Hibernate	
		6. Architecture and Environment	
		7. Configuration, Sessions, Persistent Class	
		8. Mapping Files, Mapping Types	
		9. Examples	

Reference Books:

1. The Complete Reference – JAVA Herbert Schildt
2. Professional Hibernate, by Eric Pugh, Joseph D. Gradecki by Wiley Publishing, Inc., ISBN: 0- 7645-7677-1
3. Spring In Action, Craig Walls, Ryan Breidenbach, Manning Publishing Co., ISBN: 1- 932394- 35-4

4. Head First Servlets and JSP: Passing the Sun Certified Web Component Developer Exam -2nd Edition- Bryan Basham, Kathy Sierra, Bert Bates- O'REILLY.

Semester -VI	Paper -II
Course Code: BBACA-602 P	PRACTICALS(BASED ON BBACA-601-T)
Credits: 04	Total Lectures: 30

Ass. No.	Week	Assignment
1.	First	1. Write a java program to count number of records in a table. 2. Write a java program to display all the EName from Emp table. Assume Emp (ENo, EName, Salary) table is already created. 3. Write a java program to create Student table with attributes Rno, Sname, Per. 4. Write a java program to delete salary column from Emp table. Assume Emp table with attributes ENo, EName and salary is already created. 5. Write a java program to delete the details of given Teacher. Assume Teacher table with attributes tno, tname, subject is already created.
2.	Second	1. Write a java program to accept the details of Hospital (HId, HName, Address, PH_No) and store it into the database. (Use Swing). 2. Write a java program to display the details of Doctor (DNO, DName, Specialization) on JTable. Assume Doctor table is already created. 3. Write a java program to make the changes in data which is in ResultSet if you make the changes in data in database. 4. Write a java program for the following: Create a Table Alter a Table Drop a Table 5. Write a java program to accept the details of College (CID, CName, Address) and display it on next frame. (Use Swing and PreparedStatement).
3.	Third	1. Write a multithreading program in java to display all the integers between 1 to 100 randomly after 2 seconds. 2. Write a multithreading program in java to display all the characters between Z to A after 5 seconds. 3. Write a java program to print “Hello Java” message 10 times. 4. Write a program in which thread sleep for 6 sec in the loop in reverse order from 100 to 1 and change the name of thread. 5. Write a java program to display all the vowels from a given String. Each vowel should display after 3 seconds.
4.	Fourth	1. Program to define a thread for printing text on output screen for ‘n’ number of times. Create 3 threads and run them. Pass the text ‘n’ parameters to the thread constructor. Example: i. First thread prints “Hello” 10 times. ii. Second thread prints “Good Morning” 20 times iii. Third thread prints “Sir/Mam” 30 times. 2. Write a java program in multithreading to create 3 balls and bounce them vertically. 3. Write a java program in multithreading to draw five lines vertically. 4. Write a java program to create 2 cars and move them randomly from left to right. 5. Write a java program to display the characters from a given string into the TextField. Each character should be displayed after 1 second.
5.	Fifth	1. Write a java program to display IP address of a machine. 2. Write a java program to accept a number in client application, send it to the server, server will check whether it is Armstrong or not and sends result accordingly to the client.

		3. Write a java program to send “Hi” message to the Server. 4. Write a java program to accept a user name in client application and greets to the user according to the time on server machine. 5. Write a java program to accept a string in client application and display the vowels from that string on server terminal
6.	Sixth	1. Write a java program for standalone chatting application. 2. Write a java program to accept a filename in client application and check whether it is available on server machine or not, if it there then display its contents on client’s terminal otherwise display the message “File Not Found”. (Write Client and Server application on different terminals.) 3. Write a java program to accept a number in client application, send it to the server, server will calculates its factors and display them on the client terminal. (Each factor should display after 2 seconds). Write Client and Server application on different terminals.
7.	Seventh	a) Design a servlet that provides information about a HTTP request from a client, such as IP address and browser type. The servlet also provides information about the server on which the servlet is running, such as the operating system type, and the names of currently loaded servlets. b) Write a Program to make use of following JSP implicit objects: i. out: To display current Date and Time. ii. request: To get header information. iii. response: To Add Cookie iv. config: get the parameters value defined in <init-param> v. application: get the parameter value defined in <context-param> vi. session: Display Current Session ID vii. pageContext: To set and get the attributes. viii. page: get the name of Generated Servlet c) Write a Servlet application to display Hello Java Message on the Browser. d) Write a JSP script to display all the prime number’s between 1 to n in Red Color. e) Write a JSP application to accept the details of Emp (Eno, EName, Salary) and display it in Tabular format on the browser.
8.	Eighth	a) Design an HTML page which passes customer number to a search servlet. The servlet searches for the customer number in a database (customer table) and returns customer details if found the number otherwise display error message. b) Design an HTML page containing option buttons (Maths, Physics, Chemistry and Biology) and buttons submit and reset. When the user clicks submit, the server responds by adding a cookie containing the selected subject and sends a message back to the client. Program should not allow duplicate cookies to be written. c) Write a Servlet program to display the details of PATIENT (PatientNo, PatientName, PatientAddress, Patientage, PatientDiease) in tabular form on browser. d) Write a JSP application to accept a user name and greets to the user according to the system Time. e) Write a Servlet program for the implementation of session tracking.
9.	Ninth	a) Create a Spring core example to display the message “DYPIAN”.

		b) Write a program to display the Current Date using spring. c). Write a Hibernate application to display “Hello” message.
10.	Tenth	a) Design a Spring application for accepting student information like Student_id, Student_Name and Student_Age. b) Design an Employee login form application using spring form MVC validation

Semester -VI	Paper -III
Course Code: BBACA-603 T (A)	SOFTWARE TESTING
Credits: 04	Total Lectures: 60

Prerequisite:

1. Students shall have basic Knowledge of Software Engineering.
2. Students shall have basic Knowledge of OOSE.

Objectives:

1. To provide learner with knowledge in Software Testing techniques.
2. To understand how testing methods can be used as an effective tool in providing quality assurance for software.
3. To provide skills to design test case plan for testing software.

Outcomes:

1. Students will be introduced to testing tools.
2. Students will acquire Knowledge of Basic SQA.
3. Students will be able to design basic Test Cases.

1.Introduction **10**

1. Introduction, Nature of errors,
2. Testing Objectives
3. Testing principles
4. Testing fundamentals,
5. Software reviews, Formal Technical reviews,
6. Inspection and walkthrough
7. Testing Life Cycle

2.Approaches to Testing –Testing Methods **05**

1. White Box Testing and types of white box testing
2. Test Case Design
3. Black Box Testing and types of black box testing
4. Gray Box Testing

3.Software Testing Strategies &Software metrics **15**

1. Software Testing Process
2. Unit Testing
3. Integration- Top-down ,Bottom up
4. System Testing
5. Acceptance Testing (alpha, Beta testing)
6. Validation and Verification

7. Big Bang Approach
8. Sandwich approach
9. Performance Testing
10. Regression Testing
11. Smoke Testing

3.13 Load Testing

4.Software metrics **05**

1. Introduction
2. Basic Metrics –size-oriented metric, Function –oriented metric
3. Cyclometic Complexity Metrics Examples on Cyclometic Complexity

5.Testing for Specialized Environments **10**

1. Testing GUI’s
2. Testing of Client/Server Architectures
3. Testing Documentation and Help Facilities
4. Testing for Real-Time Systems

6.Testing Tools& Software Quality Assurance (Introduction) 15

1. JUnit, Apache JMeter, Win runner
2. Load runner, Rational Robot
3. Quality Concepts, Quality Movement, Background Issues, SQA activities
4. Formal approaches to SQA
5. Statistical Quality Assurance
6. Software Reliability
7. The ISO 9000 Quality Standards
8. SQA Plan
9. Six sigma

Reading Reference:

1. **Software Engineering –A Practitioner’s approach,RogerSPressman,7th Edition Tata McGraw-Hill**
2. Effective Methods of Software Testing.**William E Perry,**Wiley Publishing Inc
3. **Software Testing Principles and Practices ,Srinivasan Desikan, Gopalswamy Ramesh,Pearson Publication**
4. **Total Quality Management,DaleH. Besterfield,Prentice Hall, 2003**

Semester -VI	**Paper -III**
Course Code: BBACA-603 T (B)	**ARTIFICIAL INTELLIGENCE CONCEPTS**
Credits: 04	**Total Lectures: 60**

UNIT 1 Artificial Intelligence **12**

1.1 Introduction& Concept of AI
1.2 Applications of AI
1.3 Artificial Intelligence, Intelligent Systems, Knowledge –based Systems, AI Techniques
1.4 Early work in AI & related fields.
1.5 Defining AI problems as a State Space Search
1.6 Search and Control Strategies
1.7 Problem Characteristics
1.8 AI Problem: Water Jug Problem, Tower of Hanoi, Missionaries & Cannibal Problem

UNIT 2 AI Search Techniques **12**

2.1 Blind Search Techniques: BFS, DFS, DLS, Iterative deepening Search, Bidirectional Search, and Uniform cost Search
2.2 Heuristic search techniques: Generate and test, Hill Climbing, Best First search, Constraint Satisfaction, Mean-End Analysis, A*, AO*

UNIT 3. **Machine Learning** **12**

3.1 Machine Learning (ML)
3.2. ML Techniques overview

- Supervised machine learning
- Unsupervised machine learning
- Semi-supervised learning
- Reinforcement machine learning

UNIT 4. Deep Learning **12**

4.1 What is Deep Learning?
4.2 Need for Data Scientists
4.3 What is Business Intelligence
4.4. What is Data Analysis
4.5. What is Data Mining
4.6. Advantage of Deep Learning over Machine learning

UNIT 5. Artificial Intelligence in Business and Society **12**

5.1. Artificial Intelligence Applications

- Artificial Intelligence applications
- Language Models
- Information Retrieval
- Information Extraction
- Natural Language Processing
- Machine Translation
- Speech Recognition
- Robotics

Reference Readings:
1. Artificial Intelligence by Elaine Rich, Kevin Knight, TMH, 2nd Edition.
2. Artificial Intelligence: A new Synthesis, Nilsson, Elsevier,ISBN 9788181471901

Semester -V	**Paper -IV**
Course Code: BBACA-604 T (A)	**DOT NET PROGRAMMING**
Credits: 04	**Total Lectures: 60**

Course Prerequisites:
Student should have basic knowledge of:

- Visual Basic
- HTML
- Object Oriented concepts
- Ms-Access, Mysql, SQL Server

Course Objectives:

- To learn Microsoft framework architecture.

- Understand development of windows application.
- To learn data access mechanism.
- Create and consume libraries.
- Create a web application.
- To develop the website and application.

Course Outcome:

- Use the features of Dot Net Framework along with the features of VB, C# and ASP
- Design and develop window based and web based .NET applications.
- Design and develop a Website.
- Design and Implement database connectivity using ADO.NET for VB, C# and ASP.

UNIT 1 Introduction to DOT NET FRAMEWORK 12

1.1. What is Framework?
1.2. Architecture of Dot Net Framework
1.3. Common Language Runtime
1.4. Common Type System(CTS)
1.5. Common Language Specification(CLS)
1.5.1. JIT Compilers
1.5.2. Base Class Library
1.6. IDE (Integrated Development Environment)
1.7. Event Driven Programming

UNIT 2 Introduction to VB.Net 12

2.1. Basics of VB.Net
2.2. Operators
2.3. Data Types
2.4. Control Structures 2.2Build Windows Applications
2.5. Controls: Form, TextBox, Button, Label, CheckBox, ListBox, ComboBox, RadioButton, DateTimePicker, MonthCalender,
2.6. Timer, Progressbar,Scrollbar, PictureBox, ImageBox, ImageList, TreeView, ListView, Toolbar, StatusBar, Datagridview
2.7. Menus and PopUp Menu
2.8. Predefined Dialog controls: Color,Save,File,Open, Font
2.9. DialogBox - InputBox(), MessageBox, MsgBox()

UNIT 3 Introduction to C# 12

3.1. Language Fundamentals
3.2. Data type and Control Constructs
3.3. Value and Reference Types,Boxing
3.4. Arrays
3.5. String class and its various operations
3.6. Functions
3.7. Object Oriented Concepts
3.8. Defining classes and Objects
3.9. Access modifiers
3.10. Constructors
3.11. Inheritance
3.12. Interface
3.12.1. Abstract Class
3.12.2. Method Overloading and Overriding
3.13. Delegates

UNIT 4 Introduction to ASP.NET 12

4.1. What isASP.NET?
4.2. ASP.NET Page Life Cycle
4.3. Architecture ofASP.NET
4.4. Forms, WebPages, HTML forms,
4.5. Request & Response in Non-ASP.NET pages
4.6. Using ASP.NET Server Controls
4.7. Overview of Control structures
4.8. Functions
4.9. HTML events
4.9.1.ASP.NET Web control events
4.9.2.Event driven programming and postback
4.10. Introduction to Web forms
4.10.1. Web Controls
4.10.2. Server Controls
4.10.3. Client Controls
4.10.4. Navigation Controls
4.10.5. Validations
4.10.6. Master Page
4.10.7. State Management Techniques
UNIT 5 Architecture of Ado.Net 12
5.1. Basics of Ado.net
5.1.1.Connection Object
5.1.2.ommand Object
5.1.3.Dataset
5.1.4.Data Table
5.1.5.Data Reader Object
5.1.6.Data Adapter Object
5.2. Datagridview& Data Binding: Insert, Update, Delete records
5.3. Navigation Using Data Source

Reference Books:

- Beginning Visual C#, WroxPublication
- BeginningASP.NET3.5,WroxPublication
- ProgrammingASP.NET3.5byJesseLiberty,DanMaharry,DanHurwitz,O'Reilly
- Programming Microsoft Visual Basic.NET – Francesco Balena
- The Complete Reference -Visual Basic .NET – Jefrey R. Shapiro
- ADO.NET Examples and Best Practices for C# Programmers, By Peter D, Blackburn, William
- VB.NET database programming with ADO.NET -Anne Prince and Doug Lowe

Semester -VI	Paper -IV
Course Code: BBACA-604 T (B)	ANDROID PROGRAMMING
Credits: 04	Total Lectures: 60

Pre-requisite:

1. Concepts of OOPs.
2. Basic Knowledge About JAVA Programming

Objective:

1. To understand the Android Operating System and develop applications using Google's Android open- source platform.
2. To understand the issues relating to Wireless applications.

Outcome:

1. Student will be able to write simple GUI applications, use built-in widgets and components, work with the database to store data locally, and much more.
2. Demonstrate their understanding of the fundamentals of Android operating systems
   Demonstrate their skills of using Android software development tools

UNIT 1 INTRODUCTION TO Android Programming 4

1. What is Android?
2. History and Versions
3. Android Architecture
4. Basic Building Blocks
5. Android API Levels
6. Application Structure
7. First Hello World Program

UNIT 2 ACTIVITY, INTENT AND LAYOUT 8

1. Introduction to Activity
2. Activity life cycle
3. Introduction to Intent
4. Types of Intent(Implicit and Explicit Intent)
5. Layout Manager

1. View and View Group
2. Linear Layout
3. Relative Layout
4. Table Layout
5. Grid Layout
6. Constraint Layout
7. Frame Layout
8. Scroll Layout

UNIT 3 BASIC UI DESIGN 12

1. Button(Push Button, Check Box, Radio Button, Toggle
   Button, Image Button)
2. Text Fields
3. Spinner
4. List View
5. Toast
6. Scroll View
   6. ProgressBar View
7. Auto Complete Text View
8. Dialog Box
   1. Alert Dialog.
   2. DatePicker Dialog.
   3. TimePicker Dialog.
   4. Custom Dialog.

UNIT 4 ADAPTER AND MENU 8

1. Base Adapter
2. Array Adapter
3. ListView using Adapter 4.4GridView
   using Adapter
   4.5Photo Gallery using Adapter

   4.6 Using Menu with Views
   4.6.1 Option Menu
2. Context Menu
3. Popup Menu

UNIT 5 THREADS AND NOTIFICATION 8

1. Worker thread
2. Handlers & Runnable
3. AsynTask (in detail)
4. Broadcast Receiver
5. Services 5.5.1Service life Cycle
   5.5.2 Bounded Service
   5.5.2 Unbounded Service
   6. Notification
   7. Alarm
8. Accessing Phone services(Call,SMS)

UNIT 6 CONTENT PROVIDER 10

1. Content Providers
2. SQLite Programming
3. SQLiteOpenHelper
4. SQLiteDatabse
5. Cursor
6. Searching for content
7. Adding, changing, and removing content
8. Building and executing queries
9. Android JSON

Semester -VI	**Paper -IV**
Course Code: BBACA-605 P	**PRACTICAL (BASED ON BBACA-604 T)**
Credits: 02	**Total Practical's: 30**

UNIT 7 LOCATION BASED SERVICES AND GOOGLE MAP 10

1. Display Google Maps
    1. Creating the project
    2. Obtaining the Maps API Key
    3. Displaying the Map
    4. Displaying the Zoom Control
    5. Changing Views
    6. Navigating to a specific location
    7. Adding Markers
    8. Getting the location that was touched
    9. Geocoding and Reverse Geocoding
2. Getting Location Data
3. Monitoring a Location

Reference Readings:

1. Beginning Android4 Application Development, By Wei-Meng Lee WILEY India Edition WROX Publication
2. Professional Android 4 Application Development, By Reto Meier WROX Publication
3. The official site for Android developers - https://developer.android.com

Section I: Android

Assignment 1: Introduction to Android

Objectives

- Study Android Studio installation.
- Create basic android application.

Reading

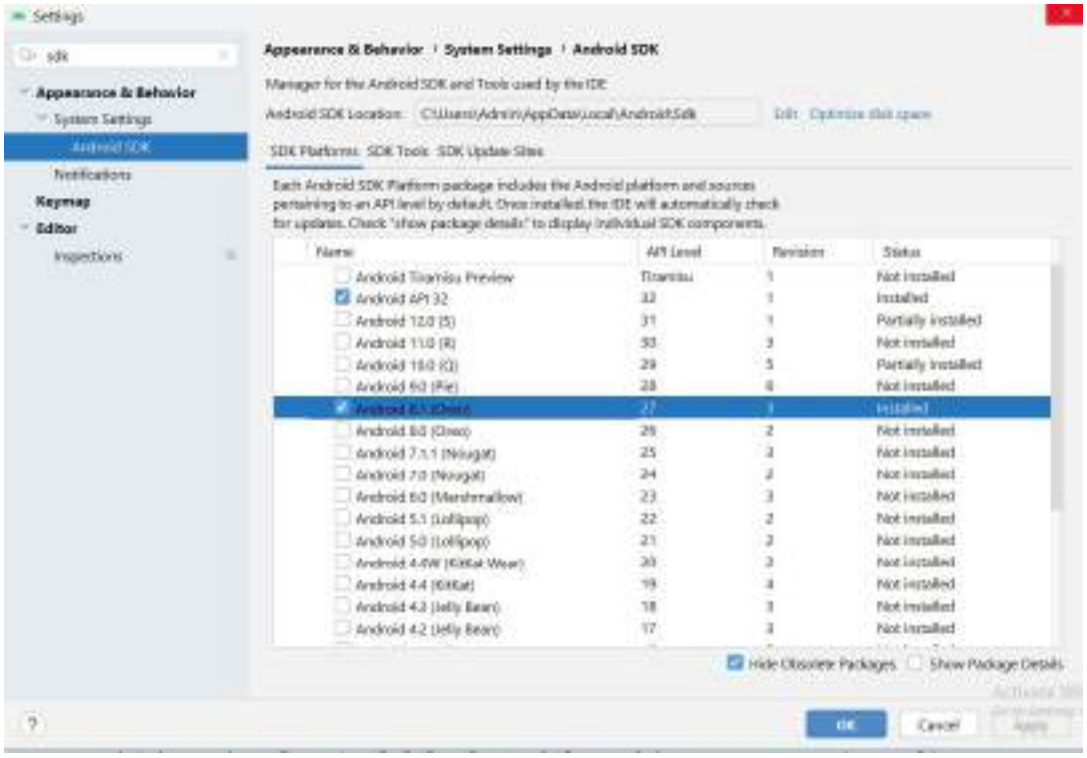
You should read the following topics before starting this exercise:

1. Installing Android Studio
2. Create "Hello World" application
3. Explore the project structure
4. The Gradle build system
5. Create a virtual device (emulator)
6. Run your app on an emulator
7. Android Architecture

1. Installing Android Studio

1) Install Java JDK: Get the latest version. The JDK may be downloaded here https://www.oracle.com/java/technologies/downloads/#java8

2) Install Android Studio bundle. Use the latest stable version. Android Studio may be downloaded



Appearance & Behavior › System Settings › Android SDK

Manager for the Android SDK and Tools used by the IDE

Android SDK Location: C:\Users\Admin\AppData\Local\Android\Sdk Edit Optimize disk space

SDK Platforms SDK Tools SDK Update Sites

Each Android SDK Platform package includes the Android platform and sources pertaining to an API level by default. Once installed, the IDE will automatically check for updates. Check "show package details" to display individual SDK components.

	Name	API Level	Revision	Status
☐	Android Tiramisu Preview	Tiramisu	1	Not installed
✓	Android API 32	32	1	Installed
☐	Android 12.0 (S)	31	1	Partially installed
☐	Android 11.0 (R)	30	3	Not installed
☐	Android 10.0 (Q)	29	5	Partially installed
☐	Android 9.0 (Pie)	28	6	Not installed
✓	Android 8.1 (Oreo)	27	3	Installed
☐	Android 8.0 (Oreo)	26	2	Not installed
☐	Android 7.1.1 (Nougat)	25	3	Not installed
☐	Android 7.0 (Nougat)	24	2	Not installed
☐	Android 6.0 (Marshmallow)	23	3	Not installed
☐	Android 5.1 (Lollipop)	22	2	Not installed
☐	Android 5.0 (Lollipop)	21	2	Not installed
☐	Android 4.4W (KitKat Wear)	20	2	Not installed
☐	Android 4.4 (KitKat)	19	4	Not installed
☐	Android 4.3 (Jelly Bean)	18	3	Not installed
☐	Android 4.2 (Jelly Bean)	17	3	Not installed

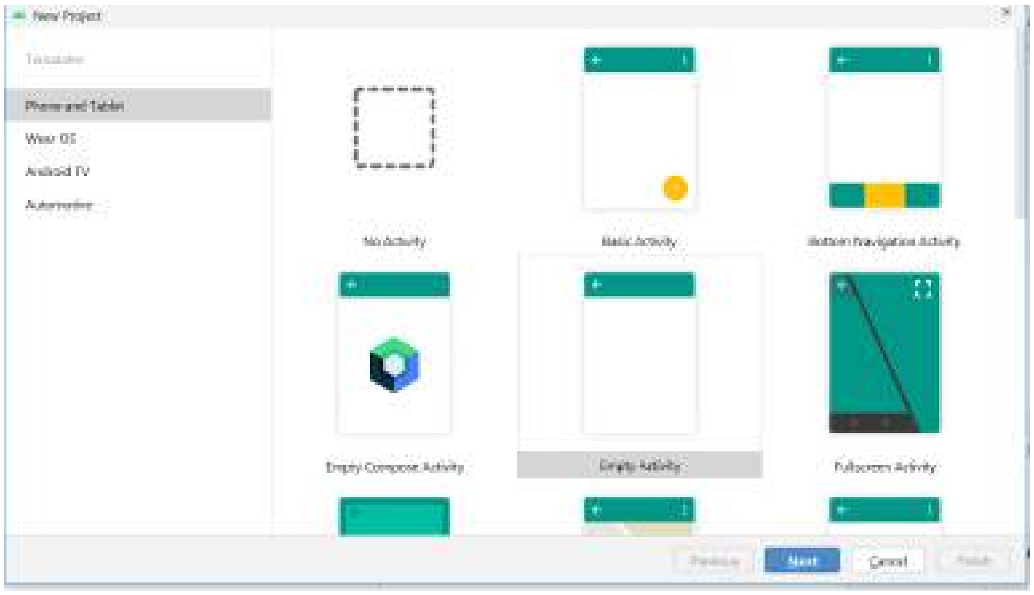
✓ Hide Obsolete Packages ☐ Show Package Details

here:https://developer.android.com/studio

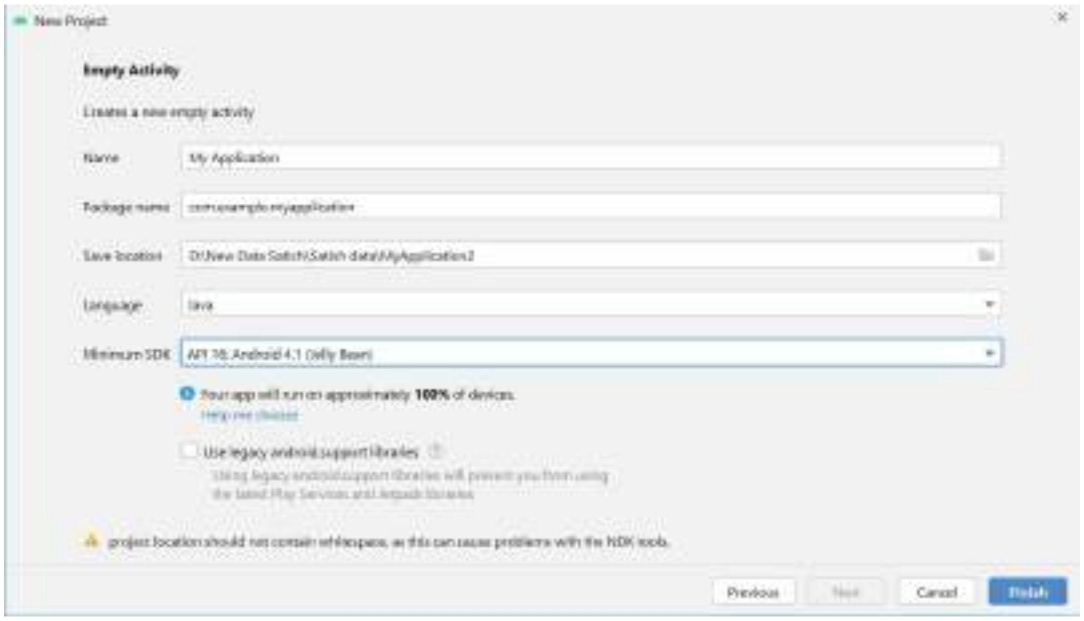
3) Follow the prompts to complete the installation. I used the default settings.
4) Allow Android Studio access to the network.
5) Select your desired UI theme.
6) Android Studio will download additional components. This will take several minutes.

7) Select —Configure/SDK Manager.
8) Deselect All. Scroll down and select —Android 8.1 (API 27) Install the packages.

2. **Create "Hello World" application**
   1. In the **Welcome to Android Studio** window, click **Create New Project**. (If you have a project alreadyopened, select **File > New > New Project**.)
   2. In the Select a Project Template window, select **Empty Activity** and click **Next.**



   3. In the Configure your project window, complete the following:
      a. Enter "MyFirstApp" in the Name field.
      b. Enter "com.example.myfirstapp" in the Package name field.
      c. If you'd like to place the project in a different folder, change its Save location.
      d. Select either Java or Kotlin from the Language drop-down menu.
      e. Select the lowest version of Android you want your app to support in the Minimum SDK field.
      f. Leave the other options as they are.



Creates a new empty activity

Name: My Application

Package name: com.example.myapplication

Save location: D:\New Data Satish\Satish data\MyApplication2

Language: Java

Minimum SDK: API 16: Android 4.1 (Jelly Bean)

Your app will run on approximately **100%** of devices.
Help me choose

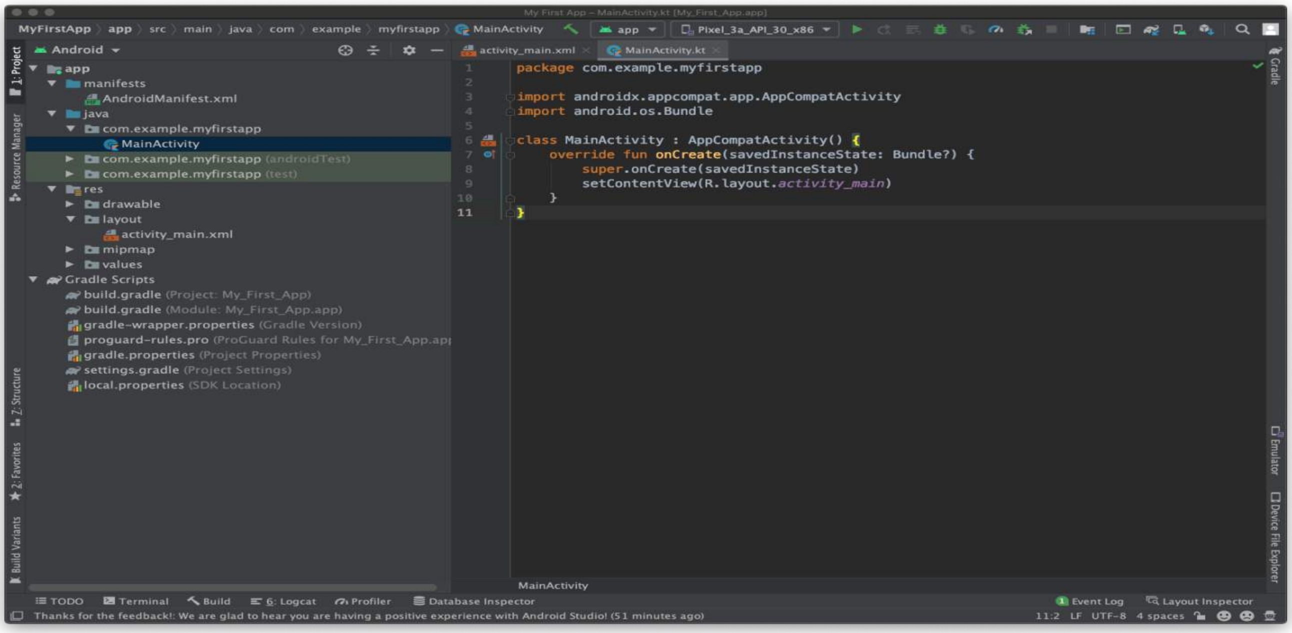
Use legacy android.support libraries

Using legacy android.support libraries will prevent you from using the latest Play Services and Jetpack libraries

project location should not contain whitespace, as this can cause problems with the NDK tools.

   4. Click Finish.

5. After some processing time, the Android Studio main window appears.


My First App – MainActivity.kt [My_First_App.app]
MyFirstApp 〉 app 〉 src 〉 main 〉 java 〉 com 〉 example 〉 myfirstapp 〉 MainActivity
app
Pixel_3a_API_30_x86
Android
app
manifests
AndroidManifest.xml
java
com.example.myfirstapp
MainActivity
com.example.myfirstapp (androidTest)
com.example.myfirstapp (test)
res
drawable
layout
activity_main.xml
mipmap
values
Gradle Scripts
build.gradle (Project: My_First_App)
build.gradle (Module: My_First_App.app)
gradle-wrapper.properties (Gradle Version)
proguard-rules.pro (ProGuard Rules for My_First_App.app)
gradle.properties (Project Properties)
settings.gradle (Project Settings)
local.properties (SDK Location)
activity_main.xml
MainActivity.kt
MainActivity
TODO
Terminal
Build
6: Logcat
Profiler
Database Inspector
Event Log
Layout Inspector
11:2 LF UTF-8 4 spaces

```
package com.example.myfirstapp

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }
}
```

Thanks for the feedback!: We are glad to hear you are having a positive experience with Android Studio! (51 minutes ago)

3. Explore the project structure

In the Project > Android view of your previous task, there are three top-level folders below your app folder: manifests, java, and res.

