

**S.Y.B.C.A. (Science) A. Y. 2022-23 Department of Computer Applications**

**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 Computer Applications (B.C.A.) Science**

**Syllabus of**

**S. Y. B. C.A. Science**

**Implemented from**

**Academic Year 2022 - 23**

New Arts, Commerce and Science College Ahmednagar (Autonomous)

**Ahmednagar Jilha Maratha Vidya Prasarak Samaj's  
New Arts, Commerce and Science College, Ahmednagar  
(Autonomous)**

**Board of Studies in B.C.A Science**

Sr. No.	Name	Designation
1.	Prof.Arun.D.Gangarde	Chairman
2.	Prof. Priyamvada Patil	Member
3.	Dr.Shraddha Ingale	Member
4.	Dr.Mudassar Shaikh	Member
5.	Dr.Santosh Khamitkar	Academic Council Nominee
6.	Dr. Shankar Mali	Academic Council Nominee
7.	Dr.Nitin Patil	Vice-Chancellor Nominee
8.	Mr.Summit Suryawanshi	Alumni
9.	Dr.Deepak Shikarpur	Industry Expert
10.	Prof.Deepali Jagdale	Member (co-opt)
11.	Dr. Madhukar Shelar	Member (co-opt)

**S.Y.B.C.A. (Science) A. Y. 2022-23 Department of Computer Applications****Programme Structure and Course Titles: (All academic years)**

<b>Sr. No.</b>	<b>Class</b>	<b>Semester</b>	<b>Course Code</b>	<b>Course Title</b>	<b>Credits</b>
1.	F.Y.B.C.A .	I	BCA-SC 101 T	Principles of Programming and C Programming Fundamentals	04
2.	F.Y.B.C.A .	I	BCA-SC 102 T	Fundamentals of Computers	04
3.	F.Y.B.C.A .	I	BCA-SC 103 T	Web Designing -I	04
4.	F.Y.B.C.A .	I	BCA-SC 104 T	Mathematics and statistical methods for Computer Science	04
5.	F.Y.B.C.A .	I	BCA-SC 105 P	Fundamentals of Computers Lab	1.5
6.	F.Y.B.C.A .	I	BCA-SC 106 P	Principles of Programming and C Programming Fundamentals Lab	1.5
7.	F.Y.B.C.A .	I	BCA-SC 107 P	Web Designing –I Lab	1.5
8.	F.Y.B.C.A .	I	BCA-SC 108 P	Mathematics and statistical methods Lab	1.5
9.	F.Y.B.C.A .	II	BCA-SC 201 T	Advance C Programming	04
10.	F.Y.B.C.A .	II	BCA-SC 202 T	Database Management Systems	04
11.	F.Y.B.C.A .	II	BCA-SC 203 T	Web Designing-II	04
12.	F.Y.B.C.A .	II	BCA-SC 204 T	Software Engineering	04
13.	F.Y.B.C.A .	II	BCA-SC 205 P	Advance C Programming Lab	1.5
14.	F.Y.B.C.A	II	BCA-SC 206 P	Database Management Systems Lab	1.5

**S.Y.B.C.A. (Science) A. Y. 2022-23 Department of Computer Applications**

15.	F.Y.B.C.A	II	BCA-SC 207 P	Web Designing-II Lab	1.5
16.	F.Y.B.C.A	II	BCA-SC 208 P	Software Engineering Lab	1.5
17.	S.Y.B.C.A	III	BCA-SC 301 T	Object Oriented Programming concept using C++	04
18.	S.Y.B.C.A	III	BCA-SC 302 T	Data Structures using C	04
19.	S.Y.B.C.A	III	BCA-SC 303 T	Advance Database Management Systems	04
20.	S.Y.B.C.A	III	BCA-SC 304 P	Object Oriented Programming concept using C++ (Lab)	02
21.	S.Y.B.C.A	III	BCA-SC 305 P	Data Structures using C (Lab)	02
22.	S.Y.B.C.A	III	BCA-SC 306 P	Advance Database Management Systems (Lab)	02
23.	S.Y.B.C.A	III	BCA-SC 307 T	React JS	02
24.	S.Y.B.C.A	III	BCA-SC 308 P	React JS (Lab)	02
25.	S.Y.B.C.A	III	309 T	MIL : Technical English	02
26.	S.Y.B.C.A	III	310 T	Critical Thinking and Scientific Temper	02
27.	S.Y.B.C.A	IV	BCA-SC 401 T	Core Java Programming	04
28.	S.Y.B.C.A	IV	BCA-SC 402 T	Python Programming	04
29.	S.Y.B.C.A	IV	BCA-SC 403 T	Operating System Concepts	04