1. Expand the manifests folder.

This folder contains **AndroidManifest.xml**. This file describes all of the components of your Android app and is read by the Android run-time system when your program is executed.

2. Expand the java folder.

All your Java language files are organized in this folder. The java folder contains three subfolders:

- **com.example.hello.helloworld** (or the domain name you have specified): All the files for a packageare in a folder named after the package. For your Hello World application, there is one package and itonly contains MainActivity.java (the file extension may be omitted in the Project view).
- **com.example.hello.helloworld(androidTest)**: This folder is for your instrumented tests, and starts out with a skeleton test file.
- **com.example.hello.helloworld(test)**: This folder is for your unit tests and starts out with an automatically created skeleton unit test file.

3. Expand the res folder.

This folder contains all the resources for your app, including images, layout files, strings, icons, and styling. It includes these subfolders:

- **drawable**: Store all your app's images in this folder.
- **layout**: Every activity has at least one layout file that describes the UI in XML. For Hello World, thisfolder contains **activity_main.xml**.
- **mipmap**: Store your launcher icons in this folder. There is a sub-folder for each supported screen density. Android uses the screen density, that is, the number of pixels per inch to determine the

required image resolution. Android groups all actual screen densities into generalized densities, suchas medium (mdpi), high (hdpi), or extra-extra-extra-high (xxxhdpi). The ic_launcher.png folder contains the default launcher icons for all the densities supported by your app.

- **values**: Instead of hard coding values like strings, dimensions, and colors in your XML and Java files,it is best practice to define them in their respective values file. This makes it easier to change and be

  consistent across your app. Expand the values subfolder within the res folder. It includes these subfolders:

  - **colors.xml**: Shows the default colors for your chosen theme, and you can add your own colorsor change them based on your app's requirements.
  - **dimens.xml**: Store the sizes of views and objects for different resolutions.
  - **strings.xml**: Create resources for all your strings. This makes it easy to translate them to otherlanguages.
  - **styles.xml**: All the styles for your app and theme go here. Styles help give your app a consistent look for all UI elements.

4. The Gradle build system:

Android Studio uses Gradle as its build system. As you progress through these practicals, you will learn moreabout gradle and what you need to build and run your apps.

1. Expand the Gradle Scripts folder. This folder contains all the files needed by the build system.
2. Look for the build.gradle(Module:app) file. When you are adding app-specific dependencies,such as using additional libraries, they go into this file

5. Create a virtual device (emulator)

In this task, you will use the Android Virtual Device (AVD) manager to create a virtual device or emulator that simulates the configuration for a particular type of Android device. Using the AVD Manager, you define the hardware characteristics of a device and its API level, and save it as a virtual device configuration. Whenyou start the Android emulator, it reads a specified configuration and creates an emulated device that behaves exactly like a physical version of that device, but it resides on your computer. With virtual devices, you can test your apps on different devices (tablets, phones) with different API levels to make sure it looks good and works for most users. You do not need to depend on having a physical device available for app development.In order to run an emulator on your computer, you have to create a configuration that describes the virtual device.

- In Android Studio, select Tools > AVD Manager, or click the AVD Manager icon in the toolbar.
- Click the **+Create Virtual Device….** (If you have created a virtual device before, the window shows all of your existing devices and the button is at the bottom.) The Select Hardware screen appears showing a list of preconfigured hardware devices. For each device, the table shows its diagonal display size (Size), screen resolution in pixels (Resolution), and pixel density (Density).
- Choose which version of the Android system to run on the virtual device. You can select the latestsystem image. There are many more versions available than shown in the recommended tab.
- If a Download link is visible next to a system image version, it is not installed yet, and you need todownload it. If necessary, click the link to start the download, and click Finish when it's done.
- On System Image screen, choose a system image and click Next.
- Verify your configuration, and click Finish. (If the Your Android Devices AVD Manager window

stays open, you can go ahead and close it.)

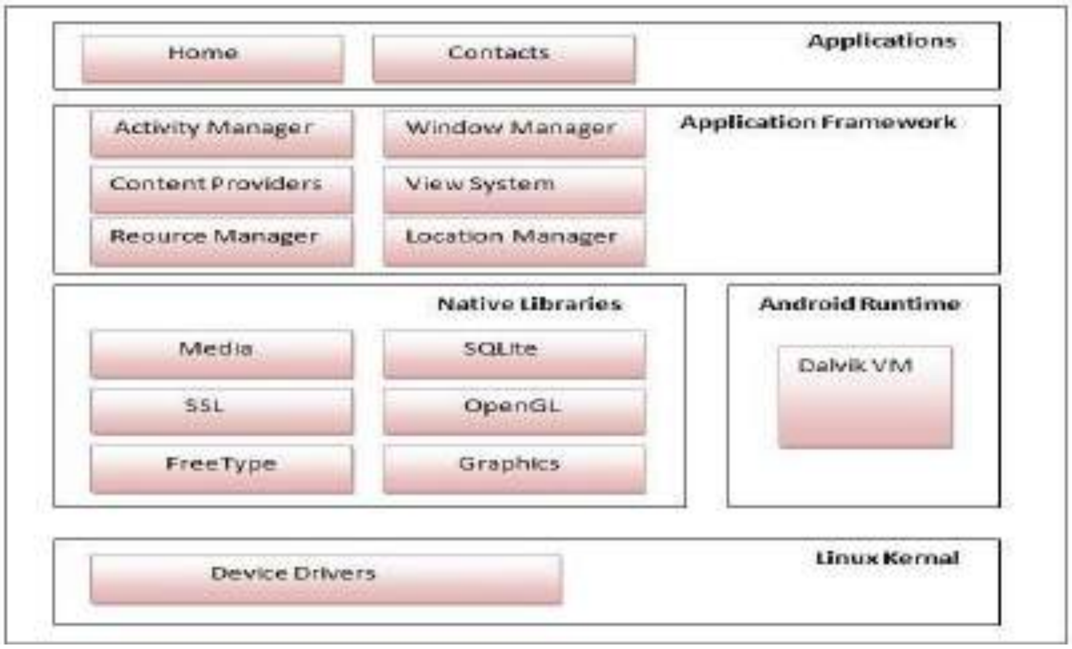
6. **Run your app on an emulator**
   1) In Android Studio, select Run > Run app or click the Run icon in the toolbar.
   2) In the Select Deployment Target window, under Available Emulators, select Pixel API 27 and clickOK
   3) You should see the Hello World app as shown in the following screenshot.

7. **Android Architecture**

Android architecture or Android software stack is categorized into five parts:

1. Linux kernel

2. Native libraries (middleware),
3. Android Runtime
4. Application Framework
5. Applications



1) Linux kernel

It is the heart of android architecture that exists at the root of android architecture. **Linux kernel** is responsiblefor device drivers, power management, memory management, device management and resource access.

2) Native Libraries

On the top of Linux kernel, there are **Native libraries** such as WebKit, OpenGL, FreeType, SQLite, Media, C runtime library (libc) etc. The WebKit library is responsible for browser support. SQLite is for database. FreeType for font support, Media for playing and recording audio and video formats.

3) Android Runtime

In android runtime, there are core libraries and DVM (Dalvik Virtual Machine) which is responsible to run android application. DVM is like JVM but it is optimized for mobile devices. It consumes less memory and provides fast performance.

4) Android Framework

On the top of Native libraries and android runtime, there is android framework. Android framework includes **Android API's** such as UI (User Interface), telephony, resources, locations, Content Providers (data) and package managers. It provides a lot of classes and interfaces for android application development.

5) Applications

On the top of android framework, there are applications. All applications such as home, contact, settings, games, browsers are using android framework that uses android runtime and libraries. Android runtime and native libraries are using Linux kernel.

Assignment 2 : Layout, Activity and Intent

Objectives

- To study how to use Activities, Layouts and Intents in the application.
- To study different layout in an Android.
- To study how to link Activities and interaction between Intent.

Reading

You should read the following topics before starting this exercise:

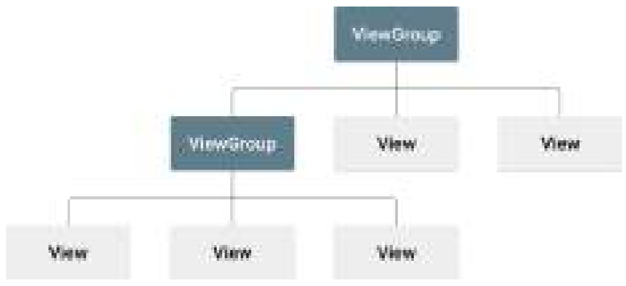
- Android - UI Layouts
- Activity, Activity Lifecycle
- Linking Activities using Intent.

Ready Reference

1) Android - UI Layouts

A layout defines the structure for a user interface in our app, such as in an activity. All elements in thelayout are built using a hierarchy of View and ViewGroup objects. A View usually draws somethingthe user can see and interact with. Whereas a View Group is an invisible container that defines the layout structure for View and other ViewGroup objects.

The View objects are usually called "widgets" and can be one of many subclasses, such as Button or TextView. The ViewGroup objects are usually called "layouts" can be one of many types that provide a different layout structure, such as LinearLayout or ConstraintLayout .



There are number of Layouts provided by

Android which we will use in almost all the Android applications to provide different view, look andfeel and they are...

LinearLayout: Android LinearLayout is a view group that aligns all children in a single direction either vertically or horizontally.	<?xml version="1.0" encoding="utf-8"?> <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android" android:layout_width="fill_parent" android:layout_height="fill_parent" android:orientation="vertical"> <TextView android:id="@+id/text" android:layout_width="wrap_content" android:layout_height="wrap_content" android:text="This is a TextView"/> <Button android:id="@+id/button" android:layout_width="wrap_content" android:layout_height="wrap_content" android:text="This is a Button"/> <EditText android:id="@+id/edtPainText" android:layout_width="wrap_content" android:layout_height="wrap_content" android:text="This is a EditText"/> <!-- More GUI components go here --> </LinearLayout>
RelativeLayout:	<?xml version="1.0" encoding="utf-8"?> <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android" android:layout_width="match_parent" android:layout_height="match_parent"

Android RelativeLayout enables you to specify how child views are positioned relative to each other. The position of each view can be specified as relative to sibling elements or relative to the parent. Using RelativeLayout, you can align two elements by right border, or make one below another, centered in the screen, centered left, and so on. By default, all child views are drawn at the top-left of the layout.	android:paddingLeft="16dp" android:paddingRight="16dp" > <EditText android:id="@+id/name" android:layout_width="match_parent" android:layout_height="wrap_content" android:hint="@string/reminder" /> <Spinner android:id="@+id/dates" android:layout_width="0dp" android:layout_height="wrap_content" **android:layout_below="@id/name"** **android:layout_alignParentLeft="true"** **android:layout_toLeftOf="@+id/times"** /> <Spinner android:id="@id/times" android:layout_width="96dp" android:layout_height="wrap_content" **android:layout_below="@id/name"** **android:layout_alignParentRight="true"** /> <Button android:layout_width="96dp" android:layout_height="wrap_content" **android:layout_below="@id/times"** **android:layout_alignParentRight="true"** android:text="@string/done" /> </RelativeLayout>
TableLayout : Android TableLayout going to be arranged groups of views into rows and columns. You will use the <TableRow> element to build a row in the table. Each row has zero or more cells; each cell can hold one View object,like ImageView, TextView or any other view. For building a row in a table you will use the <TableRow> element. Table row objects are the child views of a table layout. Total width of atable is defined by its parent container. Column can be both stretchable and shrinkable. Ifshrinkable then the width of column can be shrunk to fit the table into its parent object and ifstretchable then it can expand in width to fit any extra space available.	<TableLayout xmlns:android="http://schemas.android.com/apk/res/android" android:layout_width="fill_parent" android:layout_height="fill_parent"> <TableRow android:layout_width="fill_parent" android:layout_height="fill_parent"> <TextView android:text="First Name" android:layout_width="wrap_content" android:layout_height="wrap_content" android:layout_column="1"/> <EditText android:width="200px" android:layout_width="wrap_content" android:layout_height="wrap_content"/> </TableRow> <TableRow android:layout_width="fill_parent" android:layout_height="fill_parent"> <Button android:layout_width="wrap_content" android:layout_height="wrap_content" android:text="Submit" android:id="@+id/button" android:layout_column="2"/> </TableRow> </TableLayout>

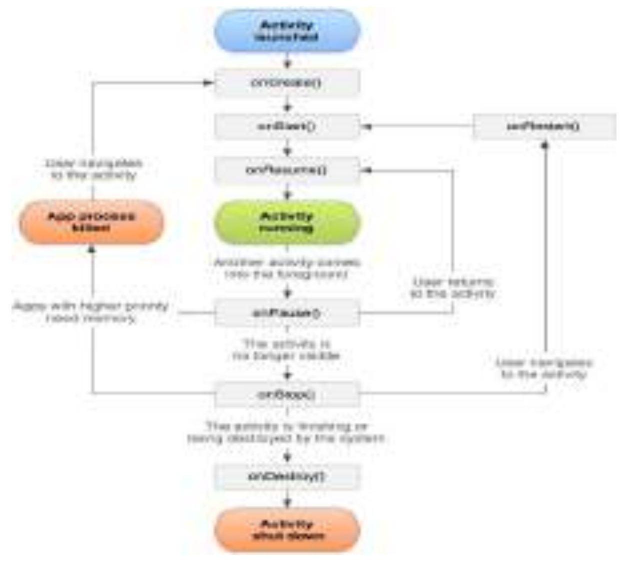
AbsoluteLayout: In Android, an Absolute Layout is a layout used to design the custom layouts. An Absolute Layout	<AbsoluteLayout xmlns:android="http://schemas.android.com/apk/res/android" android:layout_width="fill_parent" android:layout_height="fill_parent"> <Button android:layout_width="100dp"

lets you specify exact locations (x/y coordinates) of its children. Absolute layouts are less flexible and harder to maintain than other types of layouts without absolute positioning	android:layout_height="wrap_content" android:text="OK" android:layout_x="50px" android:layout_y="361px"/> <Button android:layout_width="100dp" android:layout_height="wrap_content" android:text="Cancel" **android:layout_x="225px" android:layout_y="361px"/>** </AbsoluteLayout>
FrameLayout : Frame Layout is designed to block out an area on the screen to display a single item. Generally, FrameLayout should be used to hold a single child view, because it can be difficult to organize child views in a way that's scalable to different screen sizes without the children overlapping each other. You can, however, add multiple children to a FrameLayout and control their position within the FrameLayout by assigning gravity to each child, using the android:layout_gravity attribute.	**ConstraintLayout :** Android Constraint Layout is a ViewGroup (i.e. a view that holds other views) which allows you to create large and complex layouts with a flat view hierarchy, and also allows you to position and size widgets in a very flexible way. It was created to help reduce the nesting of views and also improve the performance of layout files.
ListView : List of scrollable items can be displayed in Android using ListView. It helps you to displaying the data in the form of a scrollable list. Users can then select any list item by clicking on it. ListView is default scrollable so you do not need to use scroll View or anything elsewith ListView. A very common example of ListView is phone contact book	**GridView :** In android GridView is a view group that display items in two or more dimensional scrolling grid (rows and columns), the grid items are not necessarily predetermined but they are automatically inserted to the layout using a ListAdapter. Users can then select any grid item by clicking on it. GridView is default scrollable so you don’t need to use ScrollView or anything else with GridView. An example of GridView is your default Gallery
ScrollView : In Android, a ScrollView is a view group that is used to make vertically scrollable views. A scroll view contains a single direct child only. In order to place multiple views in the scroll view, one needs to make a view group(like LinearLayout) as a direct child and then you can define many views inside it. A ScrollView supports Vertical scrolling only, so in order to create a horizontally scrollable view, HorizontalScrollView is used.	<?xml version="1.0" encoding="utf-8"?> <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android" android:layout_width="match_parent" android:layout_height="match_parent" tools:context=".MainActivity"> <ScrollView android:layout_width="match_parent" android:layout_height="match_parent" tools:layout_editor_absoluteX="0dp" tools:layout_editor_absoluteY="-127dp"> <TextView android:id="@+id/scrolltext" android:layout_width="match_parent" android:layout_height="wrap_content" android:text="@string/scrolltext" android:textColor="@color/green"/> </ScrollView> </androidx.constraintlayout.widget.ConstraintLayout>

2) Activity, Activity Lifecycle

Activity : An activity represents a single screen with a user interface just like window or frame of Java.Almost all activities interact with the user, so the Activity class takes care of creating a window for youin which you can place your UI with setContentView(View).

Activity Lifecycle : Activities in the system are managed as an activity stack. When a new activity is started, it is placed on the top of the stack and becomes the running activity - the previous activity alwaysremains below it in the stack, and will not come to the foreground again until the new activity exits.

Activity Lifecycle Public Methods , Description & Structure	
1) onCreate ( ) : called when activity is first created. This is where you should do all of your normal staticset up: create views, bind data to lists, ect.	 Activity launched onCreate() onStart() onRestart() User navigates to the activity onResume() App process killed Activity running Another activity comes into the foreground User returns to the activity Apps with higher priority need memory onPause() The activity is no longer visible User navigates to the activity onStop() The activity is finishing or being destroyed by the system onDestroy() Activity shut down
2) onStart ( ) : called when activity is becoming visible to the user.	
3) onResume ( ): called when activity will start interacting with the user.	
4) onPause ( ) : called when activity is not visible to the user.	
5) onStop ( ) : called when activity is no longer visible to the user.	
6) onRestart( ) : called after your activity is stopped, prior to start.	
7) onDestroy ( ) : called before the activity is destroyed	

3) Linking Activities using Intent.

Intent is the objects, which is used in android for passing the information among Activities in an Application and from one app to another also.

For example: Intent facilitates you to redirect your activity to another activity on occurrence of anyevent. By calling, startActivity() you can perform this task.

Intent intent = new Intent (getApplicationContext ( ), SecondActivity.class);startActivity (intent);

In the above example, foreground activity is getting redirected to another activity i.e. SecondActivity.java.

getApplicationContext() returns the context for your foreground activity.Intent are of two types: **Explicit Intent and Implicit Intent**

Explicit Intent: Explicit Intents are used to connect the application internally.In Explicit we use the name ofcomponent which will be affected by Intent. For Example: If we know class name then we can navigate the app from One Activity to another activity using Intent. Explicit Intent work internally within an application to perform navigation and data transfer.

Intent intent = new Intent(getApplicationContext(), SecondActivity.class);
startActivity(intent);

Here SecondActivity is the JAVA class name where the activity will now be navigated.

Implicit Intent: In Implicit Intents we do need to specify the name of the component. We just specify the Action which has to be performed and further this action is handled by the component of another application.The basic example of implicit Intent is to open any web page

```
Intent intentObj = new Intent(Intent.ACTION_VIEW);
intentObj.setData(Uri.parse("https://www.google.com"));
startActivity(intentObj);
```

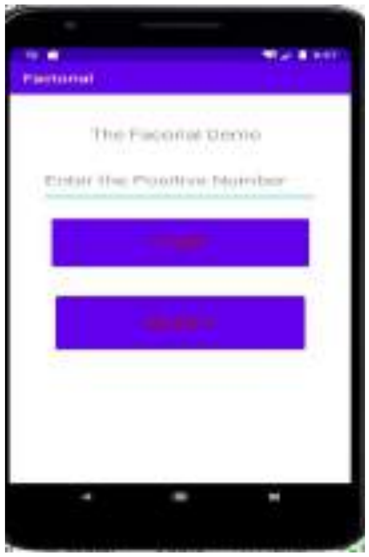
Unlike Explicit Intent you do not use any class name to pass through Intent(). In this example you have just specified an action. Now when we will run this code then Android will automatically start your web browser and it will open google home page.

Example :

Activity_main.xml	MainActivity.java

```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <TextView
        android:id="@+id/textView"
        android:layout_width="404dp"
android:layout_height="36dp"
        android:layout_marginLeft="16dp"
        android:layout_marginTop="52dp"
        android:fontFamily="sans-serif-light"
android:gravity="center"
        android:text="The Facorial Demo"        android:textSize="24sp"
        app:layout_constraintBottom_toTopOf="@+id/edNum"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="1.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
    <EditText
        android:id="@+id/edNum"
        android:layout_width="318dp"        android:layout_height="76dp"
        android:layout_marginTop="24dp"        android:ems="10"
        android:hint="Enter the Positive Number"
        android:inputType="textPersonName"
        android:textColor="#C50F0F"        android:textSize="24sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.494"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView" />
    <Button
        android:id="@+id/btFind"
        android:layout_width="296dp"        android:layout_height="95dp"
        android:layout_marginBottom="348dp"        android:text="Find"
        android:textColor="#AD1111"        android:textSize="24sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/edNum" />
    <Button
        android:id="@+id/btReset"
        android:layout_width="287dp"
android:layout_height="104dp"
        android:text="Reset"
        android:textColor="#AD1111"        android:textSize="24sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/btFind"
        app:layout_constraintVertical_bias="0.151" />
    <TextView
        android:id="@+id/txtDisplay"
        android:layout_width="269dp"        android:layout_height="78dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/btReset"
        app:layout_constraintVertical_bias="0.245" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

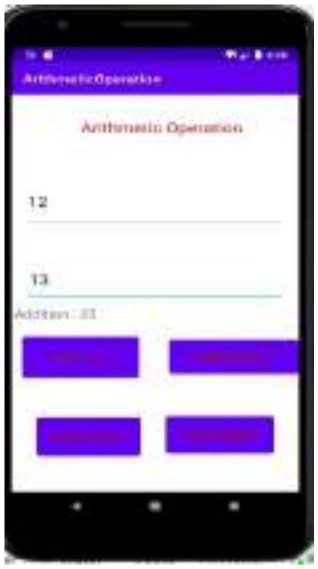
```java
package com.example.factorial;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import  android.widget.TextView;
public class MainActivity extends AppCompatActivity {
    EditText edNum;
    Button btFind,btReset;
   TextView txtDisplay;
   @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        init();
        btFind.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                int num = Integer.parseInt(edNum.getText().toString());
                int f = 1;
                for(int i = 1;i<=num;i++)
                    f = f * i;
                 txtDisplay.setText("Facorial : "+f);
            }
        });
    }
    public void init()
    {
        edNum = findViewById(R.id.edNum);
       btFind = findViewById(R.id.btFind);
       btReset = findViewById(R.id.btReset);
       txtDisplay = findViewById(R.id.txtDisplay);
    }
}
```



Lab Assignments:

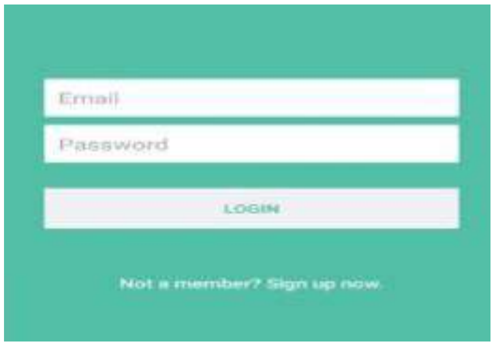
Set A

1. Create a Simple Application Which Shows Life Cycle of Activity.
2. Create a Simple Application Which Send ―Hello‖ message from one activity to another with help of Button(Use Intent).
3. Create a Simple Application, which read a positive number from the user and display its factorial value inanother activity.
4. Create a Simple Application, that performs Arithmetic Operations. (Use constraint layout)



Set B

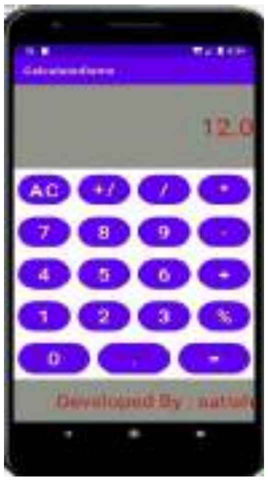
1. Create an Android App, Which reads the Students Details (Name, Surname, Class, Gender, Hobbies, Marks)and Display the all information in another activity in table format on click of Submit button.
2. Create an Android App with Login Screen. On successful login, gives message go to next Activity (WithoutUsing Database & use Table Layout).



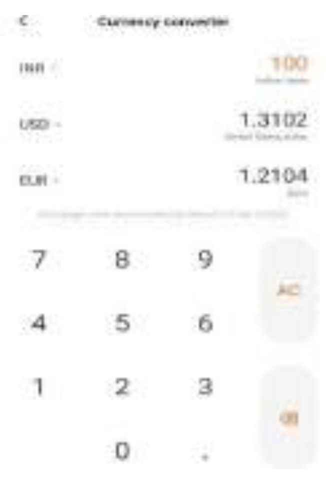
3. Create following Vertical Scroll View Creation in Android.

Set C

1. Create a Simple calculator. (Use Linear Layout)



2) Create an Android Application to convert Indian Rupee(IND) to USD & EUR.



Signature of the instructor: ------------------------- Date:------------------------

Assignment Evaluation

Assignment 3 : Android User Interface and Event Handling

Objectives

- Study how to create user interface in Android.
- Study how to perform event handling

Reading

You should read the following topics before starting this exercise:

- Android - UI Layouts
- Activity, Activity Lifecycle

Control Description	Xml code	Java Code

Android - TextView Control:
 	
In Android, TextView displays text to the user and optionally allows them to edit it programmatically.
	<TextView
	android:id="@+id/textView"
	android:layout_width="wrap_content"
	android:layout_height="wrap_content"
	android:text="Dr. D Y Patil College"
	android:textSize="20sp"
	android:gravity="center_horizontal"
	android:textColor="#f00"
	android:textStyle="bold|italic" />
		TextView textView = findViewById(R.id.textView);
		textView.setText(" Satish Mulgi "); //set text for text view
		textView.setTextColor(Color.***RED***); //set red color for text view
		textView.setTextSize(20); //set 20sp size of text
		//set background color
		textView.setBackgroundColor(Color.***BLACK);***

Android - EditText Control :
In Android, EditText is a standard entry widget in android apps. It is an overlay over TextView that configures itself to be editable. **We often use EditText in our applications in order to provide an input or text field**
	<EditText
	android:id="@+id/simpleEditText"
	android:layout_width="fill_parent"
	android:layout_height="wrap_content"
	android:layout_centerInParent="true"
	android:hint="Enter Your Name Here"
	android:textColorHint="#0f0" />
		EditText editText = findViewById(R.id.simpleEditText);
		editText.setHint("Enter Your Name Here");//display the hint
		//set the green hint color
		simpleEditText.setHintTextColor(Color.*green*(0));
		String editTextValue = editText.getText().toString();
		editText.setText(" Satish Mulgi "); //set text for text view
		editText.setTextSize(20); //set 20sp size of text

Android - Button Control :
In Android, **Button** represents a push button. A Push buttons can be clicked, or pressed by the user to perform an action.
	<Button
	android:id="@+id/simpleButton"
	android:layout_width="wrap_content"
	android:layout_height="wrap_content"
	android:gravity="right|center_vertical"
	android:textColor="#f00"
	android:textSize="25sp"
	android:textStyle="bold|italic"
	android:background="#147D03"
	android:text="ClickMe"/>
		simpleButton = findViewById(R.id.simpleButton);
		//get id of button
		simpleButton.setOnClickListener(new View.OnClickListener() {
		    @Override
		    public void onClick(View view) {
		      Toast.makeText(getApplicationContext(),
		        "Simple Button ", Toast.LENGTH_LONG).show();
		//display the text of button
		    }
		  });

Android - ImageView Control
Android ImageButton Control
In Android, ImageView class is used to display an image file in application.
In Android, ImageButton is used to display a normal button with a custom image in a button
	<ImageView
	android:id="@+id/img"
	android:layout_width="fill_parent"
	android:layout_height="wrap_content"
	android:src="@drawable/lion" />
	<ImageButton
	android:id="@+id/img1"
	android:layout_width="wrap_content"
	android:layout_height="wrap_content"
	android:src="@drawable/home"
	android:background="#000"/>
		ImageView img = findViewById(R.id.img);
		//get the id of second image view
		  img.setOnClickListener(new View.OnClickListener() {
		    @Override
		    public void onClick(View view) {
		  Toast.makeText(getApplicationContext(),
		      "Lion", Toast.LENGTH_LONG).show();
		//display the text on image click event
		    }
		  });

Android - CheckBox Control : In Android, CheckBox is a type of two state button either unchecked or checked in Android. Or you can say it is a type of on/off switch that can be toggled by the users.	`<CheckBox android:id="@+id/cb"`
	`android:layout_width="wrap_content"`
	`android:layout_height="wrap_content"`
	`    android:text="Cricket"`
	`    android:textColor="#44f"`
	`    android:textSize="20sp"/>`
		`CheckBox cb = findViewById(R.id.cb);`
		`// set the current state of a check box`
		`cb.setChecked(true);`
		`//check current state of a check box (true or false)`
		`Boolean checkBoxState = cb.isChecked();`

Android - Switch (On/Off) :

In Android, Switch is a two-state toggle, switch widget that can select between two options. It is used to display checked and unchecked state of a button providing slider control to user. It is basically an off/on button which indicate the current state of Switch. It is commonly used in selecting on/off in Sound, Bluetooth, WiFi etc.

```
<Switch
android:id="@+id/simpleSwitch"
android:layout_width="wrap_conten
t "

 android:layout_height="wrap_conten
t"

   android:checked="true
   "android:text="switch"

android:layout_centerHorizontal="tru
e"

   android:textOn="On"
   android:textOff="Off"
   android:textColor="#f00"
   android:padding="20dp"
   android:gravity="center"
   android:textSize="25sp"
   android:background="#000"/
   >
```

```
 simpleSwitch = findViewById(R.id.simpleSwitch);
      submit = (Button)
      findViewById(R.id.submitButton);

      submit.setOnClickListener(new
         View.OnClickListener() {@Override

         public void onClick(View
            view) {String ss;

            if (simpleSwitch.isChecked())

               ss =
            simpleSwitch.getTextOn().toString();
            else

               ss = simpleSwitch.getTextOff().toString();

Toast.makeText(getApplicationContext(), "Switch :" + ss + "\n",
Toast.LENGTH_LONG).show();
```

ToggleButton (On/Off) :

In Android, ToggleButton is used to display checked and unchecked state

of

a button. ToggleButton basically an off/on button with a lightindicator, which indicate the current state of toggle button. Themost simple exampleof ToggleButton is doing on/off insound, Bluetooth, wifi, hotspot etc.

```
<ToggleButton
 android:id="@+id/tb"

   android:layout_width="wrap_conte
   n

t"

  android:layout_height="wrap_conte
  n

t"

   android:checked="true"
android:textOff="Off State"
android:textOn="On State"
android:textSize="25sp"
android:layout_centerHorizontal="tru
e"
```

```
 tb = findViewById(R.id.tb);

 submit = (Button) findViewById(R.id.submitButton);
 submit.setOnClickListener(new
 View.OnClickListener() {

          @Override

          public void onClick(View view) {

             String status = "ToggleButton : " +
        tb.getText();
        Toast.makeText(getApplicationContext(),

status, Toast.LENGTH_SHORT).show();

// display the current state of toggle button's

          }  });
```

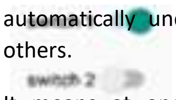
```
    android:textColor="#f00"
    android:padding="40dp"/
    >
```

RadioButton

&RadioGroup

In Android, RadioButton aremainly used together in a RadioGroup.

In RadioGroup checking the one radio button out of several radio button added in it will automatically unchecked all the others.



It means at one time we can checked only one radio button from a group of radio buttons which belong to same radio group.

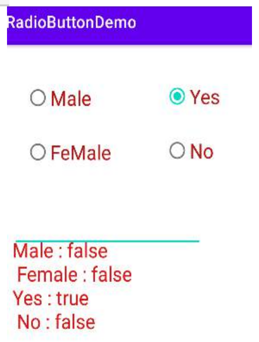
RadioButon is a two state button that can be checked or unchecked. If a radio button is unchecked then a user can check it by simply clicking on it. Once a RadiaButton is checked by user it can’t be unchecked by simply pressing on the same button.

It will automatically unchecked when you press any other RadioButton within

same RadioGroup.

```
<RadioGroup
    android:layout_width="wrap_conte
    n

t"
android:layout_height="wrap_conten
t">

<RadioButton
android:id="@+id/simpleRadioButton
"

    android:layout_width="wrap_conte
    n

t"

   android:layout_height="wrap_conte
   n

t"

    android:checked="true"
android:textSize="25sp"
android:textStyle="bold|italic"
android:padding="20dp"
android:layout_centerHorizontal="tru
e"

    android:text="Male"
    android:textColor="#f00"
    android:background="#000"/
    >

 <RadioButton
   android:id="@+id/female"
   android:la

yout_width="wrap_content"
android:la
yout_height="wrap_content"/>

 </RadioGroup>
```

```
public void showMethod(View
view){male =
findViewById(R.id.male); female =
findViewById(R.id.female);

   str = "Male : "+male.isChecked()+"\n
Female :"+female.isChecked();

   str = str+"\nYes : "+yes.isChecked()+"\n No :
   "+no.isChecked();textView.setText(str.toString());

   Toast toast =
Toast.makeText(getApplicationContext(),str,Toast.LENGTH_LONG);

   toast.setMargin(200,100);
   toast.show();                              }
```



```
Male : false
 Female : false
Yes : true
 No : false
```

Android Toast : Android Toast can be used to display information for the short period of time. Toast class isused to show notification for a particular interval of time. After sometime it disappears. It doesn't block the user interaction. Constants of toast :

public static final int LENGTH_LONG : displays view for the long duration of time.
public static final int LENGTH_SHORT : displays view for the short duration of time.

public static Toast makeText(Context context, CharSequence text, int duration) : makes the toast containing text and duration.

public void show() : displays toast.

public void setMargin (float horizontalMargin, float verticalMargin): changes the horizontal andvertical margin difference.

Example :

```
Toast toast=Toast.makeText(getApplicationContext(),"Hello Friends”, Toast.LENGTH_SHORT);
toast.setMargin(50,50);

toast.show();
```

RatingBar : RatingBar is used to get the rating from the app user. A user can simply touch, drag or click onthe stars to set the rating value. The value of rating always returns a floating point number which may be 1.0,2.5, 4.5 etc.

1) The **getRating()** method of android RatingBar class returns the rating number.

```
RatingBar simpleRatingBar = (RatingBar) findViewById(R.id.simpleRatingBar); // initiate a rating bar
Float ratingNumber = simpleRatingBar.getRating(); // get rating number from a rating bar
```

2) **numStars**: numStars attribute is used to set the number of stars (or rating items) to be displayed in a rating bar. By default a rating bar shows five stars but we can change it using numStars attribute. numStars must have a integer number like 1,2 etc.

```
<RatingBar

        android:id="@+id/simpleRatingBa
        r"
        android:layout_width="wrap_cont
        ent"
        android:layout_height="wrap_cont
        ent"android:numStars="7" />
```

Java code : simpleRatingBar.setNumStars(7); // set total number of stars

3) **getNumStars():** is used to get the number of stars of a RatingBar.

```
int numberOfStars = simpleRatingBar.getNumStars(); // get total number of stars of rating bar
```

4) **rating:** Rating attribute set the default rating of a rating bar. It must be a floating point number.

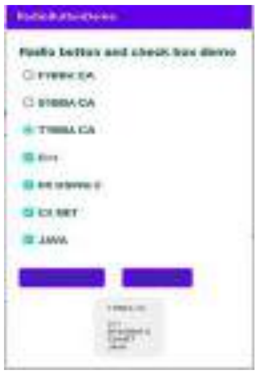
```
<RatingBar
```

```
android:id="@+id/simpleRatingBa
r"
android:layout_width="wrap_cont
ent"
android:layout_height="wrap_cont
ent"android:rating="3.5" />
```

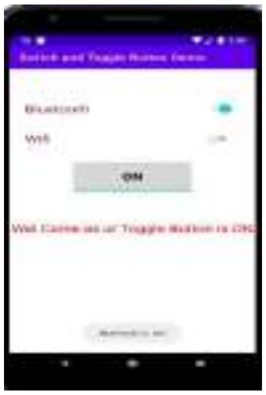
Java code : simpleRatingBar.setRating((float) 3.5); // set default rating

Lab
Assignments:
Set A

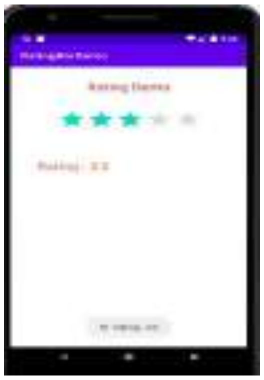
1. Create an Android Application that will change color of the College Name(Use TextView) on click of Push Button and change the font size, font style of text view using xml.
2. Create an Android Application to accept two numbers(Use PainText) and create two buttons(power and Average). Display the result on the next activity on Button click
3. Design Following Screens Using RadioButtons & CheckBoxes.Display the selected text using Toast.



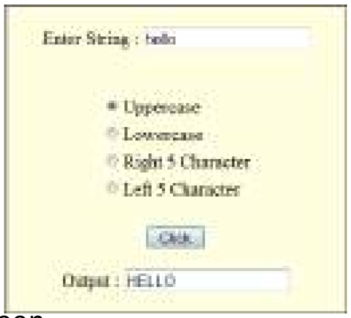
4. Create an Android Application that Demonstrate Switch and Toggle Button.



1. Create an Android Application that Demonstrate RatingBar and Display the number of stars selected onToast and TextView



2. Create an Android Application to perform following string operation according to user selectionof radio button.

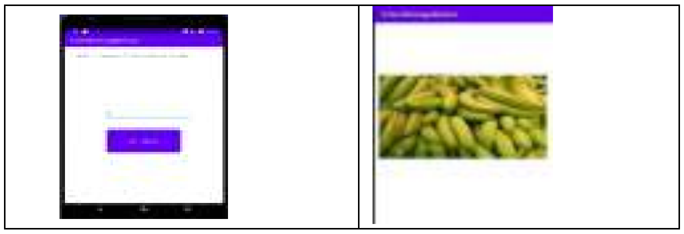


Enter String : hello

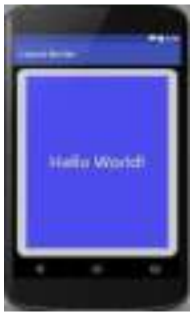
- ● Uppercase
- ○ Lowercase
- ○ Right 5 Character
- ○ Left 5 Character

Output : HELLO

3. Write an Android Application to Change the ImageDisplayed on the Screen

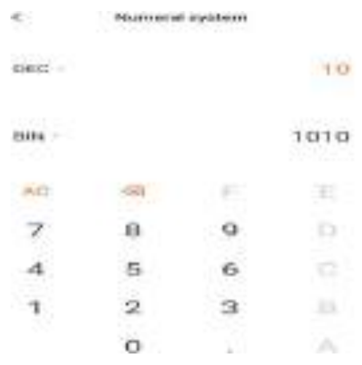


4. Design following-add a border to an Android Layout .



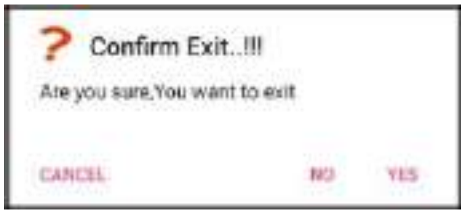
1. Construct image switcher using setFactory().

2. Create an Android Application to convert Decimal number to Binary equivalent.



Assignment 4 : Android TimePicker, DatePicker, Alert Dailog

Android - Alert Dialog: Alert Dialog in an android UI prompts a small window to make decision on mobile screen. Sometimes before making a decision, it is required to give an alert to the user without moving to nextactivity. For example you have seen this type of alert when you try to exit the App and App ask you to confirm exiting.



AlertDialog.Builder is used to create an interface for Alert Dialog in Android for setting like alert title, message, image, button, button onclick functionality etc.

AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder(this);

1) **setTitle(CharSequence title)** – This component is used to set the title of the alert dialog. It is optional component.

alertDialogBuilder.setTitle("Confirm Exit..!!!"); // Setting Alert Dialog Title

2) **setIcon(Drawable icon)** – This component add icon before the title. You will need to save image in drawable icon.

alertDialogBuilder.setIcon(R.drawable.question); // Icon Of Alert Dialog

3) **setMessage(CharSequence message)** – This component displays the required messagein the alert dialog.

alertDialogBuilder.setMessage("Are you sure,You want to exit");// Setting Alert Dialog Message

4) **setCancelable(boolean cancelable)** – This component has boolean value i.e true/false.If set to false it allows to cancel the dialog box by clicking on area outside the dialog else it allows.

alertDialogBuilder.setCancelable(false);

5) **setPositiveButton(CharSequence text, DialogInterface.OnClickListener listener)** – This component add positive button and further with this user confirm he wants the alert dialog question to happen.

```
alertDialogBuilder.setPositiveButton("Ye
    s", new
    DialogInterface.OnClickListener() {

 @Override

     public void onClick(DialogInterface arg0, int
  arg1) {finish ();

} });
```

6) **setNegativeButton(CharSequence text, DialogInterface.OnClickListener listener)** – This component add negative button and further with this user confirm he doesn't want the alert dialog question to happen.

```
alertDialogBuilder.setNegativeButton("No",
      new
      DialogInterface.OnClickListener() {
      @Override

       public void onClick(DialogInterface dialog, int which) {
Toast.makeText(MainActivity.this,"You clicked over No",Toast.LENGTH_SHORT).show();

} });
```

7) **setNeutralButton(CharSequence text, DialogInterface.OnClickListener listener)** – This component simply add a new button and on this button developer can set any other onclick functionality like cancel button on alert dialog.

```
alertDialogBuilder.setNeutralButton("Cancel
    ",new DialogInterface.OnClickListener() {
    @Override

    public void onClick(DialogInterface dialog, int which) {
Toast.makeText(getApplicationContext(),"You          clicked          on

   Cancel",Toast.LENGTH_SHORT).show();

 }

 });
```

8) alertDialogBuilder.create();
9) alertDialogBuilder.show();

Example :

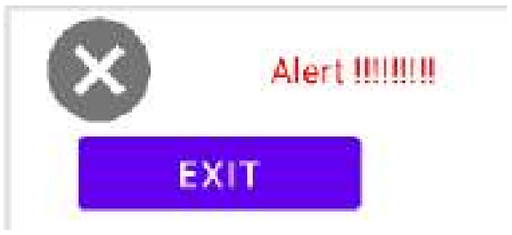
```
final AlertDialog.Builder adb = new AlertDialog.Builder(this);
adb.setTitle("Confirm Exit !");

adb.setMessage("are you sure, you want to
Exit ?");adb.setCancelable(true);

adb.setPositiveButton("OK", new
    DialogInterface.OnClickListener() {@Override

    public void onClick(DialogInterface dialog, int which) {
        finish();

  }

});

adb.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {
    @Override

    public void onClick(DialogInterface dialog, int which) {
        Toast.makeText(getApplicationContext(),"You pressed Cancel  ",Toast.LENGTH_LONG).show();

  }

});

adb.setNeutralButton("Retry", new
    DialogInterface.OnClickListener() {@Override

    public void onClick(DialogInterface dialog, int which) {
        Toast.makeText(getApplicationContext(),"You Presses cancel ",Toast.LENGTH_LONG).show();

  }

});

button.setOnClickListener(new
        View.OnClickListener() {@Override

        public void onClick(View
            v) {adb.create();
            adb.show();

    }

});
```

Custom Alert Dialog: The custom dialog uses DIALOG to create custom alert in android studio. Dialog display a small window that is a popup which draws the user attention over the activity before they continuemoving forward. The following are the steps to create custom alert Dialog.

Step 1: Create a new project and name it **CustomAlertDialogDemo**.

Step 2: Open res -> layout -> activity_main.xml and create XML for button and textview. On button click custom alert dialog will appear & textview is for asking user to click on button.

Step 3 : Now create a layout file name as custom.xml . In this file we are going to define the XML for custom dialog appearance. For example a textbox , imageview and a button.



Step 4 : Now open app -> java -> package -> MainActivity.java and add the below code. In this code onclick is added over the button click and Dialog is used to create the custom alert dialog.

```
button.setOnClickListener(new View.OnClickListener() {

        @Override

        public void onClick(View arg0) {

            final Dialog dialog = new Dialog(context); // custom dialog
            dialog.setContentView(R.layout.custom);

            // if button is clicked, close the custom dialog
           dialog.setOnClickListener(new View.OnClickListener() {

                @Override

                public void onClick(View
                    v) {dialog.dismiss();

Toast.makeText(getApplicationContext(),"Dismissed..!!",Toast.LENGTH_SHORT).show();

            }

        });
```

TimePicker: In Android, TimePicker is a widget used for selecting the time of the day in either AM/PM mode or 24 hours mode. The displayed time consist of hours, minutes and clock format. Ifwe need to show this view as a Dialog then we have to use a TimePickerDialog class.

```
<TimePicker
    android:id="@+id/simpleTimePicker“
    android:layout_width="wrap_content"
    android:layout_height="wrap_content“
    android:timePickerMode="spinner"/>  or
    clock
```

Some Methods of TimePicker:

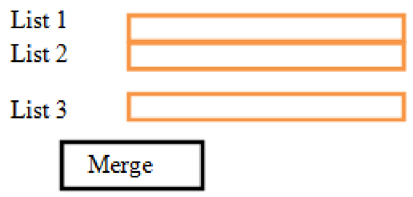
1) **setCurrentHour(Integer currentHour):** This method is used to set the current hours ina time picker.
2) **setHour(Integer hour):** setCurrentHour() method was deprecated in API level 23. From apilevel 23 we have to use setHour(Integer hour). In this method there is only one parameter ofinteger type which is used to set the value for hours.
3) **setCurrentMinute(Integer currentMinute):** This method is used to set the current minutesin a time picker.
4) **getCurrentHour():** getCurrentHour() method was deprecated in API level 23. From api level 23 you have to use getHour(). This method returns an integer value.
5) **getCurrentMinute():**This method is used to get the current minutes from a time picker.
6) **getMinute():** getCurrentMinute() method was deprecated in API level 23. From api level 23we have to use getMinute(). This method returns an integer value.
7) **setIs24HourView(Boolean is24HourView):**This method is used to set the mode of the Time picker either 24 hour mode or AM/PM mode. In this method we set a Boolean value either true or false. True value indicate 24 hour mode and false value indicate AM/PM mode.
8) **is24HourView():**This method is used to check the current mode of the time picker. This method returns true if its 24 hour mode or false if AM/PM mode is set.

```
TimePicker  simpleTimePicker  =  (TimePicker)findViewById(R.id.simpleTimePicker);
 // initiate a time picker
 simpleTimePicker.setOnTimeChangedListener(new     TimePicker.OnTimeChangedListener()
 {
 @Override
public void onTimeChanged(TimePicker view, int hourOfDay, int
                    minute) {time.setText(hourOfDay + ":" + minute);
```

Set A

```
}});
```

1. Create an Android application that demonstrate the Alert Dialog.
2. Write an Android code to merge given two Array/List



3. . Create an Android application that demonstrate the Custom Alert Dialog.
4. Create an Android Application to find the factorial of a number and Display the Result on AlertBox.

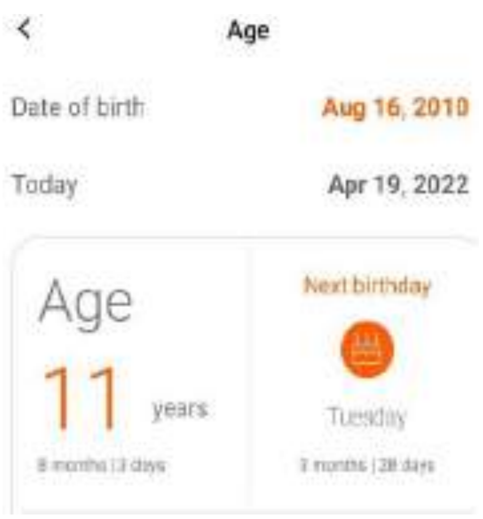
Set B

Set C

1. Develop an Android application that create custom Alert Dialog containing Friends Name andonClick of Friend Name Button greet accordingly.
2. Create an Android Application that Demonstrate DatePicker and DatePickerDailog.

3. Create an Android Application that Demonstrate TimePicker and TimePickerDailog.

1) Create a Simple Android [illegible]on to calculate age of a person. (UseTable Layout)



Assignment 5 : Android Adapter and Menu

In Android, Adapter is a bridge between UI component and data source that helps us to fill data in UI component. It holds the data and send the data to an Adapter view then view can takes the data from the adapter view and shows the data on different views like as ListView, GridView, Spinner etc. For more customization in Views we uses the base adapter or custom adapters. To fill data in a list or a grid we need to implement Adapter.

There are the some commonly used Adapter in Android used to fill the data in the UI components.

1) **BaseAdapter** – It is parent adapter for all other adapters
2) **ArrayAdapter** – It is used whenever we have a list of single items which is backed by anarray.
3) **Custom ArrayAdapter** – It is used whenever we need to display a custom list.
4) **SimpleAdapter** – It is an easy adapter to map static data to views defined in your XML file
5) **Custom SimpleAdapter** – It is used whenever we need to display a customized list andneeded to access the child items of the list or grid

List View: It is widely used in android applications. A very common example of ListView is your phone contact book, where you have a list of your contacts displayed in a ListView and if you clickon it then user information is displayed.

Adapter: To fill the data in a ListView we simply use adapters. List items are automatically inserted to a list using an Adapter that pulls the content from a source such as an arraylist, array or database. Listview is present inside Containers. From there you can drag and drop on virtual mobile screen tocreate it. Alternatively you can also XML code to create it.

divider: This is a drawable or color to draw between different list items.

dividerHeight: This specify the height of the divider between list items. This could be in dp(densitypixel), sp(scale independent pixel) or px(pixel).

XML Code :

```
 <ListView
     android:id="@+id/simpleListView“
     android:layout_width="fill_parent"
     android:layout_height="wrap_conten
     t" android:divider="#f00"
     android:dividerHeight="1dp"
     android:listSelector="#0f0"/>
```

Java code :

```
    ListView simpleList;

   String countryList[] = {"Rajesh", "Ramesh", "Suresh", "Satish", "Rakesh", "Dinesh"};
@Override

protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);

 simpleList = (ListView)findViewById(R.id.simpleListView);
ArrayAdapter<String> arrayAdapter = new ArrayAdapter<String>(this,
R.layout.activity_listview, R.id.textView, countryList);
simpleList.setAdapter(arrayAdapter);

}
```

GridView : In android GridView is a view group that display items in two dimensional scrolling grid (rows and columns), the grid items are not necessarily predetermined but they are automatically inserted to the layout using a Adapter. Users can then select any grid item by clicking on it. GridView is default scrollable. An example of GridView is your default Gallery, where you have number of images displayed using grid. To fill the data in a GridView we simply use adapter and grid items are automatically inserted to a GridView using an Adapter which pulls the content from a source such as an arraylist, array or database

XML Code :

```
<GridView

    android:id="@+id/simpleGridView" android:layout_width="fill_parent"
    android:layout_height="wrap_content" android:numColumns="3“
    android:verticalSpacing="50dp“

    android:horizontalSpacing="50
    dp“
    android:columnWidth="80dp
    " android:listSelector="#0f0"

/>
```

Spinner :

In Android, Spinner provides a quick way to select one value from a set of values. Android spinnersare nothing but the drop down-list seen in other programming languages. In a default state,a spinner shows its currently selected value. It provides a easy way to select a value from a list of values.

XML Code :

```
<Spinner
  android:id="@+id/simpleSpinner"
  android:layout_width="wrap_cont
  ent"
  android:layout_height="wrap_conte
  nt”
  android:layout_centerHorizontal="tr
  ue"
  android:layout_marginTop="100dp"
  />
```

Java code :

```
String[]   bankNames={"BOI","SBI","HDFC","PNB","OBC"};

Spinner spin = (Spinner) findViewById(R.id.simpleSpinner);
ArrayAdapter aa = new
ArrayAdapter(this,android.R.layout.simple_spinner_item,bankNam
es);

 aa.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
 spin.setAdapter(aa);

 spin.setOnItemSelectedListener(this);

public void onItemSelected(AdapterView<?> arg0, View arg1, int position,long id) {
Toast.makeText(getApplicationContext(), bankNames[position],
Toast.LENGTH_LONG).show(); }
```

Menu in Android : In android, there are three types of Menus available to define a set of options and actions in our android applications.

1) **Android Options Menu :** The options menu is the primary collection of menu items for anactivity. It's where you should place actions that have a overall impact on the app, such as Search, Compose Email and Settings.
2) **Android Context Menu** : A context menu is a floating menu that appears when the user performs a long-click on an element. It provides actions that affect the selected content or context frame.

3) **Android Popup Menu** : A popup menu displays a list of items in a vertical list that is anchored(sticked) to the view that invoked the menu. It's good for providing an overflow ofactions that relate to specific content or to provide options for a second part of a command.

How to create a Menu?

For all menu types, Android provides a standard XML format to define menu items. you should define a menu and all its items in an XML menu resource. You can then inflate the menu resource i.e load the XML files as a Menu object in your activity.

1) A new menu directory would be made under res directory.
2) Add menu_file.xml file in menu directory by right clicking on **menu --> New --> Menuresource file**.
3) Give the name as **menu_file.xml** and click on Ok.
4) The **menu_file.xml** file contains the following tags:
   a) **<menu> :** It defines a Menu, which is a container for menu items. A <menu> element must be the root node for the file and can hold one or more <item> and <group> elements.
   b) **<item>:** It creates a MenuItem, which represents a single item in a menu. This elementmay contain a nested <menu> element in order to create a submenu.
   c) **<group> :**It is an optional, invisible container for <item> elements. It allows you to categorize menu items so they share properties such as active state and visibility.

menu_file.xml

```
<?xml version="1.0"  encoding="utf-8"?>

 <menu

   xmlns:android="http://schemas.android.com/apk/res/android">

 <item

         android:id="@+id/item1"
         android:title="item1“
         android:icon="@drawable/item"
         >

<!-- "item" submenu -->

         <menu>

                 <item
```

```
                android:id="@+id/a"
                android:title="subitem a"
                android:icon="@drawable/subitem_a"/>
            <item
                android:id="@+id/b“
                 android:title="subitem b“
                 android:icon="@drawable/subitem_b"
        </menu>  />
    </item>
    </menu>
```

Making an Option Menu

- To make an option menu, we need to Override **onCreateOptionsMenu()** method asfollows:

```
@Override

public boolean onCreateOptionsMenu(Menu menu)

{

    MenuInflater        inflater        =
    getMenuInflater();
    inflater.inflate(R.menu.menu_file,
    menu); return true;

}
```

MenuInflater inflater = getMenuInflater();

This gives a MenuInflater object that will be used to inflate(convert our XML file into JavaObject) the menu_file.xml file.

inflater.inflate(R.menu.menu_file, menu);

inflate() method is used to inflate the menu_file.xml file.

Handling Click Events

- When the user selects an item from the options menu, the system calls youractivity's onOptionsItemSelected() method.
- This method passes the MenuItem selected.
- You can identify the item by calling getItemId() method, which returns the unique ID forthe menu item (defined by the android:id attribute in the menu resource).
- You can match this ID against known menu items to perform the appropriate

```
action.@Override

public boolean onOptionsItemSelected(MenuItem item) {

    //Handle item

     selection switch

     (item.getItemId()) {

         case R.id.i1:
```

```
        //perform any action; return

        true;case R.id.a:

        //perform any action; return

         true;case R.id.b:

        //perform any action; return true;

        default:  return  super.onOptionsItemSelected(item);

}

 }
```

Making Contextual Menu

- To make a floating context menu, you need to follow the following steps:
- Register the View to which the context menu should be associated bycalling registerForContextMenu() and pass it the View.
- If your activity uses a ListView or GridView and you want each item to provide the same context menu, you can register yout all items for a context menu by passing the ListView orGridView object to registerForContextMenu() method.
- Implement the onCreateContextMenu() method in your Activity.
- When the registered view receives a long-click event, the system callsyour onCreateContextMenu() method.
- This is where you define the menu items, usually by inflating a menu

resource.For example:

```
@Override
public void onCreateContextMenu(ContextMenu menu, View v, ContextMenuInfo menuInfo) {

        super.onCreateContextMenu(menu, v, menuInfo);

        MenuInflater inflater =

         getMenuInflater();
```

```
            inflater.inflate(R.menu.menu_file,

            menu);

}
```

MenuInflater allows you to inflate the **menu_file.xml** file from a menu resource. The method parameters include the View that the user selected and aContextMenu. **ContextMenuInfo** object provides additional information about the item selected. If your activity has several views such thateach provide a different context menu, you might use these parameters to determine which context menu to inflate.

Handling Click Events

```
@Override

public boolean onContextItemSelected(MenuItem item)

{

            switch (item.getItemId())

            {

                    case R.id.i1:

                    //Perform any action; return
                    true;case R.id.a:

                    //Perform any action; return
                     true;case R.id.b:

                    //Perform any action; return true;

                    default:  return  super.onContextItemSelected(item);

            }

 }
```

Making Popup Menu

- If you have define your menu_file.xml file in XML, here's how you can show the popupmenu:
- Make an object of PopupMenu, whose constuctor takes the current application Context andthe View to which the menu should be anchored.

- Use MenuInflater to inflate your menu resource into the Menu object returnedby PopupMenu.getMenu()
- Call

PopupMenu.show()For example,

```
<Button android:id="@+id/button" android:layout_width="wrap_content"
android:layout_height="wrap_content" android:text="Button"

    android:onClick="pop"  />
```

The activity can then show the popup menu like this:

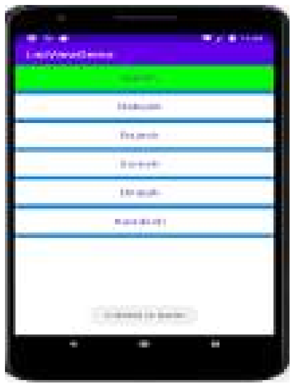
```
public void pop(View v)

{         PopupMenu popup = new PopupMenu(this,v);

    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.menu_file,popup.getMenu()); popup.show();

 }
```

Listener

```
b1.setOnClickListener(new View.OnClickListener() {
   @Override

   public void onClick(View v) {

      final PopupMenu popup = new
      PopupMenu(getApplicationContext(),v);MenuInflater inflater =
      getMenuInflater();

inflater.inflate(R.menu.menu_file,popup.getMenu());
      popup.show();

 popup.setOnMenuItemClickListener(new PopupMenu.OnMenuItemClickListener() {
         @Override

         public boolean onMenuItemClick(MenuItem item) {

  Toast.makeText(getApplicationContext(),"selected
"+item.getTitle(),Toast.LENGTH_LONG).show();

            return true;

      }

  });
```

Set A

1. Create an Android Application that Demonstrate ListView and Onclick of List Display the Toast.

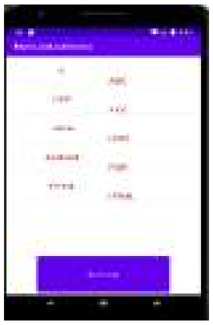


2. Create an Android Application that Demonstrate GridView and Onclick of Item Display the Toast.

3. Create an Android Application that Demonstrate GridView with Images

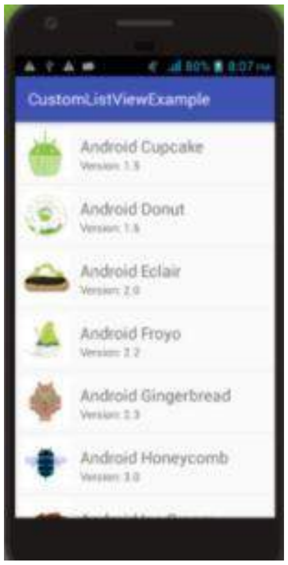
Set B

1. Create an Android Application that Demonstrate Custom ListView which shows the BookName and Author Name



2. Create a Custom ListView in Android Application

ContextMenu Demo – Please Long Click



CustomListViewExample

Android Cupcake
Version: 1.5

Android Donut
Version: 1.6

Android Eclair
Version: 2.0

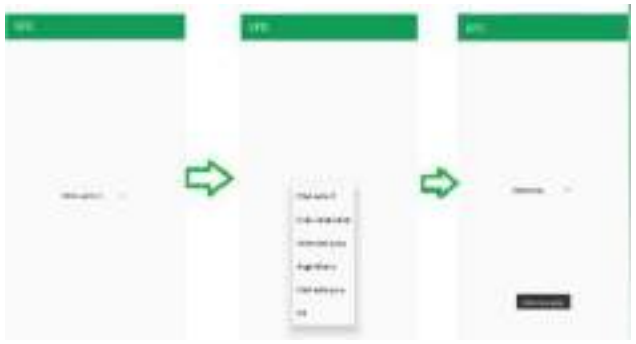
Android Froyo
Version: 2.2

Android Gingerbread
Version: 2.3

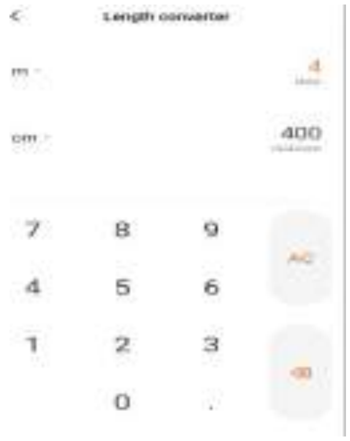
Android Honeycomb
Version: 3.0

3. Create an Android Application that Demonstrate ContextMenu.**et**
   1. Create the following layout using spinner

4. C



2. Create an Android Application to demonstrate Length converter.



Signature of the instructor: ------------------------- Date:-------------------------

Assignment Evaluation

- To provide a way to improve application performance through parallelism.
- To ensure that the application remains active in the background so that the user can operatemultiple applications at the same time.

Thread and Notification:

A thread is a thread of execution in a program. The Java Virtual Machine allows an application to have multiple threads of execution running concurrently. When an application is launched, the system creates a thread of execution for the application, called "main." This thread is very important because it is in charge of dispatching events to the appropriate user interface widgets, including drawing events.

Worker threads

Worker threads are background threads. They are the threads that are created separately, other thanthe UI thread. Since blocking the UI thread is restricted according to the rule, user should run the child processes and tasks in worker threads.

Example