**S.Y.B.C.A. (Science) A. Y. 2022-23 Department of Computer Applications**

30.	S.Y.B.C.A	IV	BCA-SC 404 P	Core Java Programming(Lab)	02
31.	S.Y.B.C.A	IV	BCA-SC 405 P	Python Programming(Lab)	02
32.	S.Y.B.C.A	IV	BCA-SC 406 P	Operating System Concepts ( Lab)	02
33.	S.Y.B.C.A	IV	BCA-SC 407 T	Cloud Computing	02
34.	S.Y.B.C.A	IV	BCA-SC 408 P	Cloud Computing Lab	02
35.	S.Y.B.C.A	IV	409 T(A)*	MIL: Foreign Language: Japanese	02
36.	S.Y.B.C.A	IV	409 T(B)*	MIL: Foreign Language: French	02
37.	S.Y.B.C.A	IV	410 T	Environmental Awareness	02
38.	T.Y.B.C.A	V	BCA-SC 501 T (A/B/C)	DSEC- 1(Group A/B/C)	04
39.	T.Y.B.C.A	V	BCA-SC 502 T (A/B/C)	DSEC- 2(Group A/B/C)	04
40.	T.Y.B.C.A	V	BCA-SC 503 T (A/B/C)	DSEC- 3(Group A/B/C)	04
41.	T.Y.B.C.A	V	BCA-SC 504 P (A/B/C)	DSEC- 4 Practical(Group A/B/C)	02
42.	T.Y.B.C.A	V	BCA-SC 505 P (A/B/C)	DSEC- 5 Practical(Group A/B/C)	02
43.	T.Y.B.C.A	V	BCA-SC 506 P (A/B/C)	DSEC- 6 Practical(Group A/B/C)	02
44.	T.Y.B.C.A	V	BCA-SC 507 T	SEC -01 *	02
45.	T.Y.B.C.A	V	BCA-SC 508 P	SEC - 02 Practical *	02
46.	T.Y.B.C.A	VI	BCA-SC 601 T (D/E/F)	DSEC- 7(Group D/E/F)	04
47.	T.Y.B.C.A	VI	BCA-SC 602 T (D/E/F)	DSEC- 8 (Group D/E/F)	04

**S.Y.B.C.A. (Science) A. Y. 2022-23 Department of Computer Applications**

48.	T.Y.B.C.A	VI	BCA-SC 603 T (D/E/F)	DSEC- 9(Group D/E/F)	04
49.	T.Y.B.C.A	VI	BCA-SC 604 P (D/E/F)	DSEC- 10 Practical (Group D/E/F)	02
50.	T.Y.B.C.A	VI	BCA-SC 605 P (D/E/F)	DSEC- 11 Practical (Group D/E/F)	02
51.	T.Y.B.C.A	VI	BCA-SC 606 P (D/E/F)	DSEC- 12 Practical (Group D/E/F)	02
52.	T.Y.B.C.A	VI	BCA-SC 607 T	SEC -03 *	02
53.	T.Y.B.C.A	VI	BCA-SC 608 Pr	SEC - 04 Project *	02

Ahmednagar Jilha Maratha Vidya Prasarak Samaj's

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

Syllabus of S. Y. B.C.A. Science

Under

Faculty of Science

<b>Semester – III</b>	<b>Paper – I</b>
<b>Course Code: BCA-SC 301 T</b>	<b>Title of the Course: Object Oriented Programming concepts using C++</b>
<b>Credits: 04</b>	<b>Total Lectures: 60 Hrs.</b>

### Course Outcomes (COs):

- Compare and contrast procedural and object oriented programming
- Apply principles of OOPs
- Design and develop applications using object oriented programming language C++

### Detailed Syllabus:

#### Unit I: Introduction to C++ (06)

- 1.1 How C++ evolved from C?
- 1.2 Features of C++
- 1.3 Procedure-oriented programming Vs Object-oriented programming
- 1.4 The basic anatomy of a C++ program, Simple "Hello World" program
- 1.5 Compiling, linking and running a C++ program.