```
public void onClick(View v) {
   new Thread(new
   Runnable() {

      public void run() {

         Bitmap b = loadImageFromNetwork("http://example.com/image.png");
         mImageView.setImageBitmap(b);

      }

   }).start();

}
```

In the above example code, the download operation is handled by a second thread other than the UI thread. But the program violates the second rule. The imageView from UI thread is manipulating from this worker thread. In the above example code, the download operation is handled by a second thread other than the UI thread. But the program violates the second rule. The imageView from UI thread is manipulating from this worker thread.

According to the second rule, UI could not be accessed from outside the UI thread. Solutionfor such a restriction is runOnUiThread(Runnable) method. The main or UI thread can be accessed from other threads using runOnUiThread(Runnable) method. As a result, the specified runnable action passed through this method will run on the UI thread. The action will execute immediately, ifthe current thread is in the UI itself. Else the action will be posted to the event queue.

Example:

```
MainClassName.this.runOnUiThread(new Runnable() {
          public void run() {
                textViewObject.setText("Set this " + value from worker thread + "!");
            }
        });
```

Handlers & Runnable:

A handler is basically a message queue. You post a message to it, and it will eventually process it by calling its run method and passing the message to it. Since these run calls will always occur in the order of messages received on the same thread, it allows you to serialize events.

Uses of Handler:

Handler has two main uses

1. Scehduling of messages and runnables that need to be executed in the future.
2. Enqueueing actions that need to be performed in the background thread.

Handler is a class fundamental to how we do threading in android, at the infrastructure level.It works hand in hand with the Looper. Together they underpin everything that the main thread doesand including the invocation of the Activity lifecycle methods.

Looper will take care of dispatching work on its message-loop thread. On the other hand

Handler will serve two roles:

1. First it will provide an interface to submit messages to its Looper queue.
2. Secondly it will implement the callback for processing those messages when they aredispatched by the Looper.

Example:

```
public void onClick(View v) {

        nstoploop = true;

        new Thread(new
           Runnable() {@Override

           public void run()
              { while
              (nstoploop){

                 try

                 {

                    Thread.sleep(500);
                    count++;


                 }catch
                    (InterruptedException e){
```

```
                    Log.i(TAG,e.getMessage()
                    );

                }

                handler.post(new Runnable() {
                    @Override

                    public void run() {
                        tv.setText(" "+count);

                }

            });

             }

          }

      }).start();

  }

});
```

We can’t touch background thread to main thread directly so handler is going to collect all eventswhich are available in main thread in a queue and posses this queue to looper class.

In android Handler is mainly used to update the main thread from background thread or other thanmain thread. There are two methods are in handler.

- **Post()** − it going to post message from background thread to main thread using looper.
- **sendmessage()** − if you want to organize what you have sent to ui (message from backgroundthread) or ui functions. you should use sendMessage().

AsynTask:

Android AsyncTask going to do background operation on background thread and update onmain thread. In android we can’t directly touch background thread to main thread in android development. asynctask help us to make communication between background thread to main thread.

Android AsyncTask is an abstract class provided by Android which gives us the liberty to perform heavy tasks in the background and keep the UI thread light thus making the application more responsive.

Android application runs on a single thread when launched. Due to this single thread model tasks that take longer time to fetch the response can make the application non-responsive. To avoidthis we use android AsyncTask to perform the heavy tasks in background on a dedicated thread andpassing the results back to the UI thread. Hence use of AsyncTask in android application keeps the UI thread responsive at all times.

Methods of AsyncTask

- **onPreExecute()** –
  Before doing background operation we should show something on screen like progressbar or any animation to user. we can directly comminicate background operation using on doInBackground().

- **doInBackground(Params)** –
  In this method we have to do background operation on background thread. Operationsin this method should not touch on any mainthread activities or fragments.

- **onProgressUpdate(Progress…)** –
  While doing background operation, if you want to update some information on UI, we can use this method.

- **onPostExecute(Result)** –
  In this method we can update ui of background operation result.

Generic Types in Async Task

- **TypeOfVarArgParams** –
  It contains information about what type of params used for execution.

- **ProgressValue** –
  It contains information about progress units. While doing background operation wecan update information on ui using onProgressUpdate().

- **ResultValue** –
  It contains information about result type.

Example:

MainActivity.java

```
public class MainActivity extends AppCompatActivity {
Button b1,b2;

TextView tv;
Boolean
stoploop;int
count = 0;

String TAG = "Thread";
private MyAsynctask
async;


   @Override

   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);

      b1 =
      (Button)findViewById(R.id.btn1);b2
      = (Button)findViewById(R.id.btn2);

      tv = (TextView)findViewById(R.id.textView);


      b1.setOnClickListener(new View.OnClickListener() {
         @Override

         public void onClick(View
           v) {stoploop = true;
```

```
                async = new
                MyAsynctask();
                async.execute(count);

            }

        });

            b2.setOnClickListener(new View.OnClickListener() {
                @Override

                public void onClick(View
                v) {stoploop = false;

            }

        });

         }

         private class MyAsynctask extends AsyncTask<Integer,Integer ,Integer>

         {

            private int ccount;


            @Override

            protected void
                onPreExecute() {
                super.onPreExecute();
                ccount=0;

            }


            @Override

            protected  Integer  doInBackground(Integer...  integers)
                {ccount = integers[0];

                while(stoploo
                    p){ try

                    {

                        Thread.sleep(1000);
                        ccount++;
                        publishProgress(ccoun
                        t);

                    }
                    catch (InterruptedException e){
                        Log.i(TAG," "+e.getMessage());

                    }

                }

                return ccount;

            }


            @Override

            protected void onProgressUpdate(Integer...
                values) {super.onProgressUpdate(values);
                tv.setText(""+values[0]);

            }


            @Override

            protected void onPostExecute(Integer
                integer) {super.onPostExecute(integer);
                tv.setText(""+integer);

                count = integer;

            }

         }

     }
```

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:app="http://schemas.android.com/apk/res-

    auto"

    xmlns:tools="http://schemas.android.com/tools"

    android:layout_width="match_parent"

    android:layout_height="match_parent"

    tools:context=".MainActivity">
```

```
<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_cont
    ent"

    android:layout_height="wrap_content"
    android:text="Hello World!"
    app:layout_constraintBottom_toBottomOf="parent
    " app:layout_constraintHorizontal_bias="0.58"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.102" />


<Button
    android:id="@+id/btn1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Start Thread"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.6"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textVie
    w" app:layout_constraintVertical_bias="0.116" />


<Button
    android:id="@+id/btn2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Stop Thread"
    app:layout_constraintBottom_toBottomOf="par
```

```
        ent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.6"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/btn
        1"

        app:layout_constraintVertical_bias="0.145"  />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Broadcast Receiver :

Broadcast in android is the system-wide events that can occur when the device starts, when amessage is received on the device or when incoming calls are received, or when a device goes to airplane mode, etc. Broadcast Receivers are used to respond to these system-wide events. BroadcastReceivers allow us to register for the system and application events, and when that event happens, then the register receivers get notified. There are mainly two types of Broadcast Receivers:

- **Static Broadcast Receivers:**
  These types of Receivers are declared in the manifest file and works even if the app is closed.
- **Dynamic Broadcast Receivers:**
  These types of receivers work only if the app is active or minimized.

Creating the Broadcast Receiver:

```
class AirplaneModeChangeReceiver:BroadcastReceiver() {
    override fun onReceive(context: Context?, intent: Intent?)
    {

        // logic of the code needs to be written here

    }

}
```

Registering a BroadcastReceiver:

```
IntentFilter(Intent.ACTION_AIRPLANE_MODE_CHANGED).also   {

            // receiver is the broadcast receiver that we have registered
```

```
                    // and it is the intent filter that we have
                    createdregisterReceiver(receiver,it)

        }
```

Intent	Description Of Event
android.intent.action.BATTERY_LOW :	Indicates low battery condition onthe device.
android.intent.action.BOOT_COMPLETED	This is broadcast once after thesystem has finished booting
android.intent.action.CALL	To perform a call to someonespecified by the data

Intent	Description Of Event
android.intent.action.DATE_CHANGED	Indicates that the date has changed
android.intent.action.REBOOT	Indicates that the device has been areboot
android.net.conn.CONNECTIVITY_CHANGE	The mobile network or wifi connection is changed(or reset)
android.intent. ACTION_AIRPLANE_MODE_CHANGED	This indicates that airplane modehas been switched on or off.

Example:

MainActivity.Java

```
public class MainActivity extends AppCompatActivity {
   private EditText txtPhone;

   private Button
   btn;
   @Override

   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);

      txtPhone =
      (EditText)findViewById(R.id.mblTxt);btn =
      (Button)findViewById(R.id.btnCall);

      btn.setOnClickListener(new View.OnClickListener()
         { @Override

         public void onClick(View
            v) {callPhoneNumber();


     }

  });


   }


   @Override

   public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions,
@NonNull int[] grantResults) {

      if(requestCode == 101)
```

```
        {
            if(grantResults[0]  ==  PackageManager.PERMISSION_GRANTED)
            {
                callPhoneNumber();
            }
        }
    }
    public void callPhoneNumber()
    {
        try
        {
            if(Build.VERSION.SDK_INT > 22)
            {
                if  (ActivityCompat.checkSelfPermission
                        (this, Manifest.permission.CALL_PHONE) !=
PackageManager.PERMISSION_GRANTED) {
                    ActivityCompat.requestPermissions(MainActivity.this, new
String[]{Manifest.permission.CALL_PHONE}, 101);
                    return;
                }

                Intent callIntent = new Intent(Intent.ACTION_CALL);
                callIntent.setData(Uri.parse("tel:" +
                txtPhone.getText().toString()));startActivity(callIntent);

            }
            else {
                Intent callIntent = new Intent(Intent.ACTION_CALL);
                callIntent.setData(Uri.parse("tel:" +
                txtPhone.getText().toString()));startActivity(callIntent);
            }
        }
        catch (Exception ex)

        {

            ex.printStackTrace();

        }

    }

}
```

Activity_main.xml

```
<?xml  version="1.0"  encoding="utf-8"?>

<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"

    android:orientation="vertical
    "
    tools:context=".MainActivity">


    <TextView
        android:id="@+id/fstTxt"

        android:layout_width="wrap_cont
        ent"
        android:layout_height="wrap_cont
        ent"
        android:layout_marginLeft="100d
        p"android:layout_marginTop="150d
        p" android:text="Mobile No"

        />

    <EditText
        android:id="@+id/mblTxt"

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="100dp"
        android:ems="10">

    </EditText>

    <Button
        android:id="@+id/btnCall"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="100dp"
        android:text="Call" />

</LinearLayout>
```

Services:

A service is a component that runs in the background to perform long-running operations without needing to interact with the user and it works even if application is destroyed.

Life Cycle of Android Service

There can be two forms of a service.The lifecycle of service can follow two different paths: startedor bound.

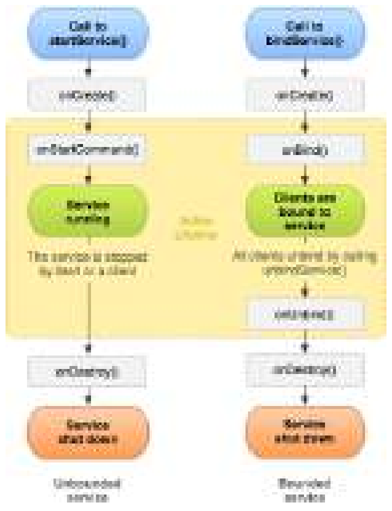
1. Started
2. Bound

1) Started Service

A service is started when component (like activity) calls **startService()** method, now it runsin the background indefinitely. It is stopped by **stopService()** method. The service can stop itself bycalling the **stopSelf()** method.

2) Bound Service

A service is bound when another component (e.g. client) calls **bindService()** method. The client can unbind the service by calling the **unbindService()** method.



Fundamentals of Android Services:

Methods	Description
onStartCommand()	The Android service calls this method when a component(eg: activity) requests to start a service using startService(). Once theservice is started, it can be stopped explicitly using stopService()or stopSelf() methods.
onBind()	This method is mandatory to implement in android service and is invoked whenever an application component calls the bindService() method in order to bind itself with a service. User-interface is also provided to communicate with the service effectively by returning an IBinder object. If the binding of service is not required then the method must return null.
onUnbind()	The Android system invokes this method when all the clients getdisconnected from a particular service interface.
onRebind()	Once all clients are disconnected from the particular interfaceof service and there is a need to connect the service with newclients, the system calls this method.
onCreate()	Whenever a service is created either using onStartCommand() or onBind(), the android system calls this method. This methodis necessary to perform a one-time-set-up.
onDestroy()	When a service is no longer in use, the system invokes thismethod just before the service destroys as a final clean up call.

Methods	Description
	Services must implement this method in order to clean upresources like registered listeners, threads, receivers, etc.

Example of Android Services:

Step 1: Working with the activity_main.xml file

Open the **activity_main.xml** file and add 2 Buttons in it which will start and stop the service. Below is the code for designing a proper activity layout.

```
<?xml  version="1.0"  encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"

   xmlns:app="http://schemas.android.com/apk/res-
   auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   tools:context=".MainActivity">


   <LinearLayout
      android:id="@+id/linearLayout"
      android:layout_width="match_parent
      "
      android:layout_height="wrap_content"
      android:layout_centerVertical="true"
      android:orientation="vertical"

      app:layout_constraintBottom_toBottomOf="parent"
      app:layout_constraintEnd_toEndOf="parent"
      app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintTop_toTopOf="parent"
      app:layout_constraintVertical_bias="1.0"
      tools:ignore="MissingConstraints">


      <TextView
         android:id="@+id/textView1"
         android:layout_width="match_par
         ent"
         android:layout_height="wrap_conte
         nt"
```

```
        android:layout_marginBottom="170dp"
        android:text="@string/heading"
        android:textAlignment="center"

        android:textAppearance="@style/TextAppearance.AppCompat.Lar
        ge" android:textColor="@android:color/holo_green_dark"
        android:textSize="36sp"

        android:textStyle="bold"  />


    <Button

        android:id="@+id/startButton"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginStart="20dp"
        android:layout_marginTop="10dp"
        android:layout_marginEnd="20dp"
        android:layout_marginBottom="20dp"
        android:background="#4CAF50"
        android:text="@string/startButtonText"
        android:textAlignment="center"

        android:textAppearance="@style/TextAppearance.AppCompat.Display1"
        android:textColor="#FFFFFF"

        android:textStyle="bold"  />


    <Button

        android:id="@+id/stopButton"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginStart="20dp"
        android:layout_marginTop="10dp"
        android:layout_marginEnd="20dp"
        android:layout_marginBottom="20dp"
        android:background="#4CAF50"
        android:text="@string/stopButtonText"
        android:textAlignment="center"

        android:textAppearance="@style/TextAppearance.AppCompat.Display1"
        android:textColor="#FFFFFF"

        android:textStyle="bold"  />


</LinearLayout>
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

Step 2: Creating the custom service class

A custom service class will be created in the same directory where the **MainActivity** class resides and this class will extend the **Service class**.

```
public class NewService extends
 Service {private MediaPlayer player;

   @Override

   public int onStartCommand(Intent intent, int flags, int startId) {
       player = MediaPlayer.create( this, R.raw.ss);


       // providing the boolean

       // value as true to play

       // the audio on loop

       player.setLooping( true );


       // starting the
       process
       player.start();


       // returns the status

       // of the program

       return START_STICKY;

   }


   @Override

   public void
       onDestroy() {
       super.onDestroy();
       player.stop();

   }
```

```
        @Null
        able
        @Over
        ride

        public IBinder onBind(Intent
            intent) {return null;

        }

    }
```

Step 5: Working with the MainActivity file

Now, the button objects will be declared and the process to be performed on clicking these buttons will be defined in the MainActivity class.

```
public class MainActivity extends AppCompatActivity implements View.OnClickListener {
    private Button start, stop;

    @Override

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        start = (Button) findViewById( R.id.startButton );


        // assigning ID of stopButton

        // to the object stop

        stop = (Button) findViewById( R.id.stopButton );


        // declaring listeners for the

        // buttons to make them respond

        // correctly according to the
        process
        start.setOnClickListener(this );
        stop.setOnClickListener(this );

    }

    public void onClick(View view) {


        // process to be performed
        // if start button is
        clickedif(view ==
        start){

            // starting the service

            startService(new Intent( this, NewService.class ) );

        }

        // process to be performed

        // if stop button is
        clickedelse if (view
        == stop){

            // stopping the service

            stopService(new Intent( this, NewService.class ) );

        }

    }

}
```

Step 6: Modify the AndroidManifest.xml file

To implement the services successfully on any android device, it is necessary to mention thecreated service in the **AndroidManifest.xml** file.

```
<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.servicedemo">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launch
        er"
        android:label="@string/app_nam
        e"

        android:roundIcon="@mipmap/ic_launcher_round
        " android:supportsRtl="true"
        android:theme="@style/AppTheme">

        <activity android:name=".MainActivity">
```

```
            <intent-filter>
               <action  android:name="android.intent.action.MAIN"  />

               <category  android:name="android.intent.category.LAUNCHER"  />
            </intent-filter>
         </activity>
         <service  android:name=".NewService"/>
      </application>
   </manifest>
```

Notification:

A **notification** is a message you can display to the user outside of your application's normal UI. When you tell the system to issue a notification, it first appears as an icon in the notification area. To see the details of the notification, the user opens the notification drawer. Both the notification area and the notification drawer are system-controlled areas that the user can view at any time.

The NotificationCompat.Builder Class

Method	Description
Notification build()	Combine all of the options that have beenset and return a new Notification object.
NotificationCompat.Builder setAutoCancel (boolean autoCancel)	Setting this flag will make it so thenotification is automatically canceled when the user clicks it in the panel.
NotificationCompat.Builder setContent (RemoteViews views)	Supply a custom RemoteViews to useinstead of the standard one.
NotificationCompat.Builder setContentTitle (CharSequence title)	Set the text (first row) of the notification,in a standard notification.

Create and Send Notifications

Step 1 - Create Notification Builder

As a first step is to create a notification builder using *NotificationCompat.Builder.build()*. You will use Notification Builder to set various Notification properties like its small and large icons,title, priority etc.

```
NotificationCompat.Builder mBuilder = new NotificationCompat.Builder(this)
```

Step 2 - Setting Notification Properties

Once you have **Builder** object, you can set its Notification properties using Builder object asper your requirement. But this is mandatory to set at least following –

- A small icon, set by **setSmallIcon()**
- A title, set by **setContentTitle()**
- Detail text, set by **setContentText()**

Example:

```
mBuilder.setSmallIcon(R.drawable.notification_icon);
mBuilder.setContentTitle("Notification Alert, Click Me!");
mBuilder.setContentText("Hi, This is Android Notification Detail!");
```

Step 3 - Attach Actions

This is an optional part and required if you want to attach an action with the notification. An action allows users to go directly from the notification to an **Activity** in your application, where they can look at one or more events or do further work.

The action is defined by a **PendingIntent** containing an **Intent** that starts an Activity in yourapplication. To associate the PendingIntent with a gesture, call the appropriate method of *NotificationCompat.Builder*. For example, if you want to start Activity when the user clicks the notification text in the notification drawer, you add the PendingIntent by calling **setContentIntent()**.

A PendingIntent object helps you to perform an action on your applications behalf, often at a later time, without caring of whether or not your application is running.

Example:

```
Intent resultIntent = new Intent(this, ResultActivity.class);
TaskStackBuilder stackBuilder = TaskStackBuilder.create(this);
stackBuilder.addParentStack(ResultActivity.class);


// Adds the Intent that starts the Activity to the top of the stack
stackBuilder.addNextIntent(resultIntent);

PendingIntent resultPendingIntent =
stackBuilder.getPendingIntent(0,PendingIntent.FLAG_UPDATE_CURRENT);
mBuilder.setContentIntent(resultPendingIntent);
```

Step 4 - Issue the notification

Finally, you pass the Notification object to the system by calling NotificationManager.notify() to send your notification. Make sure you call **NotificationCompat.Builder.build()** method on builder object before notifying it. This method combines all of the options that have been set and return a new **Notification** object.

Example:

```
NotificationManager mNotificationManager = (NotificationManager)
getSystemService(Context.NOTIFICATION_SERVICE);


// notificationID allows you to update the notification later
on.                    mNotificationManager.notify(notificationID,
mBuilder.build());
```

Accessing Phone services(Call,SMS) :Messaging and E-mail

Sending SMS messages

SMS messages can be send using SmsManager Class or Built-in SMS application.

Using SmsManager class: Using the SmsManager class, you can send SMS messages from withinyour application without the need to involve the built-in Messaging application.

Steps to send SMS

Create a new Android project and name it SMS. Replace the TextView with the following statements in the ***main.xml*** file:

Example:

```
<Button android:id=‖@+id/btnSendSMS‖
        android:layout_width=‖fill_parent‖
        android:layout_height=‖wrap_content‖
        android:text=‖Send SMS‖
        android:onClick=‖onClick‖ />
```

In the AndroidManifest.xml fi le, add the following statements:

```
<uses-permission  android:name=‖android.permission.SEND_SMS/>
```

Add the following statements to the ***SMSActivity.java***
file:import android.telephony.SmsManager;

```
import
android.view.View; public
void onClick(View v) {

        sendSMS(—5556‖, —Hello my friends!‖);

}

//---sends an SMS message to another device--

  private void sendSMS(String phoneNumber, String message)
                                   {

          SmsManager sms = SmsManager.getDefault();
          sms.sendTextMessage(phoneNumber, null, message, null,
          null);
}
```

Methods of SmsManager Class:

Methods	**Description**
ArrayList<String> divideMessage(String text)	This method divides a message textinto several fragments, none bigger than the maximum SMS message size.
static SmsManager getDefault()	This method is used to get the default instance of the SmsManager
void sendDataMessage (String destAddress , String scAddress, shortdestinationPort, byte[]data, PendingIntent sentIntent, PendingIntent deliveryIntent)	This method is used to send a data based SMS to a specific applicationport.
void sendMultipartTextMessage (String destAddress, String scAddress, ArrayList<String> parts, ArrayList<PendingIntent> sentIntents,	Send a multi-part text based SMS.
ArrayList<PendingIntent> deliveryIntents) void sendTextMessage(String destinationAddress, String scAddress, String text, PendingIntent sentIntent, PendingIntent deliveryIntent)	Send a text based SMS.

Sending SMS messages using Intent:

We can also use built-in messaging application to send the message.

Steps to send SMS

Create an Intent Object (Action to send SMS)

ACTION_VIEW action is used to launch an SMS client installed on your Android device.

Intent smsIntent = new Intent(Intent.ACTION_VIEW);

Using Intent Object set Data/Type to send SMS

smsto: is used as a URI to send message using setData()

method. data type is set to vnd.android-dir/mms-sms using

setType() method.**Example:**

smsIntent.setData(Uri.parse("smsto:"));

smsIntent.setType("vnd.android-dir/mms-

sms");

Intent Object used to set a message for one or more phone number with putExtra( ) method and amessage can be send to multiple receivers separated by ‗;‘.

smsIntent.putExtra(―address‖,new String(―0123456789;3398765587‖));

OR

smsInent.putExtra(―address‖,‖5556 ; 5553 ; 5566‖);

Sending E-mail:

We can send Email by using Email/Gmail application available on Android. An Email accountis configured by using POP3 or IMAP.

Steps to send Email

Create Android project and name it as

Emails Add the following statements in

main.xml file:

```
<Button android:id=”@+id/btnSendEmail”

android:layout_width=”fill_parent”

android:layout_height=”wrap_content” android:text=”Send Email”

android:onClick=”onClick” />
```

Add the following statements in ***MainActivity.java*** file:

```
public class MainActivity extends Activity {
     Button btnSendEmail;
     /** Called when the activity is first created.
    */@Override
     public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        btnSendEmail = (Button) findViewById(R.id.btnSendEmail);
        btnSendEmail.setOnClickListener(new  View.OnClickListener() {
            public void onClick(View v) {

                String[] to = {—weimenglee@learn2develop.net‖,
—weimenglee@gmail.co
    m‖};
                String[] cc = {— ‖};

                sendEmail(to, cc, —Hello‖, —Hello my friends!‖);
            }
        });
    }
    //---sends an SMS message to another device---
```

```
            private void sendEmail(String[] emailAddresses, String[] carbonCopies, String subject,
String message)
            {
                    Intent emailIntent = new Intent(Intent.ACTION_SEND);
                    emailIntent.setData(Uri.parse(—mailto:‖));

                    String[] to =
                    emailAddresses;String[]
                    cc = carbonCopies;

                    emailIntent.putExtra(Intent.EXTRA_EMAIL, to);
                    emailIntent.putExtra(Intent.EXTRA_CC, cc);
                    emailIntent.putExtra(Intent.EXTRA_SUBJECT, subject);
                    emailIntent.putExtra(Intent.EXTRA_TEXT, message);
                    emailIntent.setType(—message/rfc822‖);
                    startActivity(Intent.createChooser(emailIntent, —Email‖));


            }


}
```

Lab Assignments:

SET A

1. Create an Android application to service in android that plays an audio in the background. Audio will not be stopped even if you switch to another activity. To stop the audio, you needto stop the service.
2. Create application to send and receive messages using SMSManager.
3. Create application to send email.
4. Create a Notification in Android and display the notification message on second activity.

SET B

1. Create application to design login form, validate it. Write and send email with appropriatemessage.
2. Create an Android application to demonstrate Progress Dialog Box using AsyncTask
3. Create an Android application to demonstrate phone call in android using Implicit Intent.

SET C

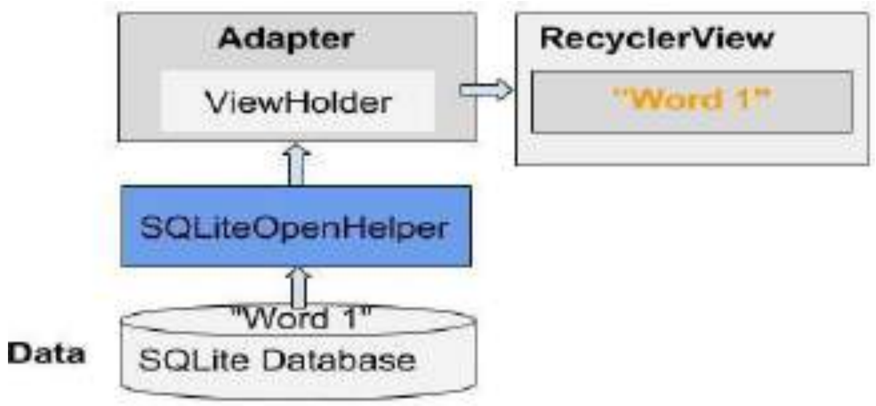
1. Create an Android application to demonstrates how to use a service to download a file from the Internet based on a button click from an activity. Once done, the service notifies the activity via a broadcast receiver that the download is complete.
2. Create application to send email with attachment.

Assignment 7: SQLite Database Connection

Objective:

- How to use database in an application.
- How to create and execute database queries

SQLite is an **open-source relational database** i.e. used to perform database operations on androiddevices such as storing, manipulating or retrieving persistent data from the database. It is embeddedin android bydefault. So, there is no need to perform any database setup or administration task. SQLite is an open source SQL database that stores data to a text file on a device. Android comes inwith built in SQLite database implementation. SQLite supports all the relational database features. In order to access this database, you don't need to establish any kind of connections for it like ODBC.



SQLiteOpenHelper

Android database sqlite.SQLiteOpenHelper manages database creation, up gradation, down gradation, version management and opening it. We need to create sub class of SQLiteOpenHelper and override onCreate and onUpgrade and optionally onOpen. If database is not created, onCreate is called where we write script for database creation. If already created, then onOpen is called which opens database.

There are two constructors of SQLiteOpenHelper class.

Constructors	Description
SQLiteOpenHelper(Context context, String name, SQLiteDatabase.CursorFactory factory, int version)	Create a helper object to create, open, and/or manage a database

SQLiteOpenHelper(Context context, String name, SQLiteDatabase.CursorFactory factory, int version, DatabaseErrorHandler errorHandler)	Create a helper object to create, open, and/or manage a database.

Public Methods	**Description**
void close()	Close any open database object.
String getDatabaseName()	Return the name of the SQLite database being opened, as given to the constructor.
SQLiteDatabase getReadableDatabase()	Create and/or open a database.
SQLiteDatabase getWritableDatabase()	Create and/or open a database that will be used for reading and writing.
void onConfigure(SQLiteDatabase db)	Called when the database connection is being configured, to enable features such as write-ahead logging or foreign key support.
abstract void onCreate(SQLiteDatabase db)	Called when the database is created for the first time.
void onDowngrade(SQLiteDatabase db, int oldVersion, int newVersion)	Called when the database needs to be downgraded.
void onOpen(SQLiteDatabase db)	Called when the database has been opened.

void setIdleConnectionTimeout(long idleConnectionTimeoutMs)	Sets the maximum number of milliseconds that SQLite connection is allowed to be idle before it is closed and removed from the pool.
public synchronized void close ()	Closes the database object.

Example:

```
public  class  DBHelper  extends

        SQLiteOpenHelper{public DBHelper(){

                super(context,DATABASE_NAME,null,1);
        }
        public  void  onCreate(SQLiteDatabase db){

    Creates new
    databaseCreate
    the tables
    execute query with the help of execSQL();
    Add initial data
          }

          public  void   onUpgrade(SQLiteDatabase  db,int  oldVersion,int  newVersion){
          }
}
```

SQLite Database:

The main package is android.database.sqlite that contains the classes to manage your owndatabases:

Database – Creation

In order to create a database you just need to call this method openOrCreateDatabase withyour database name and mode as a parameter. Its syntax is given below:

```
SQLiteDatabase  mydatabase  =  openOrCreateDatabase("Database  name",MODE_PRIVATE,null);
```

Database – Insertion

We can create table or insert data into table using execSQL method defined in SQLiteDatabase class. Its syntax is given below

```
public void insert(String name, String desc) { ContentValues contentValue = new
ContentValues(); contentValue.put(DatabaseHelper.SUBJECT,   name);
contentValue.put(DatabaseHelper.DESC, desc); database.insert(DatabaseHelper.TABLE_NAME,
null, contentValue); }
```

Content Values creates an empty set of values using the given initial size.

Database – Fetching

We can retrieve anything from database using an object of the Cursor class. We will call a method of this class called rawQuery and it will return a resultset with the cursor pointing to the table. We can move the cursor forward and retrieve the data.

Example:

```
Cursor resultSet=mydb.rawQuery(—Select * from
 Login‖,null); resultSet.moveToFirst();

String username=
resultSet.getString(0);String
password=  resultSet.getString(1);
```

Methods	**Description**
int getColumnCount()	Returns the total number of columns of the table.
int getColumnIndex(String columnName)	Returns the index number of a column by specifying the name of the column.

String getColumnName (int columnIndex)	Returns the name of the column by specifying the index of the column.
String[ ] getColumnNames()	Returns the array of all the column names of the table.
int getCount()	Returns the total number of rows in the cursor.
int getPosition()	Returns the current position of the cursor in the table.
boolean isClosed()	Returns true if the cursor is closed.

Lab AssignmentsSET A

Create table Customer (id,

name, address, phno).

Create Android Application

for performing thefollowing

operation on the table.

(using sqlite database)

i) Insert New Customer Details.
ii) Show All the Customer Details

2. Create Table Employee(Eno,Ename,Designation,Salary). Create Android Application for performing the following operation on the table. (Using SQLite Database)

i) Insert New Employee Details.
ii) Display specific employee details.
iii) Display all the Emplyee details.

3. Create simple application shown below. Create table Student(Sid ,Sname ,phno). Useautoincrement for Sid and Perform following Operation.

a. Add Student and display its information.
b. Delete Student.

4. Create sample application with login module (Check username and password). On successfullogin, pass username to next screen And on failing login, alert user using Toast (Hint :Use Login(username, password) Table.

SET B

1. Create Table project (pno, p_name, ptype, duration) and employee (id, e_name, qulification,joindate)

Project – employee have many to many

relationship.Using database perform following

operation.

1) Add new record into table.
2) Display all the project Details.

2. Create Table Employee(Eno,Ename,Designation,Salary). Create Android Application for performing the following operation on the table. (Using SQLite Database)

i) Insert New Employee Details.
ii) Display maximum and minimum salary from employees table
iii) Display average salary and number of employees

3. Create Table Student(Studno,Studname,StudTotalmarks,Studno_of_subjects). Create Android Apllication for performing the following operation on the table. (Using SQLite Database)

i) Insert New student Details.
ii) Display student details.
iii) Display the percent of all students

SET C

1. Create table Game(no,name,type, no_of_players). Create Application to perform the followingoperations.

i) Insert Game Details.
ii) Update no_of_players to four where game is Badminton.
iii) Display all the records.

2. Create Table Employee(empno,emp_name,dept,salary,branch). Create Android Application toperform following operations.

i) Display all Fields of employee table.

ii) Display the name of employee in descending order.

iii) Display the total salary of employee which is greater than > 120000

Location-Based Services and Google Map

Objectives:

- Study the location based services in android
- Use various operations of Google Map

Location based services: —Location-based services‖ is used to create an applicationwhich uses current location, location updates, and location information. The two main LBS elements are:

Location Manager — these services allow applications to obtain periodicupdates of the device's geographical location.

Location Providers — provides periodic reports on the geographical location ofthe device.

Google Map: Google Maps is web based service developed by Google. It provides many facilities such as Satellite view, Street view, real time traffic condition, and navigations for travelling by foot, car, bicycle and public transportation. To use GoogleMaps in your Android applications programmatically.

Google API (Application Programming Interface)

Google APIs is a set of application programming interfaces (APIs) developed by Google which allow communication with Google Services and their integration to other services. Your application needs an API key to access the Google Maps servers.The key is free. You can use it with any of your applications that call the Google MapsAndroid API.

Get Google Maps API Key

Use the link provided in the ***google_maps_api.xml*** file.

Copy the link provided in the ***google_maps_api.xml*** file and pastes it into your browser. The link takes you to the Google API Console and supplies the required information to the Google API Console via URL parameters.

Follow the instructions to create a new project on the Google API Console or selectan existing project.

Create an Android-restricted API key for your project.

Copy the resulting API key, go back to Android Studio, and paste the API key intothe <string> element in the ***google_maps_api.xml*** file.

Displaying the Map

Classes and Interface used for displaying Map are given below:

Interfaces:

Package Name: com.google.android.gms.maps

OnMapReadyCallback – This interface is used to execute method when Map isready.

Public Method	**Description**
abstract void onMapReady (GoogleMap googleMap)	This method execute automatically when map is ready

Package Name: android.app

FragmentManager: FragmentManager is a class used to create transactions for adding, removing or replacing fragments. It used to interact with Fragment inside the activity.

Public Methods	Description
Fragment findFragmentById(int id)	Finds a fragment that was identified by the given id.
Fragment findFragmentByTag(String tag)	Finds a fragment that was identified by the given tag.

GoogleMap: This is the main class of the Google Maps Android API. You cannot instantiate a GoogleMap object directly, rather, you must obtain one from the getMapAsync() method on a MapFragment. This class provides methods to set typeof map, add markers, add polyline or move camera etc.

Constants	Description
MAP_TYPE_HYBRID	Satellite maps with a transparent layer of major streets.
MAP_TYPE_NONE	No base map tiles.
MAP_TYPE_NORMAL	Basic maps.
MAP_TYPE_SATELLITE	Satellite maps with no labels.
MAP_TYPE_TERRAIN	Terrain maps (Topographic data).

SET A

Write a program to perform Zoom In, Zoom Out operation and display Satellite view.

Write a program to perform Zoom In, Zoom Out operation and display Terrain view of currentlocation on Google Map.

Write a program to find the specific location of an Android device and display details of theplace like Address line, city with Geocoding.

Write a program to search a specific location on Google Map.

SET B

Write a program to calculate distance between two locations on Google

Map.Write a program to demonstrate navigation using Google Map.

Write a program to add Marker for indicating specific location using Google Map

SET C

Write a program to track an android device using mobile number and display the location onGoogle Map.

Write a program to modify the above program to draw the path along a route on Google Map.

Section II: Dot Net

ASSIGNMENT -A: DOT NET FRAMEWORK

Basic:

The **.NET Framework** is a software development platform that was introduced by Microsoft in the late 1990 under the NGWS. On 13 February 2002, Microsoft launched the first version of the .NET Framework, referred to as the **.NET Framework 1.0**.

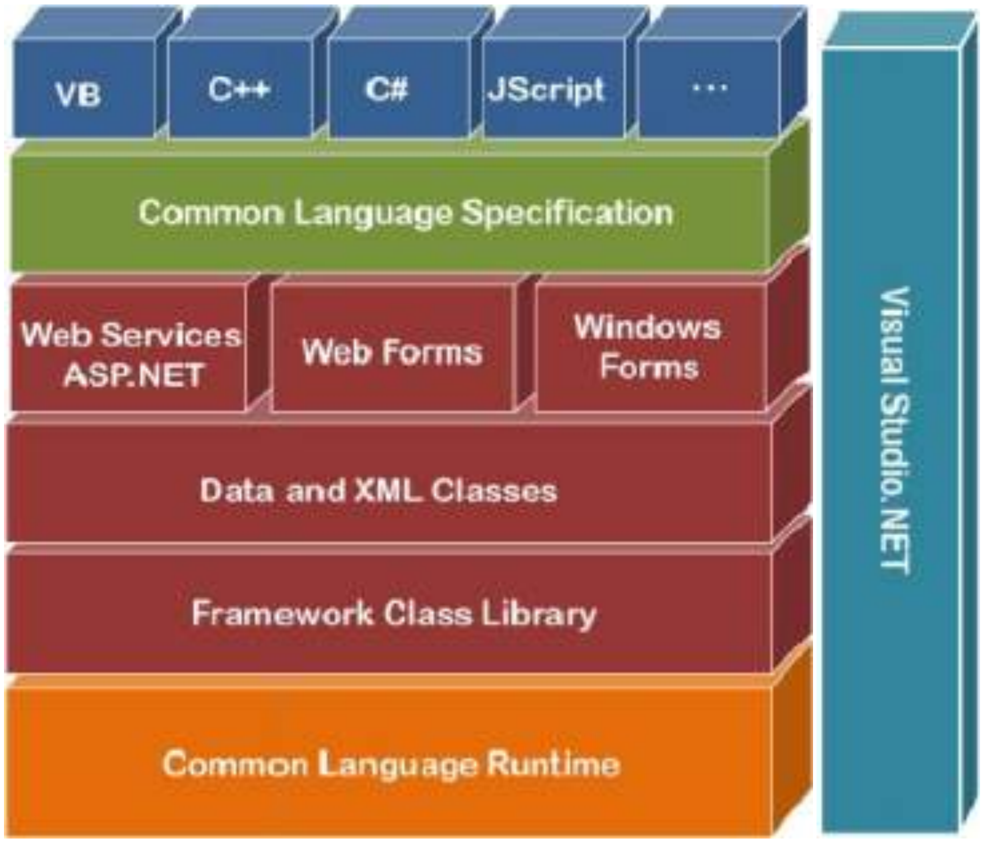
.NET Framework is a virtual machine that provide a common platform to run an application that was built using the different language such as C#, VB.NET, Visual Basic, etc. It is also used to create a form based, console-based and web-based application or services that are available in Microsoft environment. The .NET framework is a pure object oriented, that similar to the Java language . But it is not a platform independent as the Java. So, its application runs only to the windows platform.

The main objective of this framework is to develop an application that can run onthe windows platform.

IDE (Integrated Development Environment)

Visual Studio is the Integrated Development Environment (IDE) provided by Microsoftin which developers can write and execute their program to develop various types of applications such as Windows, Web-based, console-based application easily. It has a rich collection of tools that are used to write and modify the programs and also help to detect and correct the errors in your programs. It is not a language-specific IDE, it means VisualStudio is not only for VB.NET language, but you can use this to write code in C#, visual basic, C++, Python, JavaScript and many more languages supported by Visual Studio. It supports 36 different languages that are used to develop an application. It has a built-in compiler to run the application, and it is also available for Windows and Mac OS.

Architecture of Dot Net Framework



Components of .NET Framework

There are following components of .NET Framework:

1. CLR (Common Language Runtime)
2. CTS (Common Type System)
3. BCL (Base Class Library)
4. CLS (Common Language Specification)
5. FCL (Framework Class Library)
6. .NET Assemblies
7. XML Web Services
8. Window Services

CLR (common language runtime)

It is an important part of a .NET framework that works like a virtual component of the .NET Framework to executes the different languages program like C#, Visual Basic, etc. A CLR also helps to convert a source code into the byte code, and this byte code is known as CIL (Common Intermediate Language) or MSIL (Microsoft Intermediate Language). After converting into a byte code, a CLR uses a JIT compiler at run time that helps to convert a CIL or MSIL code into the machine or native code.

CTS (Common Type System)

It specifies a standard that represent what type of data and value can be defined and managed in computer memory at runtime. A CTS ensures that programming data definedin various languages should beinteract with each other to share information. For example,in C# we define data type as int, while in VB.NET we define integer as a data type.

BCL (Base Class Library)

The base class library has a rich collection of libraries features and functions that help to implement many programming languages in the .NET Framework, such as C #, Visual C++, and more. Furthermore, BCL divides into two parts:

1. *User defined class library*
   - **Assemblies -** It is the collection of small parts of deployment an application's part. It contains either the DLL (Dynamic Link Library) or exe(Executable) file.
2. *Predefined class library*
   - **Namespace -** It is the collection of predefined class and method that present in .Net. In other languages such as, C we used header files, in java we used package similarly we used "using system" in .NET, where using is a keyword and system is a namespace.

CLS (Common language Specification)

It is a subset of common type system (CTS) that defines a set of rules and regulations which should be followed by every language that comes under the .net framework. In other words, a CLS language should be cross-language integration or interoperability. Forexample, in C# and VB.NET language, the C# language terminate each statement with semicolon, whereas in VB.NET it is not end with semicolon, and when these statements execute in .NET Framework, it provides a common platform to interact and share information with each other.

FCL (Framework Class Library)

It provides the various system functionality in the .NET Framework, that includes classes, interfaces and data types, etc. to create multiple functions and different types of application such as desktop, web, mobile application, etc. In other words, it can be defined as, it provides a base on which various applications, controls and components arebuilt in .NET Framework.

ASSIGNMENT -B: VB.Net

Overview of VB.NET

The VB.NET stands for Visual Basic .Network Enabled Technologies. It is a simple, high-level, object-oriented programming language developed by Microsoft in 2002 to replace Visual Basic 6.

VB.NET is an object-oriented programming language. This means that it supports the features of object-oriented programming which include encapsulation, polymorphism, abstraction and inheritance.

Types of Applications that can be developed using VB.NET are :

- Windows Application
- Console Application
- Web Application

Microsoft Visual Studio Interface :

- Menubar
- Toolbar
- Toolbox
- Form Designer Window
- Code Window
- Properties Window
- Server Explorer
- Solution Explorer

A VB.NET define the following structure to create a program:

- Namespace declaration
- Procedure can be multiple
- Define a class or module
- Variables
- The Main procedure
- Statement and Expression
- Comments

Comments

In VB .NET, you write a comment by writing an apostrophe ' or writing REM. Thismeans the rest of the line will not be taken into account by the compiler.

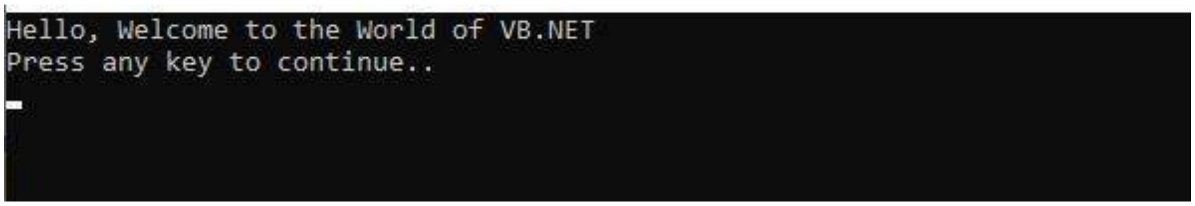
Hello World Program in VB.Net

Create a Hello_Program.vb file in MYConsoleApp project and write the following code:Imports System 'System is a Namespace

```
Module

   Hello_Program

   Sub Main()

      Console.WriteLine("Hello, Welcome to the world of VB.NET")

      Console.WriteLine("Press any key to
      continue...")Console.ReadKey()

     End
 Sub End
 Module
```

Compile and run the above program by pressing the F5 key, we get the follwoing output.

Output:



```
Hello, Welcome to the World of VB.NET
Press any key to continue..
```

Data Types in VB.Net

Data types determine the type of data that any variable can store. Variables belonging to different data types are allocated different amounts of space in the memory.VB.Net provides a wide range of data types. The following table shows all the data types

Data Type	Storage Allocation	Value Range
Boolean	Depends on implementing platform	**True** or **False**
Byte	1 byte	0 through 255 (unsigned)

SByte	1 byte	-128 through 127 (signed)
UInteger	4 bytes	0 through 4,294,967,295 (unsigned)
ULong	8 bytes	0 through 18,446,744,073,709,551,615 (unsigned)
UShort	2 bytes	0 through 65,535 (unsigned)
Short	2 bytes	-32,768 through 32,767 (signed)

Integer	4 bytes	-2,147,483,648 through 2,147,483,647 (signed)
Long	8 bytes	-9,223,372,036,854,775,808 through 9,223,372,036,854,775,807(signed)
Single	4 bytes	-3.4028235E+38 through -1.401298E-45 for negative values; 1.401298E-45 through 3.4028235E+38 for positive values
Double	8 bytes	-1.79769313486231570E+308 through -4.94065645841246544E-324, for negative values 4.94065645841246544E-324 through 1.79769313486231570E+308, for positive values
Decimal	16 bytes	0 through +/- 79,228,162,514,264,337,593,543,950,335 (+/-7.9...E+28) with no decimal point; 0 through +/- 7.9228162514264337593543950335 with 28 places to the right of the decimal
Char	2 bytes	0 through 65535 (unsigned)
String	Depends on implementing platform	0 to approximately 2 billion Unicode characters
Date	8 bytes	0:00:00 (midnight) on January 1, 0001 through 11:59:59 PM on December 31, 9999
Object	4 bytes on 32-bit platform 8 bytes on 64-bit platform	Any type can be stored in a variable of type Object

Summary of Data Types :

Integer Values	**Floating Point Values**
Byte **Sbyte** **Integer** **Uinteger** **Long** **Ulong** **Short** **Ushort**	**Decimal** **Single** **Double**
Boolean Type	**Character Type**

▯ **Boolean**	▯ **Char** ▯ **String**
Date Type ▯ **Date**	**Any type of Value** ▯ **Object**

Example of Datatypes :

```
Module
  DataTypesSub
  Main()

    Dim b As Byte
    Dim n As
    Integer Dim si
    As Single Dim d
    As Double Dim
    c As Char Dim s
    As String Dim
    bl As Boolean


    b = 1

    n = 1234567

    si = 0.12345678901234566

    d = 0.12345678901234566

   c = "U and
    me"s =
    "Me"


      Console.Write(c & " and," & s & vbCrLf)
      Console.WriteLine("We will learn VB.Net
      seriously")Console.WriteLine("The Boolean
      Value", b1) Console.WriteLine("The Integer
      Value", n)

      Console.WriteLine("Lets see what happens to the floating point
      variables:")Console.WriteLine("The Single: {0}, The Double: {1}", si, d)
```

```
            Console.ReadKey()
    End Sub
  End
  Module
```

Type Conversion Functions

There are functions that we can use to convert from one data type to another. They include:

- **CBool** (expression): converts the expression to a Boolean data type.

- **CDate**(expression): converts the expression to a Date data type.
- **CDbl**(expression): converts the expression to a Double data type.
- **CByte** (expression): converts the expression to a byte data type.
- **CChar**(expression): converts the expression to a Char data type.
- **CLng**(expression): converts the expression to a Long data type.
- **CDec**(expression): converts the expression to a Decimal data type.
- **CInt**(expression): converts the expression to an Integer data type.
- **CObj**(expression): converts the expression to an Object data type.
- **CStr**(expression): converts the expression to a String data type.
- **CSByte**(expression): converts the expression to a Byte data type.
- **CShort**(expression): converts the expression to a Short data type.

Variable Declaration

In VB.NET, the declaration of a variable involves giving the variable a name anddefining the data type to which it belongs. We use the following syntax:

Syntax **:** Dim <Variable_Name> As <Data_Type>

Example **:** Dim x As Integer

In the above example, 'x' is the variable name while Integer is the data type to which variable x belongs.

Variable Initialization

Initializing a variable means assigning a value to the variable. The following exampledemonstrates this:

Example-1 :

```
Dim x As
 Integerx = 10
```

Example-2 :

```
Dim name As String
 name  = "Akshit"
```

Example-3 :

```
Dim flag As
 Booleanflag  =
 True
```

Input Output Functions used in Console Application

Output Functions :

1) Write( ) :

Write() method displays the output but do not provide a new line character.

Example :

Console.Write("On
e")
Console.Write("Tw
o") Output :
OneTwo

2) *WriteLine( )*

WriteLine() method displays the output and also provides a new line character itthe end of the string, This would set a new line for the next output.

Example :

Console.WriteLine("On
e")
Console.WriteLine("Two
") **Output :** One

Two

Input Functions : ReadLine( ), Read( ), ReadKey( )Console.ReadLine()

It read all the characters from user input. (and finish when press enter).

Note: It return a STRING so data type should be STRING.

Example 1:

```
Dim name As String
Console.WriteLine("Enter student name
: ")name = Console.ReadLine()

Console.WriteLine( name)                    o/p ------------- Sachin Saxena

            or
```

```
    Console.WriteLine("Student Name is = {0}", name)
                                    o/p ------------- My Name is = Sachin Saxena
```

In this example, if you enter the string "**Sachin Saxena**" it read as it is. It means *itreads all characters until the end of line.*

Example 2:

```
Dim sum As Integer
Console.WriteLine("Enter two numbers :
")

sum = CInt(Console.ReadLine()) +
CInt(Console.ReadLine())Console.WriteLine("Addition =
{0}", sum)
```

Example 3:

```
Dim num1, num2, sum As Integer
Console.WriteLine("Enter two numbers
: ")num1 = Console.ReadLine()

num2 =
Console.ReadLine()sum
= num1 + num2

Console.WriteLine("num1 = {0} num 2 = {1}", num1,
num2)Console.WriteLine("Addition = {0}", sum)
```

Console.Read()

It only accept single character from user input and return its ASCII Code.

Note: Data type should be integer because it returns an integer value as ASCII.

Example :

```
Dim value As
Integer value =
Console.Read( )

Console.WriteLine("Value Read from console is : ",
```

value)As input given is "**a**" and it return its ASCII value is 97

Console.ReadKey()

It reads that which key is pressed by user and returns its name.

Example :

```
Dim key As
ConsoleKeyInfokey =
Console.ReadKey()

Console.WriteLine("You Pressed : {0}", key.Key)
```

Note: It is STRUCT Data type which is ConsoleKeyInfo.

If you press different Buttons from keyboard like Escape, Spacebar, Enter, R, P, Z, Minusand it return name.

Operators

Operator is a symbol that is used to perform various operations on variables. VB.NET hasdifferent types of Operators that help in performing logical and mathematical operations on data values.

Following are the different types of Operators available in VB.NET:

- **Arithmetic Operators**
- *Comparison Operators*
- **Assignment Operators**
- *Logical Operators*

Arithmetic Operators :

You can use arithmetic operators to perform various mathematical operations inVB.NET.

Assume variable **A = 2** and variable **B = 7**, then –

Operator	Description	Example
^	Raises one operand to the power of another	B^A will give 49
+	Adds two operands	A + B will give 9
-	Subtracts second operand from the first	A - B will give -5
*	Multiplies both operands	A * B will give 14
/	Divides one operand by another and returns a **floating point result**	B / A will give 3.5
\	Divides one operand by another and returns an **integer result**	B \ A will give 3
MOD	Modulus Operator and remainder of after an integer division	B MOD A will give 1

Example :

```
Module
   Module1
   Sub Main()

      Dim no1 As Integer =
      11Dim no2As Integer
```

= 5 Dim no3 As
Integer = 2 Dim res1
As Integer Dim res2
As Single

res1 = no1 + res2

```
            Console.WriteLine(" Result of 11 + 5 is {0} ",
                  res1)res1 = no1 – no2

            Console.WriteLine(" Result of 11 - 5 is {0} ",
                  res1)res1 = no1 * no2

            Console.WriteLine(" Result of 11 * 5 is {0} ",
                  res1)res2 = no1 / no2

            Console.WriteLine(" Result of 11 / 5 is {0}",
                  res2)res1 = no1 \ no2

            Console.WriteLine(" Result of 11 \ 5 is {0}",
                  res1)res1 = no1 Mod no2

            Console.WriteLine(" Result of 11 MOD 5 is {0}",
                  res1)res1 = no2^ no2

            Console.WriteLine(" Result of 5 ^ 5 is {0}",
            res1)Console.ReadLine()


          End
        Sub End
        Module
```

Comparison Operators

These operators are used for making comparisons between variables.Assume variable **A** holds 10 and variable **B** holds 20, then –

Operator	Description	Example
=	Checks if the values of two operands are equal or not; if yes, then condition becomes true.	(A = B) is not true.
<>	Checks if the values of two operands are equal or not; if values are not equal, then condition becomes true.	(A <> B) is true.
>	Checks if the value of left operand is greater than the value of right operand; if yes, then condition becomes true.	(A > B) is not true.

<	Checks if the value of left operand is less than the value of right operand; if yes, then condition becomes true.	(A < B) is true.
>=	Checks if the value of left operand is greater than or equal to the value of right operand; if yes, then condition becomes true.	(A >= B) is not true.
<=	Checks if the value of left operand is less	(A <= B) is true.

	than or equal to the value of right operand; if yes, then condition becomes true.	

Assignment Operators :

There are following assignment operators supported by VB.Net –

Operator	Description	Example
=	Simple assignment operator, Assigns values from right side operands to left side operand	C = A + B will assign value of A + B to C
+=	Add and assignment operator, It adds right operand to the left operand and assigns the result to left operand	C += A is equivalent to C = C + A
-=	Subtract and assignment operator, It subtracts right operand from the left operand and assigns the result to left operand	C -= A is equivalent to C = C - A
*=	Multiply and assignment operator, It multiplies right operand with the left operand and assigns the result to left operand	C *= A is equivalent to C = C * A
/=	Divide AND assignment operator, It divides left operand with the right operand and assigns the result to left operand (floating point division)	C /= A is equivalent to C = C / A
\=	Divide and assignment operator, It divides left operand with the right operand and assigns the result to left operand (Integer division)	C \= A is equivalent to C = C \A
^=	Exponentiation and assignment operator. It raises the left operand to the power of the right operand and assigns the result to left operand.	C^=A is equivalent to C = C ^ A
&=	Concatenates a String expression to a String variable or property and assigns the result to the variable or property.	Str1 &= Str2 is same as Str1 = Str1 & Str2

Logical Operators

Following table shows all the logical operators supported by VB.Net.

Assume variable A holds Boolean value True and variable B holds Boolean value False,then −

Operator	Description	Example
And	It is the logical as well as bitwise AND operator. **If both the operands are true,** then condition becomes true. This operator does not perform short-circuiting, i.e., it evaluates both the expressions.	(A And B) is False.
Or	It is the logical as well as bitwise OR operator. If **any of the two operands is true, then condition becomes true**. This operator does not perform short-circuiting, i.e., it evaluates both the expressions.	(A Or B) is True.
Not	It is the logical as well as bitwise NOT operator. Use to **reverses the logical state** of its operand. If a condition is true, then **Logical NOT** operator will make false.	Not(A And B) is True.
Xor	It is the logical as well as bitwise Logical Exclusive OR operator. **It returns True if both expressions are True or both expressions are False**; otherwise it returns False. This operator does not perform short-circuiting, it always evaluates both expressions and there is no short-circuiting counterpart of this operator.	A Xor B is True.
AndAlso	It is the **logical AND** operator. It works only on Boolean data. **It performs short-circuiting**.	(A AndAlso B) is False.
OrElse	It is the **logical OR** operator. It works only on Boolean data. It **performs short-circuiting.**	(A OrElse B) is True.

Example :

X=10, Y = 30

(X < 10) AndAlso (Y < 20)

Control Structure

1. **Decision Making Statements**
2. *Loops Statements*

Decision Making Statements :

1. If statement :
   It executes a set of code when a condition is true.
2. If...Then... Else statement :
   It selects one of two sets of lines to execute.
3. If...Then...ElseIf statement (Nested If statement) :
   It selects one of many sets of lines to execute.
4. Select Case statement :
   It select one of many sets of lines to execute.

1) If Then Statement ***If Then statement*** is a control structure which executes a set of code onlywhen the given condition is True.

Syntax:

```
If [Condition]
   Then
   [Statements]
```

In the above syntax when the **Condition** is true then the **Statements** after **Then** are executed.

Example:

```
Dim x As
Integerx = 9

If (x Mod 2) = 0 Then

    Console.WriteLine("x is an even
number")End If
```

2) If Then Else Statement *If Then Else* statement is a control structure which executes differentset of code statements when the given condition is true or false.

Syntax:

```
If [Condition]
   Then
   [Statements]
Else
   [Statements]
```

In the above syntax when the **Condition** is true, the **Statements** after **Then** are executed. If thecondition is false then the statements after the **Else** part is executed.

Example:

```
Dim x As
Integerx = 9

If (x Mod 2) = 0 THEN
    Console.WriteLine("x is an even
number")Else
```

```
    Console.WriteLine("x is an odd
number")End If
```

3) Nested If Then Else Statement ***Nested If..Then..Else*** statement is used to check multipleconditions using if then else statements nested inside one another.

Syntax :

```
If condition1 Then

        Statement1
        statement2

ElseIf condition2 Then

        Statement3
        Statement4
```

```
 Else
          Statement5
          Statement6
End If
```

Example:

```
Dim x As Integer

x =
Console.ReadLine()
If x > 0 Then

     Console.WriteLine("It is
positive.")Elseif x<0

    Console.WriteLine("It is
negative.")Else

    Console.WriteLine("It is
zeo.")End If
```

Case Statement ***Select case statement*** is used when the expected results for a condition can beknown previously so that different set of operations can be done based on each condition.

Syntax:

```
Select Case
ExpressionCase
Expression1
       Statemen
t1 Case
Expression2
       Statemen
t2 Case
Expressionn
       Statementn
       ...
Case Else
       State
mentEnd
Select
```

In the above syntax, the value of the **Expression** is checked with **Expression1..n** to check if the condition is true. If none of the conditions are matched the statements under the **Case Else** is executed.

Example - 1:

```
Dim grade As
   Chargrade
   = "B" Select
   grade

     Case "A"
       Console.WriteLine("Excellent!")

     Case "B", "C"

       Console.WriteLine("Well
     done")Case "D"

       Console.WriteLine("You
     passed")Case "F"

       Console.WriteLine("Better try
     again")Case Else

       Console.WriteLine("Invalid
   grade")End Select
```

Example - 2:

```
Dim no As Integer

Console.WriteLine("Enter no. between 1-
10")no = Console.ReadLine()

Select Case no
  Case 1, 3, 5,
  7, 9

    Console.WriteLine("Odd number")
  Case 2, 4, 6, 8, 10

    Console.WriteLine("Even number")
  Case Else

    Console.WriteLine("Invalid number-
OutofRange")End Select
```

Looping Statements : *A loop* statement allows us to execute a statement or group of statements multiple times.

1) For Next Loop Statement ***For Next Loop*** Statement executes a set of statements repeatedly in a loop for the given initial, final value range with the specified step by step increment or decrement value.

Syntax:

```
For counter = start To end
    [Step][Statement]
Next [counter]
```

In the above syntax the **Counter** is range of values specified using the **Start**, **End** parameters. The **Step** specifies step increment or decrement value of the counter for which the statements areexecuted.

Example:

```
Private Sub Form1_Load(ByVal sender As
System.Object,ByVal e As System.EventArgs)
Handles MyBase.Load Dim i As Integer
Dim j As
Integerj = 0
For i = 1 To 10
Step 1j = j + 1
MsgBox("Value of j is::"
& j)Next i
End Sub
```

Description:

In the above For Next Loop example the counter value of i is set to be in the range of 1 to 10 andis incremented by 1. The value of j is increased by 1 for 10 times as the loop is repeated.

For Each...Next *For Each...Next* repeats a group of statements for each element in a collection. For-Each, requires a collection. This loop is used for accessing and manipulating all elements in an array or a VB.Net collection.

Syntax :

```
For Each element [As datatype ] In
  groupstatements

 Next [ element ]
```

Example : Print elements of an array

```
Private Sub Form1_Load(ByVal sender As
System.Object,ByVal e As System.EventArgs)
Handles MyBase.Load


    Dim arr() As Integer = {1, 3, 5, 7,
    9}Dim x As Integer


  For Each x In arr             'displaying the values
      MsgBox("Value of x is::" & x)
    N
 ext
 End
 Sub
```

Description:

In the above For Each….Next Loop example the value of array is stored in variable x and displayed in message box. The loop is repeated until all the values are fetched. Total number of iteration of For Each….Next loop is equal to the total number of elements in array. In above example For Each….Next loop executes 5 times as there are 5 values in an array.

3) Do While Loop Statement ***Do While Loop*** Statement is used to execute a set of statements only if the condition is satisfied. But the loop get executed once for a false condition once before exiting the loop. This is also known as **Entry Controlled** loop.

Syntax:

```
Do While
[Condition]
[Statements]
Loop
```

In the above syntax the **Statements** are executed till the **Condition** remains true.

Example:

```
Private Sub Form1_Load(ByVal sender As System.Object,ByVal e As
System.EventArgs) Handles MyBase.Load
Dim a As
Integera = 1
Do While a <
100a = a * 2
MsgBox("Product is::"
& a)Loop
End Sub
```

Description:

In the above Do While Loop example the loop is continued after the value 64 to display 128which is false according to the given condition and then the loop exits.

4) While Wend Statement ***While Wend Statement*** is a looping statement where a condition is checked first, if it is true a set of statements are executed. The condition can also become false atleast once while this statement is used.

Syntax:

```
While
  condition
  [statement
  s]
Wend
```

In the above syntax the **Statements** are executed when the **Condition** is true. The condition mayalso become false while the statements are executed.

Example:

```
Private Sub Form1_Load(ByVal sender As
System.Object,ByVal e As System.EventArgs)
Handles MyBase.Load Dim n As Integer
n = 1
     While n
        <= 1n
        = n +
        1
        MsgBox("First incremented value is:"
     & n)End While
End Sub
```

Description:

In the above example, the value of **n** is 2 in the loop, which is false according to the condition but the loop is continued until the condition is checked. Thus the while wend statement can be used.

5) While Statement *Do Loop While* Statement executes a set of statements and checks thecondition, this is repeated until the condition is true. .It is also known as an **Exit Control** loop **Syntax:**

```
Do
    [Statements]
Loop While
[Condition]
```

In the above syntax the **Statements** are executed first then the **Condition** is checked to find if itis true.

Example:

```
Private Sub Form1_Load(ByVal sender As
System.Object,ByVal e As System.EventArgs)
Handles MyBase.Load Dim cnt As Integer
    Do
        cnt = 10
        MsgBox("Value of cnt is::" &
    cnt)Loop While cnt <= 9
End Sub
```

Description:

In the above Do Loop While example, a messa

6) Do Loop Until Statement *Do Loop Until* Statement executes a set of statements until a condition becomes false, this is an infinite loop might have to be terminated using **Ctrl + Break** .**Syntax:**

```
Do
    [Statements
] Loop Until
[Condition]
```

In the above syntax the **Statements** are executed until the **Condition** becomes false.

Example:

```
Private Sub Form1_Load(ByVal sender As
System.Object,ByVal e As System.EventArgs)
Handles MyBase.Load Dim X As String
    Do
        X = InputBox$("Correct Password
    Please")Loop Until X = "Ranger"
End Sub
```

Description:

In the above Do Until Loop example, a input box is displayed until the correct password is typed.

Windows Applications

Form Control :

Properties – Following properties can be set or read during application execution.

Sr. No	Properties	Description
1	**BackColor**	Sets the form background color.
2	**FormBorderStyle**	The BorderStyle property determines the style of the form's border and the appearance of the form − • **None** − Borderless window that can't be resized. • **Sizable** − This is default value and will be used for resizable window that's used for displaying regular forms. • **Fixed3D** − Window with a visible border, "raised" relative to the main area. In this case, windows can't be resized. • **FixedDialog** − A fixed window, used to create dialog boxes. • **FixedSingle** − A fixed window with a single line border. • **FixedToolWindow** − A fixed window with a Close button only. It looks like the toolbar displayed by the drawing and imaging applications. • **SizableToolWindow** − Same as the FixedToolWindow but resizable. In addition, its caption font is smaller than the usual.
3	**ControlBox**	By default, this property is True and you can set it to False to hide the icon and disable the Control menu.
4	**Height**	This is the height of the Form in pixels.
5	**MinimizeBox**	By default, this property is True and you can set it to False to hide the Minimize button on the title bar.

6	**MaximizeBox**	By default, this property is True and you can set it to False to hide the Maximize button on the title bar.
7	**MinimumSize**	This specifies the minimum height and width of the window you can minimize.
8	**MaximumSize**	This specifies the maximum height and width of the window you maximize.
9	**Name**	This is the actual name of the form.
10	**StartPosition**	This property determines the initial position of the form when it's first displayed. It will have any of the following values − • **CenterParent** − The form is centered in the area of its parent form. • **CenterScreen** − The form is centered on the monitor. • **Manual** − The location and size of the form will determine its starting position. • **WindowsDefaultBounds** − The form is positioned at the default location and size determined by Windows. • **WindowsDefaultLocation** − The form is positioned at the Windows default location and has the dimensions you've set at design time.
11	**Text**	The text, which will appear at the title bar of the form.
12	**TopMost**	This property is a True/False value that lets you specify whether the form will remain on top of all other forms in your application. Its default property is False.
13	**Width**	This is the width of the form in pixel.
14	**AcceptButton**	The button that's automatically activated when you press Enter, no matter which control has the focus at the time. Usually the OK button on a form is set as AcceptButton for a form.
15	**CancelButton**	The button that's automatically activated when you hit the Esc key.

23

		Usually, the Cancel button on a form is set as CancelButton for a form.
16	**AutoScroll**	This Boolean property indicates whether scroll bars will be automatically attached to the form if it is resized to a point that not all its controls are visible.

Form Events:

Following table lists down various important events related to a form −

Sr. No.	Event	Description
1	**Click**	Occurs when the form is clicked.
2	**Closed**	Occurs before the form is closed.
3	**DoubleClick**	Occurs when the form control is double-clicked.
4	**GotFocus**	Occurs when the form control receives focus.
5	**LostFocus**	Occurs when the form loses focus.
6	**KeyDown**	Occurs when a key is pressed while the form has focus.
7	**KeyPress**	Occurs when a key is pressed while the form has focus.
8	**KeyUp**	Occurs when a key is released while the form has focus.
9	**Load**	Occurs before a form is displayed for the first time.
10	**MouseDown**	Occurs when the mouse pointer is over the form and a mouse button is pressed.
11	**MouseEnter**	Occurs when the mouse pointer enters the form.
12	**MouseLeave**	Occurs when the mouse pointer leaves the form.
13	**MouseMove**	Occurs when the mouse pointer is moved over the form.

14	**MouseUp**	Occurs when the mouse pointer is over the form and a mousebutton is released.

Examples on Form Event :

```
Private Sub Form1_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles
Me.Click

            MsgBox("Form Click Event

raised")End Sub

Private Sub Form1_DoubleClick(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.DoubleClick
        MsgBox("Form Double Click Event
raised")End Sub

Private Sub Form1_FormClosed(ByVal sender As Object, ByVal e As
System.Windows.Forms.FormClosedEventArgs) Handles
Me.FormClosed
        MsgBox("Form Closed Event

raised")End Sub
```

Button:

Properties – Following properties are commonly used properties of the Button control –.

Sr. No.	Property & Description
1	**AutoSizeMode** Gets or sets the mode by which the Button automatically resizes itself
2	**BackColor** Gets or sets the background color of the control.
3	**ForeColor** Gets or sets the foreground color of the control.
4	**Image**

	Gets or sets the image that is displayed on a button control.
5	**Location** Gets or sets the coordinates of the upper-left corner of the control relative to the upper-left corner of its container.
6	**TabIndex** Gets or sets the tab order of the control within its container.
7	**Text** Gets or sets the text associated with this control.

TextBox:

The TextBox Control allows you to enter text on your form during runtime.

Properties – Following properties are commonly used properties of the TextBox control –

Sr. No.	Property & Description
1	**Text** Gets or sets the current text in the TextBox.
2	**Font** Gets or sets the font of the text displayed by the control.
3	**ForeColor** Gets or sets the foreground color of the control.
4	**Enabled** For enabling the textbox control
5	**Multiline** Gets or sets a value indicating whether this is a multiline TextBox control.
6	**PasswordChar** Gets or sets the character used to mask characters of a password in a single-line TextBox control.

7	**ReadOnly** Gets or sets a value indicating whether text in the text box is read-only.
8	**ScrollBars** Gets or sets which scroll bars should appear in a multiline TextBox control. This property has values – None, Horizontal, Vertical, Both
9	**TabIndex** Gets or sets the tab order of the control within its container.
10	**TextAlign** Gets or sets how text is aligned in a TextBox control. This property has values – Left, Right, Center
11	**TextLength** Gets the length of text in the control.
12	**WordWrap** Indicates whether a multiline text box control automatically wraps words to the beginning of the next line when necessary.

The Methods of the TextBox Control :

The following are some of the commonly used methods of the TextBox control –

Sr.No.	Method Name & Description
1	**AppendText** Appends text to the current text of a text box.
2	**Clear** Clears all text from the text box control.
3	**Copy** Copies the current selection in the text box to the **Clipboard**.
4	**Cut** Moves the current selection in the text box to the **Clipboard**.
5	**Paste** Replaces the current selection in the text box with the contents of the **Clipboard**.
6	**Undo**

	Undoes the last edit operation in the text box.

- SelectionStart - for setting or getting the starting point for the TextBoxControl.
- SelectionLength - for setting or getting the number of characters that havebeen selected in the TextBox Control.
- SelectedText - returns the TextBox Control that is currently selected.

Events of the TextBox Control :

The following are some of the commonly used events of the Text control –

Sr.No.	Event & Description
1	**Click** Occurs when the control is clicked.
2	**DoubleClick** Occurs when the control is double-clicked.
3	**TextChanged** Occurs when the Text within the textbox changes.

Creating TextBox at Runtime:

```
  Dim txt1, txt2 As New TextBox     //Declare variables of type

  txt1.Name = "txtNo1"        //Set the
propertiestxt1.Left = 100

txt1.Top = 50
txt1.Visible =
True


Me.Controls.Add(txt1)          //Add the control on

formtxt2.Name = "txtNo2"

txt2.Left = 100

txt2.Top = 150
txt2.Visible = True
Me.Controls.Add(txt2)
```

```
Public Class Form1
```

```
    Dim txtNo1 As New
    TextBoxDim btn1 As New
    Button()

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e
AsSystem.EventArgs) Handles Button2.Click

        txtNo1.Left = 100

        txtNo1.Top = 200
        txtNo1.Visible = True
        Me.Controls.Add(txtNo1)

    End Sub

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

        'Dim btn1 As Button = New Button()

        btn1.Parent = Me
        btn1.Name =
        "btn1" btn1.Top =
        10 btn1.Text =
        "Btn1"
        Me.Controls.Add(bt
        n1)

        'adding handler for click event

        AddHandler btn1.Click, AddressOf

    HandleDynamicButtonClickEnd Sub

    Private Sub HandleDynamicButtonClick(ByVal sender As Object, ByVal e As
EventArgs)

        MsgBox(txtNo1.Text)
    End Sub

End Class
```

Label :

The Label Control allows you to enter text on your form during runtime.

Properties – Following properties are commonly used properties of the Label control –

Sr.No.	Property & Description
1	**Autosize**

	Gets or sets a value specifying if the control should be automatically resized to display all its contents.
2	**BorderStyle** Gets or sets the border style for the control.
3	**FlatStyle** Gets or sets the flat style appearance of the Label control
4	**Font** Gets or sets the font of the text displayed by the control.
5	**FontHeight** Gets or sets the height of the font of the control.
6	**ForeColor** Gets or sets the foreground color of the control.
7	**PreferredHeight** Gets the preferred height of the control.
8	**PreferredWidth** Gets the preferred width of the control.
10	**Text** Gets or sets the text associated with this control.
11	**TextAlign** Gets or sets the alignment of text in the label.

Anchoring and Docking Controls to a Form :

Anchoring Control:

- The Anchor property lets you attach one or more edges of the control to corresponding edges of the form.
- The anchored edges of the control maintain the same distance from the corresponding edges of the form.
- Place a TextBox control on a new form, set its MultiLine property to True, and then open the control’s Anchor property in the Properties window.

Docking Control:

- Dock property, which determines how a control will dock on the form. The defaultvalue of this property is None.

CheckBox :

The CheckBox control allows the user to set true/false or yes/no type options. The user can select or deselect it. When a check box is selected it has the value True, and when itis cleared, it holds the value False.

Properties of the CheckBox Control :

The following are some of the commonly used properties of the CheckBox control −

Sr. No.	Property & Description
1	**Appearance** Gets or sets a value determining the appearance of the check box.
2	**CheckAlign** Gets or sets the horizontal and vertical alignment of the check mark on the check box.
3	**Checked** Gets or sets a value indicating whether the check box is selected.
4	**CheckState** Gets or sets the state of a check box.
5	**Text** Gets or sets the caption of a check box.
6	**ThreeState** Gets or sets a value indicating whether or not a check box should allow three check states rather than two.

Events of the CheckBox Control :

The following are some of the commonly used events of the CheckBox control −

Sr. No.	Event & Description
1	**AppearanceChanged** Occurs when the value of the Appearance property of the check box is changed.
2	**CheckedChanged** Occurs when the value of the Checked property of the CheckBox

	control is changed.
3	**CheckStateChanged** Occurs when the value of the CheckState property of the CheckBox control is changed.

Following Example shows use of CheckState Property and CheckStateChangedEvent

```
Private Sub CheckBox1_CheckStateChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles CheckBox1.CheckStateChanged

     If CheckBox1.CheckState = CheckState.Checked
       ThenMsgBox("Yes")

    End If

    If CheckBox1.CheckState = CheckState.Indeterminate
       ThenMsgBox("May Be")

    End If

    If CheckBox1.CheckState = CheckState.Unchecked
       ThenMsgBox("No")

    En

d IfEnd

Sub
```

Example 2:

Add four check boxes on form. The check boxes will allow the users to choose the programming languages. When the user clicks the Submit button, he/she gets an appropriate message.

Solution:

```
If CheckBox1.Checked = True
      Thenstr &=
      CheckBox1.Text
```

```
                str &=
            " "End If
```

RadioButton :

The RadioButton control is used to provide a set of mutually exclusive options.The user can select one radio button in a group.

If you need to place more than one group of radio buttons in the same form, you shouldplace them in different container controls like a GroupBox control.

Properties :

The following are some of the commonly used properties of the RadioButton control –

Sr. No.	Property & Description
1	**Appearance** Gets or sets a value determining the appearance of the radio button.
2	**CheckAlign** Gets or sets the location of the check box portion of the radio button.
3	**Checked** Gets or sets a value indicating whether the control is checked.
4	**Text** Gets or sets the caption for a radio button.

ListBox:

The ListBox represents a Windows control to display a list of items to a user. A user can select an item from the list. It allows the programmer to add items at design time by usingthe properties window or at the runtime.

Properties:

The following are some of the commonly used properties of the ListBox control –

Sr. No.	Property & Description
1	**AllowSelection :** Gets a value indicating whether the ListBox currently enables selection of list items.
2	**HorizontalScrollBar :**Gets or sets the value indicating whether a horizontal scrollbar is displayed in the list box.

3	**Items :** Gets the items of the list box.
4	**MultiColumn :** Gets or sets a value indicating whether the list box supports multiple columns.
5	**ScrollAlwaysVisible :**Gets or sets a value indicating whether the vertical scroll bar is shown at all times.
6	**SelectedIndex :**Gets or sets the zero-based index of the currently selected item in a list box.
7	**SelectedIndices :** Gets a collection that contains the zero-based indexes of all currently selected items in the list box.
8	**SelectedItem:** Gets or sets the currently selected item in the list box.
9	**SelectedItems :**Gets a collection containing the currently selected items in the list box.
10	**SelectionMode** Gets or sets the method in which items are selected in the list box. This property has values – • None – No item can be selected • One – Only one item can be selected • MultiSimple – Multiple items can be selected • MultiExtended – Multiple items can be selected with the help of Shift Ctrl and arrow key
11	**Sorted** Gets or sets a value indicating whether the items in the list box are sorted alphabetically.
12	**Text** Gets or searches for the text of the currently selected item in the list box.
13	**TopIndex** Gets or sets the index of the first visible item of a list box.

Adding Items to ListBox at Runtime :

Add( ) method : It is used to add items in ListBox at Runtime

Syntax :

ListBoxName.Items.Add(“DataItme”)

Example :

lstCity.Items.Add(“Amravati”)

Insert( ) method : It is used to insert items at particular position in ListBox at Runtime

Syntax :

ListBoxName.Items.Insert(Index, “DataItme”)

Example :

lstCity.Items.Insert(2,“Amravati”)

Count( ) method :

It returns the total number of items in the ListBox

Syntax : Example :

Variable name =

ListBoxName.Items.Countn =

lstCity.Items.Count

Clear( ) method : It is used to clear the ListBox

Syntax :

ListBoxName.Items.Clear( )

Example :

lstCity.Items.Clear( )

Removing Items from ListBox at Runtime :

Remove( ) and RemoveAt( ) methods :

These two methods are used to remove items from ListBox at Runtime

i) Remove( ) : Removes item by passing corresponding item.

Syntax :

ListBoxName.Items.Remove(ItemName)

Example :

i) lstCity.Items.Remove("Pune") : It removes city Pune from ListBox lstCity
ii) lstCity.Items.Remove(lstCity.SelectedItem) : It removes city Pune from ListBoxlstCity

ii) RemoveAt( ) : Removes item by passing corresponding index.

Syntax :

ListBoxName.Items.RemoveAt(Index)

Example :

i) lstCity.Items.RemoveAt(2) : It removes city placed at index 2 from ListBox lstCity
ii) lstCity.Items.RemoveAt(lstCity.SeletedIndex) : It removes currently selected Cityfrom ListBox lstCity

Events of the ListBox Control :

The following are some of the commonly used events of the ListBox control –

Sr.No.	Event & Description
1	**Click** Occurs when a list box is selected.
2	**SelectedIndexChanged** Occurs when the SelectedIndex property of a list box is changed.

ComboBox Control :

- The ComboBox control is used to display a drop-down list of various items.
- It is a **combination of a text box** in which the user enters an item and a drop-down list from which the user selects an item.

Properties of the ComboBox Control :

Sr.No.	Property & Description
1	**DropDownStyle :** It specifies the appearance and behavior of the ComboBox Values : 1) Simple 2) DropDown 3) DropDownList
2	**ItemHeight** **Gets** or sets the height of an item in the combo box.
3	**Items** Gets an object representing the collection of the items contained in this ComboBox.
4	**MaxDropDownItems** Gets or sets the maximum number of items to be displayed in the drop-down part of the combo box. To apply this first set the property IntegralHeight = False
5	**MaxLength** Gets or sets the maximum number of characters a user can enter in the editable area of the combo box.
6	**SelectedIndex** Gets or sets the index specifying the currently selected item.
7	**SelectedItem** Gets or sets currently selected item in the ComboBox.

8	**SelectedText** Gets or sets the text that is selected in the editable portion of a ComboBox.
9	**Sorted** Gets or sets a value indicating whether the items in the combo box are sorted.
10	**Text** Gets or sets the text associated with this control.

RichTextBox :

The RichTextBox control allows the user to display, enter, and edit text while alsoproviding more advanced formatting features than the conventional TextBox control.

Methods :

1] SaveFile : The SaveFile method saves the contents of the control to a disk file.

Syntax : RichTextBox1.SaveFile(path, filetype)

Example :

2] LoadFile : The LoadFile method loads a text or RTF file to the control.

Syntax : RichTextBox1.LoadFile(path, filetype)

Example :Properties :

Following properties are used to read or change the alignment of one or more paragraphs:

- SelectionIndent : Sets / Gets the left indentation of paragraph.
- SelectionRightIndent : Sets / Gets the right indentation of paragraph.
- SelectionHangingIndent : Indents all other lines of paragraph with respect to SelectionIndent.

Example:

```
RichTextBox1.SelectionIndent          = 20

RichTexBox1.SelectionHangingIndent = -25

RichTextBox1.RightIndent              = 10
```

Formatting Text of RichTextBox

```
:Dim fd As New FontDialog
fd.ShowDialog()
TextBox1.Font = fd.Font
```

MsgBox ( ) Function

The objective of MsgBox is to produce a pop-up message box and prompt the user to click on a command button before he /she can continues. This format is as follows:

myMsg=MsgBox(Prompt, Style Value, Title)

The first argument, Prompt, will display the message in the message box. The Style Value will determine what type of command buttons appear on the message box, please refer to the below Table for types of command button displayed. The Title argument will display the title of the message board.

Style Value	Named Constant	Buttons Displayed
0	vbOkOnly	Ok button
1	vbOkCancel	Ok and Cancel buttons
2	vbAbortRetryIgnore	Abort, Retry and Ignore buttons.
3	vbYesNoCancel	Yes, No and Cancel buttons
4	vbYesNo	Yes and No buttons
5	vbRetryCancel	Retry and Cancel buttons

We can use named constants in place of integers for the second argument to make the programs more readable.

For example:

myMsg=MsgBox("Click OK to Proceed", 1, "Startup Menu")and
myMsg=Msg("Click OK to Proceed". vbOkCancel,"Startup Menu")are the same.

myMsg is a variable that holds values that are returned by the MsgBox ( ) function. The values are determined by the type of buttons being clicked by the users. It has to be declared as Integer data type in the procedure or in the general declaration section. Below table shows the values, thecorresponding named constant and buttons.

Value	Named Constant	Button Clicked
1	vbOk	Ok button
2	vbCancel	Cancel button
3	vbAbort	Abort button
4	vbRetry	Retry button
5	vbIgnore	Ignore button

6	vbYes	Yes button
7	vbNo	No button

Example:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
        Dim testmsg As Integer
        testmsg = MsgBox("Click to test", 1, "Test
        message")If testmsg = 1 Then
                MessageBox.Show("You have clicked the OK button")
        Else
                MessageBox.Show("You have clicked the Cancel button")
        End If
    End
    Sub
```

You can add an icon besides the message. There are four types of icons available as shown inTable

Value	Named Constant	Icon
16	vbCritical	⊗
3	vbQuestion	❓
48	vbExclamation	⚠
64	vbInformation	ℹ

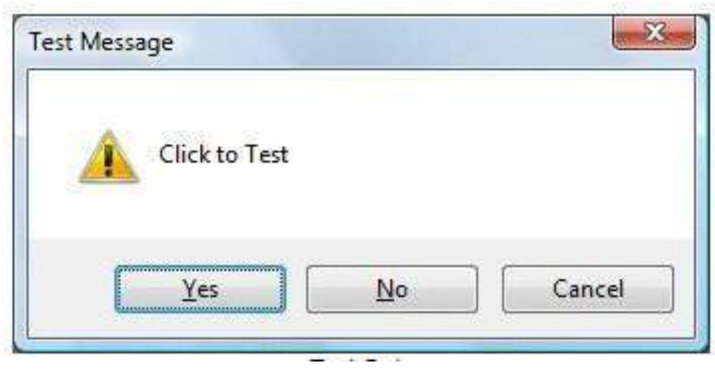
Example:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
        Dim testMsg As Integer
        testMsg = MsgBox("Click to Test", vbYesNoCancel + vbExclamation,
"TestMessage")
        If testMsg = 6 Then
                MessageBox.Show("You have clicked the yes
        button")ElseIf testMsg = 7 Then
                MessageBox.Show("You have clicked the NO button")
```

```
Else
        MessageBox.Show("You have clicked the Cancel button")
End

If
```

End Sub

The first argument, Prompt, will display the message



The InputBox( ) Function

An InputBox( ) function will display a message box where the user can enter a value or amessage in the form of text.
Format of InputBox Function:
myMessage=InputBox(Prompt, Title, default_text, x-position, y-position)

myMessage is a variant data type but typically it is declared as string, which accept the messageinput by the users.

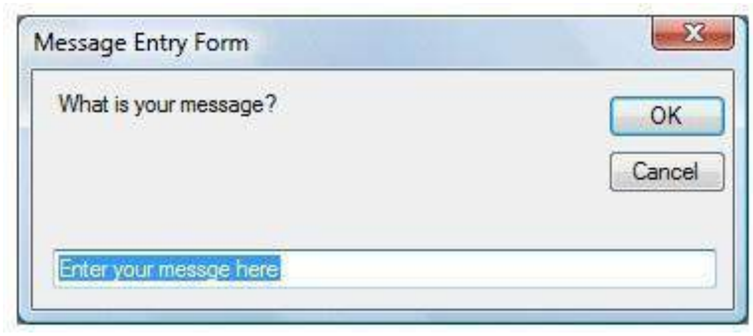
The arguments are explained as follows:

Prompt - the message displayed normally as a question asked.
Title - The title of the Input Box.
default-text - The default text that appears in the input field where users can use it as hisintended input or he may change to the message he wish to enter.
x-position and y-position - the position or tthe coordinates of the input box.
InputBox(Prompt, Title, default_text, x-position, y-position)
The parameters remain the same.

Example:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
          Dim userMsg As String
          userMsg = Microsoft.VisualBasic.InputBox("What is your message?", "Message
EntryForm", "Enter your messge here", 500, 700)
          If userMsg <> "" Then
                    MessageBox.Show(userMsg)
          Else
                    MessageBox.Show("No Message")
          End If
End
Sub
```

The InputBox will appear as shown in the figure below when you press the command button.



Predefined Dialog Box in VB.Net:

- There are many built-in dialog boxes to be used in Windows forms for various tasks like opening and saving files, printing a page, providing choices for colors, fonts,page setup, etc., to the user of an application.
- The ShowDialog method is used to display all the dialog box controls at run-time.
- The ShowDialog method returns a value of the type of DialogResult enumeration.The values of DialogResult enumeration are −
  - Abort − returns DialogResult. Abort value, when user clicks an Abort button.
  - Cancel − returns DialogResult. Cancel, when user clicks a Cancel button.
  - Ignore − returns DialogResult. Ignore, when user clicks an Ignore button.
  - No − returns DialogResult. No, when user clicks a No button.
  - None − returns nothing and the dialog box continues running.
  - OK − returns DialogResult. OK, when user clicks an OK button
  - Retry − returns DialogResult. Retry , when user clicks an Retry button
  - Yes − returns DialogResult. Yes, when user clicks an Yes button

1] **ColorDialog :** It represents a common dialog box that displays available colors alongwith controls that enable the user to define custom colors.

Example :

```
Dim ColorDialog1 As New ColorDialog

Private Sub Button1_Click(sender As Object, e As EventArgs)
          HandlesButton1.Click

  If ColorDialog1.ShowDialog < > Windows.Forms.DialogResult.Cancel
    ThenLabel1.ForeColor = ColorDialog1.Color

  En
d If
End
Sub
```

2] **FontDialog :** It prompts the user to choose a font from among those installed on thelocal computer and lets the user select the font, font size, and color.

Example :

```
Dim FontDialog1 As New FontDialog

If FontDialog1.ShowDialog <> Windows.Forms.DialogResult.Cancel
    ThenRichTextBox1.ForeColor = FontDialog1.Color

    RichTextBox1.Font =
  FontDialog1.FontEnd If
```

3] OpenFileDialog : It prompts the user to open a file and allows the user to select a fileto open.

Example:

```
Dim OpenFileDialog1 As New OpenFileDialog

 If OpenFileDialog1.ShowDialog <> Windows.Forms.DialogResult.Cancel
     ThenPictureBox1.Image = Image.FromFile(OpenFileDialog1.FileName)

   End If
```

4] SaveFileDialog : It prompts the user to select a location for saving a file and allowsthe user to specify the name of the file to save data.

Example :

```
SaveFileDialog1.Filter = "TXT Files (*.txt*)|*.txt"

    If SaveFileDialog1.ShowDialog = Windows.Forms.DialogResult.OK _
          Then

      My.Computer.FileSystem.WriteAllText _
      (SaveFileDialog1.FileName, RichTextBox1.Text,
      True)

    End If
```

5] PrintDialog : It lets the user to print documents by selecting a printer and choosingwhich sections of the document to print from a Windows Forms application.

Example :

```
PrintDialog1.Document = PrintDocument1
  PrintDialog1.PrinterSettings =
  PrintDocument1.PrinterSettings
  PrintDialog1.AllowSomePages = True


  If PrintDialog1.ShowDialog = DialogResult.OK Then
    PrintDocument1.PrinterSettings =
    PrintDialog1.PrinterSettingsPrintDocument1.Print()

  End If
```

Assignment 1:

Set A

1. Write a Vb.net program to accept string and count the number of words and display thecount.
2. Write VB.Net program to covert decimal number into binary, hexadecimal, and octalnumber.
3. Write a Vb.net program to print second highest value in an array.

Set B

1. Write a Vb.net program to find Prime Numbers between 1 to 500.
2. Write a Vb.net program to find Spy Numbers between 100 to 500.
3. Write a Vb.net program to check whether entered string is palindrome or not.

Set C

1. Write a Vb.net program to check whether entered number is Spy Number or not UsingConsole Application.
   [Note : A positive integer is called a spy number if the sum and product of its digits are equal.]

Assignment Evaluation

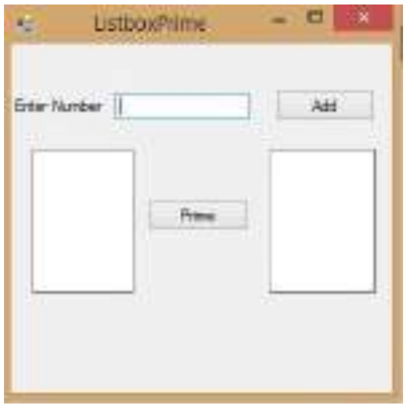
0: Not Done [ ]	**1: Incomplete [ ]**	**2: Late Complete**
[ ]3: Need Improvement [ ]	**4: Complete [ ]**	**5: Well Done [ ]**

Signature of Instructor

Assignment 2

Set A

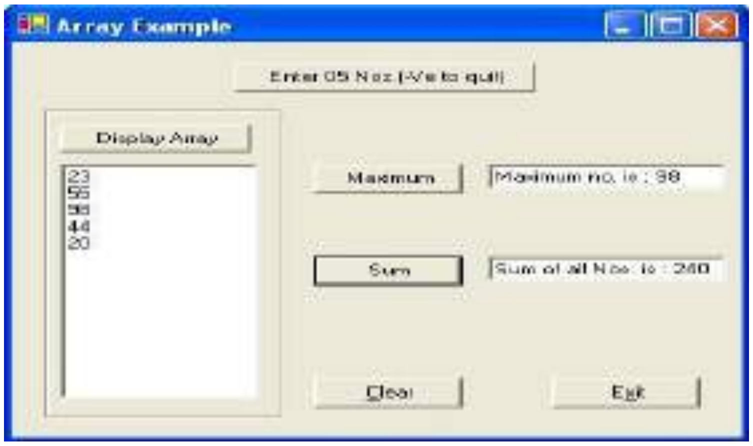
1. Write a Vb.net program to design the following form, accept the numbers through textbox and add them into the ListBoxe1 by clicking on Add button. When user click on Prime button then all the prime numbers from ListBox1 should get added into ListBox2.



2. Write a Vb.net program for blinking an image.
3. Design a Vb.net form to pick a date from DateTimePicker Control and display day, month and year in separate text boxes.

Set B

1. Write VB.Net program to add three text boxes at runtime. Allow user to enter values and then display Maximum and Minimum value using message box after click on command button.
2. Write VB .Net program to take three text boxes and two buttons on the form. Enter different strings in first and second textbox. On clicking the First command button, concatenation of two strings should be displayed in third text box and on clicking second command button, reverse of string should display in third text box.
3. Write VB.Net Program to Design following form to store numbers in single dimensional array and to find Maximum no & sum of all numbers in array.



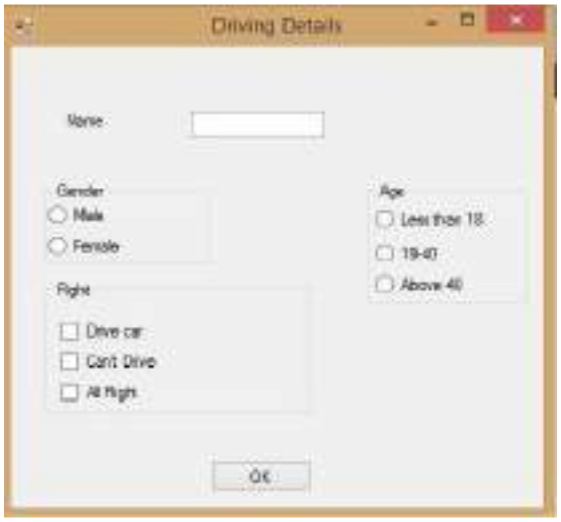
23
55
98
44
20

Maximum no. is : 98

Sum of all Nos is : 240

Set C

1. Write a Vb.net program to design the following form, allow the user to select radio buttons from Gender and Age Panel. After Selection appropriate CheckBox from Right Panel should be selected automatically. Display appropriate message into the MessageBox by clicking on Ok button.



2. Write VB.net program to design a calculator with proper validation.

Assignment Evaluation

0: Not Done [ ]	**1: Incomplete [ ]**	**2: Late Complete [ ]**
3: Need Improvement [ ]	**4: Complete [ ]**	**5: Well Done [ ]**

Signature of Instructor

Assignment 3

1. Write a Vb.net program to check whether entered number is Armstrong or.
2. Write a Vb.net program to accept a character from keyboard and check whether it isvowel or not. Also display the case of that character.
3. Write a Vb.net program to accept number from user into the TextBox. Calculate the square root of that number also convert the entered number into binary number and display result into the Message Box
4. Write a Vb.net program to accept a number from an user through InputBox and displayits multiplication table into the ListBox.
5. Write a Vb.Net program to move the Text "Arts Commerce Science College" continuously from Left to Right.
6. Write a VB.NET program to do the following operations on RichTextBox values
   i) Font Style
   ii) Font Color
   iii) Save
   iv) Open

7. Write a Vb.net program to design screen to accept the details from the user. Clicking on Submit button Net Salary should be calculated and displayed into the TextBox. Display the MessageBox informing the Name and Net Salary of employee.

Assignment Evaluation

0: Not Done [ ] **1: Incomplete [ ]** **2: Late Complete [ ]**

3: Need Improvement [ ] **4: Complete [ ]** **5: Well Done [ ]**

Signature of Instructor

ASSIGNMENT - C: C#.NET

C# is a general-purpose, object-oriented, modern programming language, which is developed byMicrosoft and is a part of ".Net framework".

C# programming supports multi-paradigm, strong typing, lexically scope, imperative, declarative, functional, etc.

Most of the syntaxes of the C# programming language are similar to the C and C++programming languages.

we created a C# file called Program1.cs, and we used the following code to print "Hello World"to the screen:

```
using System;

namespace HelloWorld
{
  class Program1
  {
    static void Main(string[] args)
    {
      Console.WriteLine("Hello World!");
    }
  }
}
```

```
Hello World!
```

Example explained

Line 1: using System means that we can use classes from the **System** namespace.

Line 2: namespace is used to organize your code, and it is a container for classes and other namespaces.

Line 3: The curly braces **{}** marks the beginning and the end of a block of code.

Line 4: class is a container for data and methods, which brings functionality to your program. Every line of code that runs in C# must be inside a class. In our example, we named the class Program1.

Line 6: The **Main** method. Any code inside its curly brackets {} will be executed. It is an entrypoint method, Program execution is stared from main method.

Line 7: Console is a class of the **System** namespace, which has a **WriteLine()** method that is used to output/print text. In our example it will output "Hello World!".

Note:

1. If you omit the **using System** line, you would have to write **System.Console.WriteLine()** toprint/output text.

2. Every C# statement ends with a semicolon (;)

3. C# is case-sensitive: "MyClass" and "myclass" has different

meaning.**C# Comments:** Comments are the statements ignored by

the compiler. **Types of C# compiler:**

1. Single-line Comments

Single-line comments start with two forward slashes (//).
Any text between // and the end of the line is ignored by C# compiler (will not be executed).

Example

```
// This is a comment
Console.WriteLine("Hello
World!");
```

2. C# Multi-line Comments

Multi-line comments start with /* and ends with */.
Any text between /* and */ will be ignored by C# compiler.

Example

```
/* This is the multi line comment and
the output of this example is Hello
Worldon the screen. */
Console.WriteLine("Hello World!");
```

C# Variables

Variables are the identifier, used for the storing values.

Syntax:

```
datatype variableName = value;
```

Example:

```
string name = "Vikas";
Console.WriteLine(name);
```

Constants

Constant is a variable that can't be change its value during the program once initialized.
Examlple : const doublePI = 3.142;

C# Data Types

Data Type tells us that which kind of value store into a variable and how many bytes allocatedfor that variable.

Data Type	Size	Description
int	4 bytes	Stores whole numbers from -2,147,483,648 to 2,147,483,647
long	8 bytes	Stores whole numbers from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
float	4 bytes	Stores fractional numbers. Sufficient for storing 6 to 7 decimal digits
Double	8 bytes	Stores fractional numbers. Sufficient for storing 15 decimal digits
Bool	1 bit	Stores true or false values
Char	2 bytes	Stores a single character/letter, surrounded by single quotes
String	2 bytes per character	Stores a sequence

Note: A floating point number can also be a scientific number with an "e" to indicate the powerof 10:

```
float f1 = 35e3F;
double d1 = 12E4D;
Console.WriteLine(f1);
Console.WriteLine(d1);
```

```
Outpu
t
35000
```

C# Type Casting

Type Casting is the converting the value of variable into one data type to another data

type.In C#, there are two types of casting:

- **Implicit Casting** (automatically) - converting a smaller type to a larger type size
  char -> int -> long -> float -> double
  Example:

```
int intNum = 10;
double doubleNum = intNum;       // Automatic casting: int to double
Console.WriteLine(intNum);      // Outputs 10
Console.WriteLine(doubleNum);  // Outputs 10
```

- **Explicit Casting** (manually) - converting a larger type to a smaller size type
  double -> float -> long -> int -> char

  Example:

```
double doubleNum= 13.22;
int intNum= (int) doubleNum;    // Manual casting: double to int
Console.WriteLine(doubleNum); // Outputs 13.22
Console.WriteLine(intNum);      // Outputs 13
```

Type Conversion Methods

There are some built-in methods which converts data types explicitly, such as Convert.ToBoolean, Convert.ToDouble, Convert.ToString, Convert.ToInt32 (int)and Convert.ToInt64 (long):

```
int intNum = 10;
double doubleNum= 13.22;
bool boolNum = true;
Console.WriteLine(Convert.ToString(intNum));    // convert int to string
Console.WriteLine(Convert.ToDouble(intNum));    // convert int to double
Console.WriteLine(Convert.ToInt32(doubleNum));  // convert double to int
Console.WriteLine(Convert.ToString(boolNum));  // convert bool to string
```

Get User Input
Console.ReadLine() is the method that read the line of text from keyboard.Example:

```
using System;

namespace MyApplication
{
  class Program2
  {
    static void Main(string[] args)
    {
Console.WriteLine("Enter your full name:");
string fullName = Console.ReadLine();
Console.WriteLine("Enter your age:");
int age = Console.ReadLine();
Console.WriteLine("Your Full Name is: " + fullName+" and Age is:"+age);

}
  }
}
```

C# Operators

Operators are used to perform operations on variables and values.

Operator	Name	Example
Arithmetic Operators		
+	Addition	x + y
-	Subtraction	x - y
*	Multiplication	x * y
/	Division	x / y
%	Modulus	x % y
++	Increment	x++
--	Decrement	x--
Comparison Operators		
==	Equal to	x == y
!=	Not equal	x != y
>	Greater than	x > y
<	Less than	x < y
>=	Greater than or equal to	x >= y
<=	Less than or equal to	x <= y
Logical Operators		
&&	Logical and	x < 5 && x < 10
||	Logical or	x < 5 || x < 4
!	Logical not	!(x < 5 && x < 10)
Assignment Operator		
=	Assigning the RHS value to LHS	x = 5
Short hand assignment operator		
+=	x += 3	x = x + 3
-=	x -= 3	x = x - 3
*=	x *= 3	x = x * 3
/=	x /= 3	x = x / 3
%=	x %= 3	x = x % 3
&=	x &= 3	x = x & 3
|=	x |= 3	x = x | 3
^=	x ^= 3	x = x ^ 3

>>=	x >>= 3	x = x >> 3
<<=	x <<= 3	x = x << 3

Control Structure:

There are three types of control structures in c# these are:

1. Decision Making Control Statements:

i. **if Statement:**When condition is true then statements within if block are executed.Syntax:

```
if (condition)
{
  // block of code to be executed if the condition is True
}
```

```
using System;

namespace MyApplication {
  class Program3
  {
    static void Main(string[] args)
    {
      if (20>10)
      {
        Console.WriteLine("20 is greater than 10");
      }
    }
  }
}
```

ii. **if-else statement:** if condition is true then statements within if block are executedotherwise statements within else block are executed.

Syntax:

```
if (condition)
{
  // block of code to be executed if the condition is True
}
else
{
  // block of code to be executed if the condition is False
}
```

```
using System;

namespace MyApplication {
  class Program4
```

```
 {
   static void Main(string[] args)
   {
           Console.Write("Enter number:");
           int num = Convert.ToInt32(Console.ReadLine());
          if (num>=0)
 {
Console.WriteLine(num+" is +ve");
}
       else
          {
Console.WriteLine(num+" is -ve");
}
   }
  }
}
```

iii. **Nested if statement:**if with ladder or if within if are called nested if statement.Syntax:

```
if (condition1)
{
  // block of code to be executed if condition1 is True
}
else if (condition2)
{
  // block of code to be executed if the condition1 is false and condition2 is True
}
else
{
  // block of code to be executed if the condition1 is false and condition2 is False
}
```

```
using System;

namespace MyApplication
{
  class Program5
  {
    static void Main(string[] args)
    {
          double per;
Console.Write("Enter the Percentage : ");
                         per = Convert.ToDouble(Console.ReadLine());
      if (per >=75 10)
      {
        Console.WriteLine("First class with Distinction.");
      }
      else if (per >=60 && per<75)
      {
        Console.WriteLine("First Class.");
      }
     else if (per >=50 && per<60)
      {
        Console.WriteLine("Second Class.");
      }
     else if (per >=40 && per<50)
      {
        Console.WriteLine("Pass Class.");
      }
else {
        Console.WriteLine("Fail.");
      }
    }
  }
}
```

iv. **Switch-case statement:**select one of many code blocks to be executed.

Syntax:

```
switch(expression)
{
  case literal1:     // code block
                     break;
  case literal2:
               // code block
                       break;
  default:
               // code block
}
```

```
using System;

namespace MyApplication
{
class Program6
{

static void Main(string[] args)
{
int monthNumber;
Console.Write("Enter Month Number (1 - 12): ");
monthNumber = Convert.ToInt32(Console.ReadLine());
```

```
switch (monthNumber)
{
case 1:
Console.WriteLine("January");
break;
case 2:
Console.WriteLine("February");
break;
case 3:
Console.WriteLine("March");
break;
case 4:
Console.WriteLine("April");
break;
case 5:
Console.WriteLine("May");
break;
case 6:
Console.WriteLine("June");
break;
case 7:
Console.WriteLine("July");
break;
case 8:
Console.WriteLine("August");
break;
case 9:
Console.WriteLine("September");
break;
case 10:
Console.WriteLine("October");
break;
case 11:
Console.WriteLine("November");
break;
case 12:
Console.WriteLine("December");
break;
default:
Console.WriteLine("you did not enter correct value for month name");
break;
}

}
}
```

57

```
}
```

2. Loop Control Structures :

Loops are the statements executed frequently until a specified condition becomes false.

i. While loop: it is an entry point loop it executed as long as a specified condition is true.

Syntax:

```
while (condition) {
  // code block to be executed
}
```

Example:Find the Sum of First N Numbers in C#

```
using System;

namespace MyApplication
{
class Program7
   {
      static void Main(string[] args)
      {
          int num, sum=0;

          Console.Write("Enter a Number : ");
          num = Convert.ToInt32(Console.ReadLine());

          if(num <0)
          {
             Console.Write("Please Enter Positive Number");
          }
          else
          {
             while(num >0)
             {
                sum =sum+num;
                num = num - 1;
             }
          }
          Console.WriteLine("The sum is "+sum);

          Console.ReadKey();
      }
   }
}
```

ii. **do-while loop:** it is an exit control loop it executed at least once even thoughcondition is false.

Syn
tax:

```
do
{
  // code block to be executed
} while (condition);
```

Example: Find the Sum of Digits of a given number in C#

```
using System;

namespace MyApplication
{
class Program8
   {
       static void Main(string[] args)
       {
           int number, sum=0,lastDigit;

           Console.Write("Enter a Number : ");
           number = Convert.ToInt32(Console.ReadLine());

               do
               {
                  lastDigit=number%10;
                   sum = sum+ lastDigit;
                   number =number/10;
               }while(number>0);

           Console.WriteLine("The sum is "+sum);

           Console.ReadKey();
       }
   }
}
```

iii. **for loop:** When we know the number of iterations in advance then for loop isused instead of while loop.

Syntax:

```
for (Initialization ; Condition ; Updation)
{
  // code block to be executed
}
```

Example: C# Program to Find the Factorial of a Number

```
using System;
```

```
namespace MyApplication
{
class Program9
   {
static void Main(string[] args)
       {
           int i, number, fact;
           Console.WriteLine("Enter the Number");
           number = int.Parse(Console.ReadLine());
           fact = number;
           for (i = 1; i <= number; i++)
           {
               fact = fact * i;
           }
           Console.WriteLine("\nFactorial of Given Number is: "+fact);
           Console.ReadLine();

      }
   }
}
```

3. **Jump Control Structures:** Transfer the control of flow unorderly.

i. **break statement:** it is use for abnormal termination of loop; it means break transferthe control out of loop.

```
using System;


namespace MyApplication

{

  class Program10

  {

    static void Main(string[] args)

    {

      for (int i = 0; i < 10; i++)

      {

        if (i == 4)
```

Output

```
0
1
2
3
```

ii. **continue statement:** it is use for breaks one iteration in the loop, if a specified condition occurs, and continues with the next iteration in the loop.

```
using System;


namespace MyApplication

{

  class Program11

  {

    static void Main(string[] args)

    {

      for (int i = 0; i < 10; i++)

      {

        if (i == 4)
```

```
Outpu
t0

1

2

3

5
```

Array

Array is use to storing the heterogeneous elements into single variable.
Array element is accessed using an index, it starts from 0 to size-1.

Syntax:
Datatype[] array_name=new Datatype[size];

Example:program to find min and max in an

```
using System;
```

array.

```
namespace MyApplication
{
 class Program12
 {
      static void Main(string[] args)
      {
         int[] numbers = new
         int[10]; Random rnd =
         new Random();int min,
         max;

         for (int i = 0; i < numbers.Length; i++)
         {
            numbers[i] = rnd.Next(1,
            100);
            Console.WriteLine(numbers[i
            ]);
         }
         min =
         numbers[0];
         max =
         numbers[0];
         for (int i = 1; i < numbers.Length; i++)
         {
            if (min >
               numbers[i])
               min =
               numbers[i];
            if (max <
               numbers[i])
               max =
               numbers[i];
         }
         Console.WriteLine("====================================");
         Console.WriteLine("The highest number in the array: " +
         max);Console.WriteLine("The lowest number in the array:"
         + min); Console.ReadKey();
      }
   }
}
```

Ou
tpu
t

```
56
95
23
75
63
45
```

```
19
7
32
83
The highest number in the
array:95The lowest number in
the array:7
```

Method: It is a block of statements used to perform a specific task and it may return value to thecalling program.
Syntax to define method:

```
static return_type method_name([parameter_list])
{
        //method statements
}
```

Note: [] indicates it is optional.

Example: Check whether given number is even or odd using method

```
using System;

namespace MyApplication
{
class Program13
   {
      static bool IsEvenNumber(int num)
      {
         if (num % 2 == 0)
         {
            return true;
         }
         else
         {
            return false;
         }
      }
      static void Main(string[] args)
      {
         int n;
         Console.Write("Enter an integer : ");
         n = Int32.Parse(Console.ReadLine());

         if (IsEvenNumber(n))
         {
            Console.WriteLine("{0} is even", n);
         }
         else
         {
            Console.WriteLine("{0} is odd", n);
         }

         Console.ReadKey();
      }
   }
}
```

Output

```
Enter an integer:
```

String class:

In C#, string is keyword which is an alias for System.String class. That is why string and Stringare equivalent. We are free to use any naming convention.

string s1 = "india";//creating string using string

keywordString s2 = "Bharat";//creating string using

String class **String methods**

Method Name	**Description**
Clone()	It is used to return a reference to this instance of String.
Compare(String, String)	It is used to compares two specified String objects. It returns an integer that indicates their relative position in the sort order.
CompareOrdinal(String, String)	It is used to compare two specified String objects by evaluating the numeric values of the corresponding Char objects in each string..
CompareTo(String)	It is used to compare this instance with a specified String object. It indicates whether this instance precedes, follows, or appears in the same position in the sort order as the specified string.
Concat(String, String)	It is used to concatenate two specified instances of String.
Contains(String)	It is used to return a value indicating whether a specified substring occurs within this string.
Copy(String)	It is used to create a new instance of String with the same value as a specified String.
CopyTo(Int32, Char[], Int32, Int32)	It is used to copy a specified number of characters from a specified position in this instance to a specified position in an array of Unicode characters.
EndsWith(String)	It is used to check that the end of this string instance matches the specified string.
Equals(String, String)	It is used to determine that two specified String objects have the same value.
Format(String, Object)	It is used to replace one or more format items in a specified string with the string representation of a specified object.
GetEnumerator()	It is used to retrieve an object that can iterate through the individual characters in this string.
GetHashCode()	It returns the hash code for this string.
GetType()	It is used to get the Type of the current instance.
GetTypeCode()	It is used to return the TypeCode for class String.

IndexOf(String)	It is used to report the zero-based index of the first occurrence of the specified string in this instance.
Insert(Int32, String)	It is used to return a new string in which a specified string is inserted at a specified index position.
Intern(String)	It is used to retrieve the system's reference to the specified String.
IsInterned(String)	It is used to retrieve a reference to a specified String.
IsNormalized()	It is used to indicate that this string is in Unicode normalization form C.
IsNullOrEmpty(String)	It is used to indicate that the specified string is null or an Empty string.
IsNullOrWhiteSpace(String)	It is used to indicate whether a specified string is null, empty, or consists only of white-space characters.
Join(String, String[])	It is used to concatenate all the elements of a string array, using the specified separator between each element.
LastIndexOf(Char)	It is used to report the zero-based index position of the last occurrence of a specified character within String.
LastIndexOfAny(Char[])	It is used to report the zero-based index position of the last occurrence in this instance of one or more characters specified in a Unicode array.
Normalize()	It is used to return a new string whose textual value is the same as this string, but whose binary representation is in Unicode normalization form C.
PadLeft(Int32)	It is used to return a new string that right-aligns the characters in this instance by padding them with spaces on the left.
PadRight(Int32)	It is used to return a new string that left-aligns the characters in this string by padding them with spaces on the right.
Remove(Int32)	It is used to return a new string in which all the characters in the current instance, beginning at a specified position and continuing through the last position, have been deleted.
Replace(String, String)	It is used to return a new string in which all occurrences of a specified string in the current instance are replaced with another specified string.
Split(Char[])	It is used to split a string into substrings that are based on the characters in an array.
StartsWith(String)	It is used to check whether the beginning of this string instance matches the specified string.
Substring(Int32)	It is used to retrieve a substring from this instance. The substring starts at a specified character position and continues to the end of the string.
ToCharArray()	It is used to copy the characters in this instance to a Unicode character array.
ToLower()	It is used to convert String into lowercase.
ToLowerInvariant()	It is used to return convert String into lowercase using the casing rules of the invariant culture.

ToString()	It is used to return instance of String.
ToUpper()	It is used to convert String into uppercase.
Trim()	It is used to remove all leading and trailing white-space characters from the current String object.
TrimEnd(Char[])	It Is used to remove all trailing occurrences of a set of characters specified in an array from the current String object.
TrimStart(Char[])	It is used to remove all leading occurrences of a set of characters specified in an array from the current String object.

Example:

```
using System.Text;
namespace MyApplication
{
    class Program13
    {
        static void Main(string[] args)
        {
            string firstname;
            string lastname;

            firstname = "Steven Clark";
            lastname = "Clark";


  Console.WriteLine(firstname.Clone());
// Make String Clone
            Console.WriteLine(firstname.CompareTo(lastname));
//Compare two string value and returns 0 for true and1 for false

 Console.WriteLine(firstname.Contains("ven")); //Check whether specified value exists or not
in string

  Console.WriteLine(firstname.EndsWith("n")); //Check whether specified value is the last
character of string
            Console.WriteLine(firstname.Equals(lastname));
//Compare two string and returns true and false


  Console.WriteLine(firstname.GetHashCode());
//Returns HashCode of String

  Console.WriteLine(firstname.GetType());
//Returns type of string
```

```
  Console.WriteLine(firstname.GetTypeCode());
//Returns type of string

  Console.WriteLine(firstname.IndexOf("e")); //Returns the first index position of specified
value
the first index position of specified value

  Console.WriteLine(firstname.ToLower());
//Covert string into lower case

  Console.WriteLine(firstname.ToUpper());
//Convert string into Upper case

  Console.WriteLine(firstname.Insert(0, "Hello")); //Insert substring into string

  Console.WriteLine(firstname.IsNormalized());
//Check Whether string is in Unicode
normalizationfrom C


   Console.WriteLine(firstname.LastIndexOf("e")); //Returns the last index position
ofspecified value

 Console.WriteLine(firstname.Length);
//Returns the Length of String

 Console.WriteLine(firstname.Remove(5));
//Deletes all the characters from begining to specified index.

 Console.WriteLine(firstname.Replace('e','i')); // Replace the character

  string[] split = firstname.Split(new char[] { 'e' }); //Split the string based on specified value


            Console.WriteLine(split[
            0]);
            Console.WriteLine(split[
            1]);
            Console.WriteLine(split[
            2]);

  Console.WriteLine(firstname.StartsWith("S"));
//Check wheter first character of string is same as specified value

  Console.WriteLine(firstname.Substring(2,5));
//Returns substring

  Console.WriteLine(firstname.ToCharArray());
//Converts an string into char array.
```

```
  Console.WriteLine(firstname.Trim());

//It removes starting and ending white spaces fromstring.

        }
```

```
Output
Steven Clark
1

True
False
False

1470518261

System.String
String

2

steven clark
STEVEN CLARK

HelloSteven Clark
True

4

12

Steve Stivin
```

Object Oriented Concepts

Object-oriented programming has several advantages over procedural programming:

- OOP is faster and easier to execute
- OOP provides a clear structure for the programs
- OOP helps to keep the C# code DRY "Don't Repeat Yourself", and makes the code easierto maintain, modify and debug
- OOP makes it possible to create full reusable applications with less code and shorterdevelopment time.

Object and Classes:

In C#, Object is a real world entity, for example, chair, car, pen, mobile, laptop etc.

```
Student s1 = new Student();//creating an object of Student
```

It is a template from which objects are created. It can have fields, methods, constructorsetc.

Example:

```
using System;
  public class Student
   {
       public int id;
       public String name;
       public void insert(int i, String n)
       {
           id = i;
           name = n;
       }
       public void display()
       {
           Console.WriteLine(id + " " + name);
       }
  }
  class TestStudent{
      public static void Main(string[] args)
```

Output

```
1001 Abhijeet
```

Access Modifiers / Specifiers

Access modifiers or specifiers are the keywords that are used to specify accessibility or scope ofvariables and functions in the C# application.
C# provides five types of access specifiers.

Access Specifier	**Description**
Public	It specifies that access is not restricted.
Protected	It specifies that access is limited to the containing class or in derived

	class.
Internal	It specifies that access is limited to the current assembly.
protected internal	It specifies that access is limited to the current assembly or types derived from the containing class.
Private	It specifies that access is limited to the containing type.

Constructors

In C#, constructor is a special method its name should be same as its class name.which is invoked automatically at the time of object creation.
It is used to initialize the data members of new object generally.It does not have any return type even though void.

There can be two types of constructors in C#.

1. **Default constructor:** constructor does not have any parameters or arguments.
2. **Parameterized constructor:** constructor have at least one parameter.

```
    using System;
      public class Employee
       {
          public int id;
          public String name;
          public float salary;
public Employee()
     {
    id = 101;
             name = “Rahul Bhosale”;
             salary = 750000f;


        Console.WriteLine("Default Constructor Invoked");
     }

          public Employee(int i, String n,float s)
          {
             id = i;
             name = n;
             salary = s;
    Console.WriteLine("Parameterised Constructor Invoked");
          }
          public void display()
          {
             Console.WriteLine(id + " " + name+" "+salary);
          }
    }

    class TestEmployee{

       public static void Main(string[] args)

        {

           Employee e1 = new Employee();

           Employee e2 = new Employee(102, "Mahesh Raut", 460000f);
           e1.display();

           e2.display();
```

```
Output

101Rahul Bhosale 750000
```

Inheritance

In C#, inheritance is a process in which object of one class acquires the properties of object of another class.
Inheritance provides reusability of code and extends or modifies the attributes, behaviors whichis defined in other class.
The class whose members are inherited is called base class or parent class or super class and theclass which inherits the members of another class is called derived class or child class or sub class.

Types of inheritance:

1. Single Inheritance
2. Multilevel Inheritance
3. Hierarchical Inheritance

Example of Multilevel Inheritance:

```
using System;
  public class Animal
   {
     public void eat() { Console.WriteLine("Eating..."); }
  }
  public class Dog: Animal
  {
     public void bark() { Console.WriteLine("Barking..."); }
  }
  public class BabyDog : Dog
  {
     public void weep() { Console.WriteLine("Weeping..."); }
  }
  class TestInheritance{
     public static void Main(string[] args)
      {
            BabyDog d1 = new BabyDog();
            d1.eat();

            d1.bark();
        }
    }
```

```
Output
Eating...
Barking...
Weeping
```

Abstract Classes and Methods

The abstract keyword is used for classes and methods:

Abstract class: It can have abstract and non-abstract methods. It cannot be instantiated.Its implementation must be provided by derived classes. Here, derived class is forced to providethe implementation of all the abstract methods.

Abstract method: A method which is declared abstract and has no body is called abstract method. It can be declared inside the abstract class only. Its implementation must beprovided by derived classes.

Example:

```
using System;
public abstract class Shape
{
   public abstract void draw();
}
public class Rectangle : Shape
{
   public override void draw()
   {
      Console.WriteLine("drawing rectangle...");
   }
}
public class Circle : Shape
{
   public override void draw()
   {
      Console.WriteLine("drawing circle...");
   }
}
public class TestAbstract
{
  public static void Main()
  {
      Shape s;
      s = new Rectangle();
      s.draw();
}
```

```
Output
drawing
```

Interface

It is used to achieve multiple inheritance which can't be achieved by class

An interface is a completely "abstract class", which can only contain abstract methods (with empty bodies)

Example:

```
using System;
public interface Drawable
{
    void draw();
}
public class Rectangle : Drawable
{
    public void draw()
    {
        Console.WriteLine("drawing rectangle...");
    }
}
public class Circle : Drawable
{
    public void draw()
    {
        Console.WriteLine("drawing circle...");
    }
}
public class TestInterface
{
    public static void Main()
    {
        Drawable d;
        d = new Rectangle();
        d.draw();

        d = new Circle();
    }
}
```

```
Output

drawing
```

Method Overloading

Method having same name but different prototyping or signature called as method overloading.

The advantage of method overloading is that it increases the readability of the program becauseyou don't need to use different names for same action.

Example

```
using System;

public class Calculate{

    public static int sum(int a,int b){
        return a + b;

    }

    public static int sum(int a, int b, int c)

    {

        return a + b + c;

    }

}

public class TestMemberOverloading
```

```
Outpu
t30
```

Method Overriding

If derived class defines same method as defined in its base class, it is known as method overriding.
It is used to achieve runtime polymorphism.

To perform method overriding, you need to use virtual keyword with base class methodand override keyword with derived class method.
Example:

```
using System;

public class Animal{ public
   virtual void eat(){

      Console.WriteLine("Eating...");

   }

}

public class Dog: Animal

{

   public override void eat()

   {

      Console.WriteLine("Eating bread...");

   }

}
```

```
Output

Eating bread
```

Delegates

In C#, Delegate is a reference to the method. It works like function pointer in C and C++.
But itis objected-oriented, secured and type-safe than function pointer.
For static method, delegate encapsulates method only. But for instance method, it encapsulatesmethod and instance both.
The best use of delegate is to use as event.
Internally a delegate declaration defines a class which is the derived class of System.Delegate.

Example:

```
using System;
delegate int Calculator(int n);//declaring delegate
public class DelegateExample
{
   static int num = 100;
   public static int addition(int n)
   {
      num = num + n;
```

```
        return num ;
    }
    public static int multiplication(int n)
    {
        num = num * n;
        return num;
    }
    public static int getNumber()
    {
        return num;
    }
    public static void Main(string[] args)
    {
        Calculator c1 = new Calculator(addition);//instantiating delegate
        Calculator c2 = new Calculator(multiplication);
        c1(20);//calling method using delegate
        Console.WriteLine("After c1 delegate, Number is: " + getNumber());
        c2(3);
        Console.WriteLine("After c2 delegate, Number is: " + getNumber());

    }
}
```

Output

```
After c1 delegate, Number is: 120
After c2 delegate, Number is: 360
```

Assignment 4

Set A:

1. Program to display the addition, subtraction, multiplication and division of two number.
2. Program to display the first 10 natural numbers and their sum.
3. Accept three and find their maximum and minimum.
4. C# Program to Calculate the Power of a Number Without Using Math.Pow.
5. Define a Student class (roll number, name, and percentage). Define a default andparameterized constructor. Override the toString method. Keep a count objects created. Create objects using parameterized constructor and display the object count aftereach object is created. (Use static member and method).
6. Define a class Employee having members – id, name, department, salary. Define default and parameterized constructors. Create a subclass called "Manager" with member bonus. Define methods accept and display in both the classes. Create no. objects of the Manager class and display the details of the manager having the maximum total salary (salary + bonus)
7. Define an interface "IntOperations" with methods to check whether an integer ispositive,negative, even, odd, prime and operations like factorial and sum of digits. Define a class MyNumber having one private int data member. Write a default constructor toinitialize itto 0 and another constructor to initialize it to a value (Use this). Implement theabove interface. Create an object in main.

Set B:

1. Define an interface "StackOperations" which declares methods for a static stack. Define a class "MyStack" which contains an array and top as data members and implements the above interface. Initialize the stack using a constructor. Write a menu driven program to perform operations on a stack object.
2. Define a class CricketPlayer (name, no_of_innings, no_times_notout, total_runs,bat_avg). Create an array of n player objects. Calculate the batting average for each Player using a static method avg(). Define a static method "sortPlayer"which sorts the array on the basis of average. Display the player details in sorted order.

3. Create a class date with day, month and year as members. Write appropriate member functions. Create another class student, which has id, name, date of birth and marks of 3 subjects as members. Write appropriate constructor for the student which assigns values to the members. Accept the details from keyboard and create a student object using the arguments. Display the student details in a proper format.

Set C:

1. Define a class “Employee” which has members id, name, date of birth. Define another class “Manager” which has members department name and joining dateand extends Employee. Create n objects of the manager class.

2. Define an abstract class “Staff” with members name and address. Define twosubclasses of this class – “FullTimeStaff” (department, salary) and “PartTimeStaff” (numberof_hours, rate_per_hour). Define appropriate constructors. Create n objects which could be of either FullTimeStaff or PartTimeStaff class by asking the user’s choice. Display details of all “FullTimeStaff” objects and all “PartTimeStaff” objects.

3. Create an interface “CreditCardInterface” with methods to viewCreditAmount,viewPin, changePin, useCard and payBalance. Create a class Customer (name, cardnumber, pin, creditAmount – initialized to 0). Implement methods viewCreditAmount,viewPin, changePin and payBalance of the interface. From Customer,create classesRegularCardHolder (maxCreditLimit) andGoldCardHolder (String specialPrivileges)and define the remaining methods of the interface. Create n objects of the RegularCardHolder and GoldCardHolder classes and writea menu driven program to perform the following actions
    1. Use Card
    2. Pay Balance
    3. Change Pin

ASSIGNMENT -D: ASP.NET

ASP.NET is a web development platform, which provides a programming model, a comprehensive software infrastructure and various services required to build up robust web applications for PC as well as mobile devices. ASP.NET works on top of the HTTP protocol, anduses the HTTP commands and policies to set a browser-to-server bilateral communication and cooperation. ASP.NET is a part of Microsoft .Net platform. ASP.NET applications are compiled codes, written using the extensible and reusable components or objects present in .Net framework. These codes can use the entire hierarchy of classes in .Net framework.

ASP.NET application codes can be written in any of the following languages:

- C#
- Visual Basic.Net
- Jscript
- J#

ASP.NET is used to produce interactive, data-driven web applications over the internet. It consists of a large number of controls such as text boxes, buttons, and labels for assembling, configuring, and manipulating code to create HTML pages.

The architecture of the.Net framework is based on the following key components

1. **Language** – A variety of languages exists for .net framework. They are VB.net and C#. These can be used to develop web applications.
2. **Library** – The .NET Framework includes a set of standard class libraries. The most common library used for web applications in .net is the Web library. The web library has all the necessary components used to develop.Net web-based applications.
3. **Common Language Runtime** – The Common Language Infrastructure or CLI is a platform. .Net programs are executed on this platform. The CLR is used for performing key activities. Activities include Exception handling and Garbage collection.

ASP.Net Page Lifecycle

When an ASP.Net page is called, it goes through a particular lifecycle. This is done beforethe response is sent to the user. There are series of steps which are followed for the processing of an ASP.Net page.

The various stages of the lifecycle of an ASP.Net web page are as:

1. **Page Request**– This is when the page is first requested from the server. When the page isrequested, the server checks if it is requested for the first time. If so, then it needs to compile the page, parse the response and send it across to the user. If it is not the first time the page is requested, the cache is checked to see if the page output exists. If so, that response is sent to the user.

2. **Page Start** – During this time, 2 objects, known as the Request and Response object are created. The Request object is used to hold all the information which was sent when the page was requested. The Response object is used to hold the information which is sent back to the user.
3. **Page Initialization** – During this time, all the controls on a web page is initialized. So if you have any label, textbox or any other controls on the web form, they are all initialized.
4. **Page Load** – This is when the page is actually loaded with all the default values. So if a textbox is supposed to have a default value, that value is loaded during the page loadtime.
5. **Validation** – Sometimes there can be some validation set on the form. For example, therecan be a validation which says that a list box should have a certain set of values. If the condition is false, then there should be an error in loading the page.
6. **Postback event handling** – This event is triggered if the same page is being loaded again. This happens in response to an earlier event. Sometimes there can be a situation that a user clicks on a submit button on the page. In this case, the same page is displayed again. In such a case, the Postback event handler is called.
7. **Page Rendering** – This happens just before all the response information is sent to the user. All the information on the form is saved, and the result is sent to the user as a complete web page.
8. **Unload** – Once the page output is sent to the user, there is no need to keep the ASP.net web form objects in memory. So the unloading process involves removing all unwanted objects from memory.

Working with Views and Windows

You can work with windows in the following ways:

- To change the Web Forms Designer from one view to another, click on the Design orsource button.
- To close a window, click on the close button on the upper right corner and to redisplay,select it from the View menu.
- To hide a window, click on its Auto Hide button. The window then changes into a tab.
- To display again, click the Auto Hide button again.
- To change the size of a window, just drag it.

Adding Folders and Files to your Website

When a new web form is created, Visual Studio automatically generates the starting HTML for the form and displays it in Source view of the web forms designer. The Solution Explorer is usedto add any other files, folders or any existing item on the web site.

- To add a standard folder, right-click on the project or folder under which you are going toadd the folder in the Solution Explorer and choose New Folder.
- To add an ASP.NET folder, right-click on the project in the Solution Explorer and selectthe folder from the list.
- To add an existing item to the site, right-click on the project or folder under which youare going to add the item in the Solution Explorer and select from the dialog box.

Projects and Solutions

A typical ASP.NET application consists of many items: the web content files (.aspx), source files (.cs files), assemblies (.dll and .exe files), data source files (.mdb files), references, icons, user controls and miscellaneous other files and folders. All these files that make up thewebsite are contained in a Solution.

When a new website is created, dotnet environment automatically creates the solution and displays it in the solution explorer.

Solutions may contain one or more projects. A project contains content files, source files, andother files like data sources and image files. Generally, the contents of a project are compiledinto an assembly as an executable file (.exe) or a dynamic link library (.dll) file.

Typically a project contains the following content files:

- Page file (.aspx)
- User control (.ascx)
- Web service (.asmx)
- Master page (.master)
- Site map (.sitemap)
- Website configuration file (.config)

Building and Running a Project:

You can execute an application by:

- Selecting Start
- Selecting Start Without Debugging from the Debug menu
- pressing F5
- Ctrl-F5

The program is built meaning, the .exe or the .dll files are generated by selecting a commandfrom the Build menu.

ASP.NET Application Life Cycle

The application life cycle has the following stages:

1. User makes a request for accessing application resource, a page. Browser sends this request tothe web server.

2. A unified pipeline receives the first request and the following events take place:

i. An object of the class ApplicationManager is created.

ii. An object of the class HostingEnvironment is created to provide information regardingthe resources. iii. Top level items in the application are compiled.

3. Response objects are created. The application objects such as HttpContext, HttpRequest andHttpResponse are created and initialized.

4. An instance of the HttpApplication object is created and assigned to the request.

5. The request is processed by the HttpApplication class. Different events are raised by this classfor processing the request.

ASP.NET Page Life Cycle

The page life cycle phases are:

- Initialization
- Instantiation of the controls on the page
- Restoration and maintenance of the state
- Execution of the event handler codes
- Page rendering

EVENT HANDLING

An event is an action or occurrence such as a mouse click, a key press, mouse movements, or any system-generated notification. A process communicates through events. For example, interrupts are system-generated events. When events occur, the application should be able to respond to it and manage it.

Events in ASP.NET are raised at the client machine and handled at the server machine. For example, a user clicks a button displayed in the browser. A Click event is raised. The browser handles this client-side event by posting it to the server.

Event Arguments

ASP.NET event handlers generally take two parameters and return void. The first parameter represents the object raising the event and the second parameter is event argument. The general syntax of an event is:

private void EventName (object sender, EventArgs e);

Application and Session Events

The most important application events are:

- Application_Start - It is raised when the application/website is started
- Application_End - It is raised when the application/website is stopped. Similarly, themost used Session events are
  - Session_Start - It is raised when a user first requests a page from the application.
  - Session_End - It is raised when the session ends.

Page and Control Events

Common page and control events are:

- DataBinding – It is raised when a control binds to a data source.
- Disposed – It is raised when the page or the control is released.
- Error - It is a page event, occurs when an unhandled exception is thrown.
- Init – It is raised when the page or the control is initialized.
- Load – It is raised when the page or a control is loaded.
- PreRender – It is raised when the page or the control is to be rendered.
- Unload – It is raised when the page or control is unloaded from memory.

Event Handling Using Controls:

All ASP.NET controls are implemented as classes, and they have events which are fired when a user performs a certain action on them. For example, when a user clicks a button the 'Click' eventis generated. For handling events, there are in-built attributes and event handlers. Event handleris coded to respond to an event and take appropriate action on it.

The event handler for the Click event:

```
Protected Sub btnCancel_Click(ByVal sender As Object, ByVal e As System.EventArgs)
Handles btnCancel.Click

End Sub
```

Example: Write a Program to calculate factorial of number

```
Partial Class _Default
Inherits System.Web.UI.Page
Protected Sub Button1_Click(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Button1.Click
Dim i As Integer
Dim a As Double
Dim f As Double
f = 1
i = 1
a = TextBox1.Text
While i <= a
f = f * i
i = i + 1
End While
Label3.Text = f & "unit"
End Sub
End Class
```

Write a Program to bind data using template in data list

```
Imports System.Data.SqlClient
Partial Class _Default
Inherits System.Web.UI.Page
Dim constr As String =
ConfigurationManager.ConnectionStrings("DatabaseConnectionString1").Connection  String
Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles
Me.Load
Dim conpubs As SqlConnection Dim cmdselect As SqlCommand Dim dtrAuthor As
SqlDataReader
conpubs = New SqlConnection(constr)
cmdselect = New SqlCommand("Select * from Author", conpubs) conpubs.Open()
dtrAuthor = cmdselect.ExecuteReader() dtrlstAuthor.DataSource = dtrAuthor
dtrlstAuthor.DataBind() dtrAuthor.Close()
conpubs.Close()
End Sub
End Class
```

Assignment 5

1. Write a Program to convert temperature from Fahrenheit to Celsius or vice versa.
2. Write a Program to display the selected date in the calendar
3. Write a Program to display the Difference between the two dates in the calendar.

Assignment Evaluation

0: Not Done [ ] **1: Incomplete [ ]** **2: Late Complete [ ]**

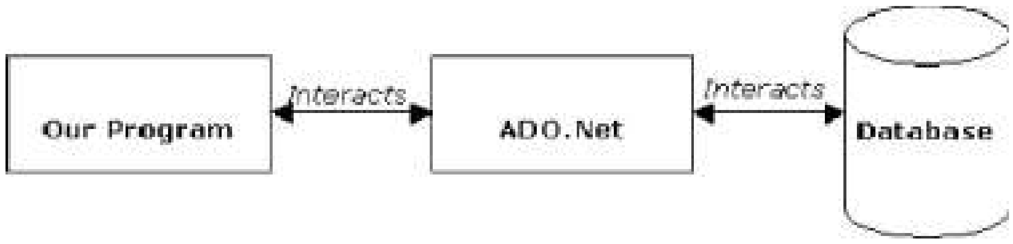
3: Need Improvement [ ] **4: Complete [ ]** **5: Well Done [ ]**

Signature of Instructor

ASSIGNMENT -E: ADO .NET

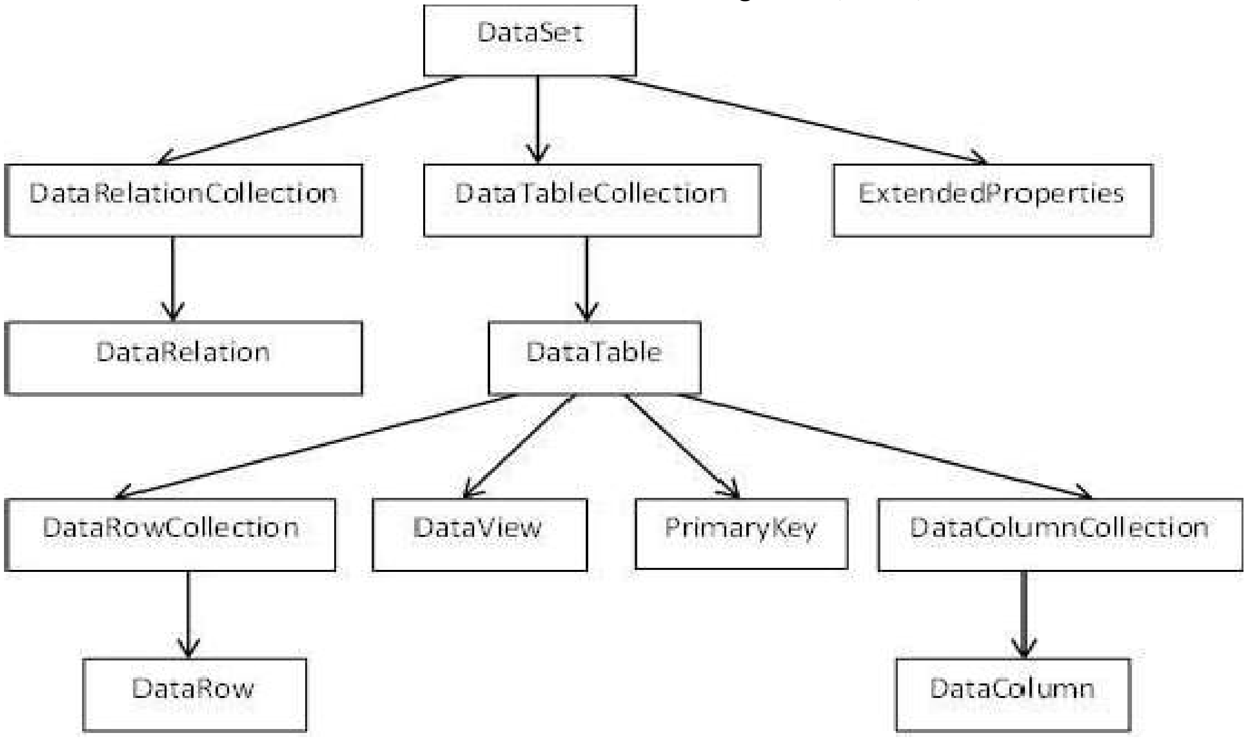
A data-access technology that enables applications to connect to data stores and manipulate datacontained in them in various ways.

An object oriented framework that allows you to interact with database systems



ADO.NET components

- 1) Dataset
- 2) Data Provider
- **DataSet** is an in-memory representation of data. It is a disconnected, cached set of records that are retrieved from a database.
- When a connection is established with the database, the data adapter creates a dataset andstores data in it. After the data is retrieved and stored in a dataset, the connection with thedatabase is closed. This is called the 'disconnected architecture'. The dataset works as a virtual database containing tables, rows, and columns.



Connecting to a Database

- The .Net Framework provides two types of Connection classes −
- **SqlConnection** − designed for connecting to Microsoft SQL Server.
- **OleDbConnection** − designed for connecting to a wide range of databases, likeMicrosoft Access and Oracle.

Steps :

- We have a table stored in Microsoft SQL Server, named Customers, in a database named testDB.
- Select TOOLS → Connect to Database
- Select a server name and the database name in the Add Connection dialog box.
- Click on the Test Connection button to check if the connection succeeded.
- Add a DataGridView on the form.
- Click on the Choose Data Source combo box. Click on the Add Project Data Source link.
- This opens the Data Source Configuration Wizard.
  Select Database as the data source type
- Choose DataSet as the database model.
- Choose the connection already set up.
- Save the connection string.
- Choose the database object, Customers table in our example, and click the Finish button.
- When the application is run using Start button available at the Microsoft Visual Studio tool bar, it will show the following window

2) Data Provider:

A data provider is used for connecting to a database, executing commands and retrieving data,storing it in a dataset, reading the retrieved data and updating the database.

The data provider in ADO.Net consists of the following four objects –

1) Connection

This component is used to set up a connection with a data source.

2) Command

A command is a SQL statement or a stored procedure used to retrieve, insert, delete or modifydata in a data source.

3)DataReader

Data reader is used to retrieve data from a data source in a read-only and forward-only mode.

4) DataAdapter

This is integral to the working of ADO.Net since data is transferred to and from a database through a data adapter. It retrieves data from a database into a dataset and updates the database. When changes are made to the dataset, the changes in the database are actually done by the data adapter.

There are following different types of data providers included in ADO.Net

1) The .Net Framework data provider for SQL Server - provides access to Microsoft SQLServer.

2) The .Net Framework data provider for OLE DB - provides access to data sources exposed byusing OLE DB.

3) The .Net Framework data provider for ODBC - provides access to data sources exposed byODBC.

4) The .Net Framework data provider for Oracle - provides access to Oracle data source.

5) The Entity Client provider - enables accessing data through Entity Data Model (EDM)applications.

CONNECTIONS:

ADO.NET provids both Sqlconnection and OLEDB connection classes for use with datasource.

1) Sql Connection :-

To declare and instantiate Sql connection,u can use

foll.stmtDim con AS Sqlconnection

con = New SqlConnection("server=YASH;database=test;integrated security=true")

2) OleDbConnection

Dim con As

OleDbConnectionDim con

As OleDbConnection

con = New OleDbconnection ("Provider=Microsoft.Jet.OLEDB.4.0;Data Source=yourdatabasename.mdb")

Command

We use command object to tell database about operation we need to perform.The typical command on database will be :-

1) Select
2) Insert
3) Delete
4) Update

EXAMPLE:creating OLEDB Data connection

```
  imports
  System.Data.OleDb
  Public Class Form1
   Dim cn As New OleDbConnection
   Private Sub Form1_Load(ByVal sender As System.Object, ByVal
e AsSystem.EventArgs) Handles MyBase.Load

       cn = New
OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=C:\Users\user\Documents\Database5.accdb")
       cn.Open()
       MsgBox("oledb connection is successfully
 established")End Sub
End Class
```

Data Adapter

- DataAdapter is a part of the ADO.NET Data Provider. DataAdapter provides the communication between the Dataset and the Datasource. We can use the DataAdapter in combination with the DataSet Object. That is these two objects combine to enable both data access and data manipulation capabilities.

- The DataAdapter can perform Select , Insert , Update and Delete SQL operations in the Data Source. The Insert , Update and Delete SQL operations , we are using the continuation of the Select command perform by the DataAdapter. That is the DataAdapter uses the Select statements to fill a DataSet and use the other three SQL commands (Insert, Update, delete) to transmit changes back to the Database

- Types of DataAdapter

1) SqlDataAdapter

2) OleDbDataAdapter

Types of DataAdapter

1) SqlDataAdapter :-

```
Dim ad as SqlDataAdapter
  ad=new  SqlDataAdapter
  (cmd)
```

2) OleDbDataAdapter :-

```
  Dim ad as OleDbDataAdapter
  ad=new OleDbDataAdapter
  (cmd)
```

Example

```
Imports
    System.Data.OleDb
    Public Class Form1
      Dim cn As New
      OleDbConnection Dim cmd As
      New OleDbCommandDim ad
      As New OleDbDataAdapter
      Dim dt As New DataTable
        Private Sub Form1_Load(ByVal sender As System.Object, ByVal
        e AsSystem.EventArgs) Handles MyBase.Load
          'TODO: This line of code loads data into the 'Database5DataSet.doctor' table. You
    canmove, or remove it, as needed.
```

```
Me.DoctorTableAdapter.Fill(Me.Database5DataSet.doctor)
```

```
        cn = New
OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=C:\Users\user\Documents\Database5.accdb")

        cn.Open()

        MsgBox("oledb    connection    is    successfully

        established") cmd = New OleDbCommand("select *

        from doctor", cn)ad = New OleDbDataAdapter(cmd)

        ad.Fill(dt)

        DataGridView1.DataSource

        = dt

 End

Sub

End

Class
```

Dataset

- The DataSet contains the copy of the data we requested through the SQL statement. Wecan use Dataset in combination with SqlDataAdapter class . The SqlDataAdapter objectallows us to populate Data Tables in a DataSet.

Example

```
 Imports

 System.Data.OleDb

 Public Class Form1

Dim cn As New

OleDbConnection Dim cmd As

New OleDbCommandDim ad

As New OleDbDataAdapter

Dim ds As New DataSet()

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal
 e AsSystem.EventArgs) Handles MyBase.Load
```

```
        Me.DoctorTableAdapter.Fill(Me.Database5DataSet.doctor)

        cn = New
OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=C:\Users\user\Documents\Database5.accdb")
```

```
        cn.Open()

        MsgBox("oledb connection is successfully

    established")adp = New OleDbDataAdapter(cmd)

        adp = New OleDbDataAdapter("select * from doctor",

       cn)ad.Fill(ds, "doctor")

        DataGridView1.DataSource = ds

       DataGridView1.DataMember =

       "doctor" End Sub

    End Class
```

Data Reader

- To retrieve data using a DataReader, create an instance of the Command object, and then create a DataReader bycalling Command.ExecuteReader to retrieve rows from a data source.
- DataReader Object in ADO.NET is a stream-based , forward-only, read-only retrieval ofquery results from the Data Source, which do not update the data. The DataReader cannotbe created directly from code, they created only by calling the Execute Reader method of a Command Object.

```
DataReader = Command.ExecuteReader()
```

DataTable

- DataTable represents relational data into tabular form. ADO.NET provides a DataTable class to create and use data table independently. It can also be used with DataSet also. Initially, when we create DataTable, it does not have table schema. We can create table schema by adding columns and constraints to the table. After defining table schema, we can add rows to the table.
- We must include System.Data namespace before creating DataTable.

Example

```
    Imports System.Data.

    DataTablePublic Class

    Form1

  Dim table As New DataTable("table")
```

```
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal
e AsSystem.EventArgs) Handles MyBase.Load

  table.Columns.Add("ID", Type.GetType("System.Int32"))

  table.Columns.Add("FIRST NAME",

  Type.GetType("System.Int32"))table.Columns.Add("LAST

  NAME", Type.GetType("System.Int32"))

  table.Columns.Add("SUM", Type.GetType("System.Int32"))

  DataGridView1.DataSource = table

End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click

  DataGridView1.DataSource =

    tableEnd Sub
```

Data Validation

- *Data validation* is the process of ensuring, at least as far as is possible, that the data givento a program by a user or from a file is of the correct type, and in the correct format.
- Although the programmer will obviously take every precaution to ensure the correct operation of the program, and will attempt to eliminate bugs that could cause a problem through a rigorous process of testing, they have no real control over mistakes made by theuser during data entry. Nor can they guarantee that any data files used by the program will be free of errors and in the correct format.
- Validation is a form of self protection. It is the checking to verify that appropriate values have been entered for textbox or any control .

Validation may include :-

1) Making sure data is numeric or text depending on application.
2) Checking for specific values.
3) Checking for range of values
4) Making sure required items are entered.

Write a program to access data source through ADO.NET.

```
using System;
using
System.Data;
using
System.Configuration;
using
System.Collections;
using System.Web;
using
System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using
System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Data.OleDb;

public partial class marksheet : System.Web.UI.Page
{
        OleDbConnection con = new OleDbConnection("Provider=MSDAORA; User Id=result;
Password=college");
        OleDbCommand cmd = new
        OleDbCommand(); OleDbCommand cmd1 =
        new OleDbCommand();OleDbDataReader
        dr,dr1;
        protected void Page_Load(object sender, EventArgs e)
        {
      cmd.Connection =
      con;
      cmd1.Connection =
      con;
        con.Open();

        string s1 =
        Session["rollno"].ToString(); string
        s2 = Session["sem"].ToString();
        string s3 =
        Session["branch"].ToString();

      cmd.CommandText = "select * from DATABASE where ROLLNO='"+s1+"' and
SEM='"+s2+"' and BRANCH='"+s3+"' ";
        dr =
cmd.ExecuteReader();if
(dr.Read())
        {


         string                d1              =
        dr.GetValue(0).ToString();    string
```

```
d2 = dr.GetValue(1).ToString();
string d3 =
dr.GetValue(2).ToString(); string
d4 = dr.GetValue(3).ToString();
string d5 =
dr.GetValue(4).ToString(); string
d6 = dr.GetValue(5).ToString();
string d7 =
dr.GetValue(6).ToString(); string
d8 = dr.GetValue(7).ToString();
```

```
            TextBox1.Text =
            d1;
            TextBox2.Text =
            d2;
            TextBox3.Text =
            d3;Label16.Text
            = d4;
            Label17.Text =
            d5; Label18.Text
            = d6;
            Label19.Text =
            d7; Label20.Text
            = d8;
            dr.Dispose();
    }
    cmd1.CommandText = "select * from SEM_BRANCH_SUB where sem='"+s2+"' and
    branch='"+s3+"' ";
          OleDbDataReader dr1 =
           cmd1.ExecuteReader();if (dr1.Read())
           {
           string d9 =
           dr1.GetValue(2).ToString(); string
           d10 = dr1.GetValue(3).ToString();

           string          d11          =
           dr1.GetValue(4).ToString();   string
           d12  =  dr1.GetValue(5).ToString();
           string          d13          =
           dr1.GetValue(6).ToString();

            Label10.Text =
     d9;  Label11.Text   =
     d10;
            Label12.Text  =
            d11;
            Label13.Text  =
            d12;
            Label14.Text  =
            d13;

           }

           }
     }
```

Database programs with ASP.NET and ADO.NET Create a Login Module which adds Usernameand Password in the database. Username in the database should be a primary key.

CODE:

LoginModule.

aspxusing

System;

using

System.Collections.Generic;

using System.Linq;

```
using
System.Web;
using
System.Web.UI;
using System.Web.UI.WebControls;
public partial class LoginModule : System.Web.UI.Page
{
protected void Page_Load(object sender, EventArgs e)
{
}
protected void btnSignUp_Click(object sender, EventArgs e)
{
SqlDataSource1.InsertParameters["Username"].DefaultValue = txtUserName.Text;
SqlDataSource1.InsertParameters["Password"].DefaultValue = txtPassword.Text;
SqlDataSource1.Insert();
lblResult.Text = "User Added";
}
}
```

Assignment: 6

1. Create a web application to insert 3 records inside the SQL database table having following fields(DeptId, DeptName, EmpName, Salary). Update the salary for any oneemployee and increment it to 15% of the present salary. Perform delete operation on 1 row of the database table.
2. Create the table with the given fields. FIELD NAME DATA TYPE (EmpNo number EmpName varchar EmpSal number EmpJob varchar EmpDeptNo number)For the giventable design a web page to display the employee information from table to grid control.
3. Create the table with the given fields. FIELD NAME DATA TYPE SRollno int SName string SAddress string SFees int For the given table design a web page to display the employee information from table to grid control.

4. Write a program to connect to the master database in SQL Server, in the Page_Loadevent. When the connection is established, the message

“Connection has been established” should be displayed in a label in the form.

Semester -V	**Paper -IV**
Course Code: BBACA-606 T	**RECENT TRENDS IN INFORMATION TECHNOLOGY**
Credits: 04	**Total Lectures: 60**

Prerequisites:

1. Basic knowledge of related programming and database concepts.
2. Fundamentals of Mathematical logic & Data structures.

Course Objectives

1. To introduce upcoming trends in Information technology.
2. To study Eco friendly software development concepts.
3. To provide a strong foundation of fundamental concepts in Artificial Intelligence.
4. To evaluate the performance of various data mining task.
5. To understand Data analytics using Spark Programming.

Course Outcomes: On completion of the course, student will be able

1. To discuss the basic concepts AI.
2. To apply basic, intermediate and advanced techniques to mine the data.
3. To provide an overview of the concept of Spark programming.

UNIT 1 Distributed Databases 10

1.1. Standalone v/s Distributed databases, Replication, Fragmentation, Client / Server architecture, types of distributed databases

1.2. Object – Relational Databases Abstract Data types, Nested Tables, Varying Arrays, Large Objects, Naming Conventions for Objects, Case Study

UNIT 2. Data Warehouse 15

2.1. What is Data Warehouse? , A Multidimensional Data Model, Data Warehouse Architecture, Data Warehouse Implementation, Data cube Technology,

2.2. OLAP Vs. OLTP, OLAP Operations

Types of OLAP Servers: ROLAP versus MOLAP versus HOLAP

2.3. From Data Warehousing to Data Mining, Data Mining, Functionalities, Data Cleaning, Data Integration and Transformation, Data Reduction

2.4. Accuracy Measures: Precision, recall, F-measure, confusion matrix, cross-validation, bootstrap

2.5. Data Mining Techniques : Frequent item-sets and Association rule mining: Apriori algorithm, FP tree algorithmGraph Mining: Frequent sub-graph mining

Software for data mining : R, Weka, Sample applications of data mining

Introduction to Text Mining, Web Mining, Spatial Mining,

Temporal Mining

UNIT 3 Network Security 15

3.1. Cryptography; Introduction to Cryptography, Substitution Ciphers, Transposition Ciphers, One-Time Pads, Two Fundamental Cryptographic Principles; Symmetric Key Algorithms; DES-The Data

3.2. Encryption Standards, AES – The Advances Encryption Standard; Public Key algorithms; RSA, Other Public Key algorithms; Digital Signatures, Symmetric-Key Signature, Public key Signature, Message Digests

UNIT 4 Computing and Informatics 10

4.1. Introduction to computing, Types of computing: Cloud, Green, Soft,

UNIT 5. Spark

5.1. Introduction to Apache Spark

5.2. Spark Installation

5.3. Apache Spark Architecture

5.4. Components of Spark

5.5. Spark RDDs

5.6. RDD Operations: Transformation & Actions

5.7. Spark SQL and Data Frames

5.8. Introduction to Kafka for Spark Streaming

Reference Readings :

1. Data Mining Concepts and Techniques, by Jiawei Micheline Kamber, Morgan Kauf Mann Publishers.
2. Advanced Analytics with Spark by Sandy RyzaPublicatio O'REILLY
3. Apache Spark for Data Science Cookbook by Padma Priya Chitturi
4. Jiawei Micheline Kamber, "Data Mining Concepts and Techniques",Morgan Kauf Mann, Publishers.
5. William Stallings, "Network Security Essentials", Prentice-Hall.

Semester -VI	**Paper -VI**
Course Code: BBACA-607 T (A)	**Interpersonal skills and Professional Ethics**
Credits: 02	**Total Lectures: 30**

COURSE OBJECTIVES

1. To create an awareness in Computer Field & Ethics in Computer profession.
2. To understand professional responsibility of Computer professionals
3. To appreciate ethical dilemma while discharging duties in professional life.

UNIT 1 - Interpersonal skills with pronunciation and communicative skills 10

1.1. Introduction to Body language as part of communicative skills

1.2. Introduction to group dynamics and group effectiveness and leadership

1.3. Various steps to give a successful oral presentation

1.4. Making Decision & Emotional Intelligence with Group Discussion

Unit 2: Introduction to Professional Ethics: 10

2.1. Basic concepts, Governing Ethics, Personal & Professional Ethics, Ethical dilemmas, Life Skills, Emotional Intelligence,

2.2. Value education, Professional accountability, Ethics & Profession,

Unit 3 : Computer profession Ethics 10

3.1. Computer profession; ethical issues in Computer Field;

3.2. Conflicts between business demands and professional ideals;

3.3. Social and ethical responsibilities of the Computer Professional; codes of professional ethics.

Lab Assignments