#### Unit II : Object Oriented Programming Concepts (08)

- 2.1 Classes and objects
- 2.2 Abstraction
- 2.3 Inheritance
- 2.4 Polymorphism
- 2.5 Data Binding
- 2.6 Encapsulation
- 2.7 Message Passing

**Unit III: Programming in C++ (08)**

- 3.1 Data Types
- 3.2 Keywords and new Operator
- 3.3 Type Casting in C++
- 3.4 Reference Variables
- 3.5 Classes and Access Specifiers
- 3.6 Defining Data Members and Member Functions
- 3.7 Arrays and Array of objects
- 3.8 Usage of namespace, Usage of Manipulators, Managing Console I/O

**Unit IV: Constructors, Destructors, Function and Operator Overloading (14)**

- 4.1 Constructor & Destructor ,Types, Static Members
- 4.2 Function
  - 4.2.1 Call by reference, Call by value
  - 4.2.2 Inline Function
  - 4.2.3 Friend Function
- 4.4 Overloading
  - 4.4.1 Function Overloading
  - 4.4.2 Operator Overloading
    - 4.4.2.1 Overloading unary and binary operators  
(with member function and with friend function)
    - 4.4.2.2 Overloading using friend functions,Overloading “<< >>” operator ,  
Type Conversion
- 4.5 Usage of this pointer

**Unit V: Inheritance (14)**

- 5.1 Introduction



5.2 Types of Inheritance – Single , Multiple,Multilevel,Heirarchical,Hybrid

5.3 Ambiguity in Multiple Inheritance, Virtual base class ,Abstract base class

5.4 Overriding - Virtual Functions and Pure Virtual Functions

**Unit VI : Files and Exception Handling**

**(10)**

6.1 Introduction

6.2 Classes for file stream operations

6.3 Opening and closing a file

6.4 File Updation with Random Access

6.5 Introduction to Exception handling

6.5.1 Basics of Exception handling

6.5.2 Exception handling Mechanism –try, catch and throw block

**Suggested Readings:**

1. Object Oriented Programming (C++) Balaguruswamy, McGraw Hill Education; Seventh edition
2. Deitel, —C++ How to Program, 4th Edition, Pearson Education
3. The Complete Reference C++ by Herbert Schildt, McGraw Hill Education; 4 edition
4. Mastering C++ by Venugopal, T Ravishankar, McGraw Hill Education; 2 edition

Ahmednagar Jilha Maratha Vidya Prasarak Samaj's

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

**Syllabus of S. Y. B.C.A. Science**

Under

**Faculty of Science**

<b>Semester – III</b>	<b>Paper – II</b>
<b>Course Code: BCA-SC 302 T</b>	<b>Title of the Course: Data Structures using C</b>
<b>Credits: 04</b>	<b>Total Lectures: 60 Hrs.</b>

**Course Outcomes (COs):**

- To understand the concepts of ADTs
- To understand sorting, searching and hashing algorithms
- To learn linear data structures – lists, stacks, and queues
- To apply Tree and Graph structures

**Detailed Syllabus:**

**Unit I: Basic Concept and Introduction to Data Structure (05)**

- 1.1 Algorithm-Definition and characteristics
- 1.2 Algorithm Analysis -Space Complexity ,Time Complexity , Asymptotic Notation
- 1.3 Introduction to Data structure
- 1.4Types of Data structures
- 1.5 Abstract Data Types (ADT)

**Unit II: Arrays (08)**

- 2.1 Introduction to Arrays
- 2.2 Types of array and Representation of array
- 2.3 Sorting Techniques with efficiency - Bubble sort, Insertion sort, Merge sort, Quick Sort, Selection Sort
- 2.3 Searching Techniques –Linear Search, Binary search

**Unit III: Linked List (12)**

- 3.1 Introduction to Linked List
- 3.2 Implementation of Linked List
- 3.3 Types of Linked List
  - 3.3.1 Singly Linked List
  - 3.3.2 Doubly Linked List
  - 3.3.3 Circular - Singly Circular Linked List, Doubly Circular Linked List
- 3.4 Operations on Linked List-Create,Display,Insert,Delete,Reverse,Search,Sort,Concatenation
- 3.5 Applications of Linked List –Polynomial Equation

**Unit IV: Stack** (08)

- 4.1 Introduction
- 4.2 Representation- Static & Dynamic
- 4.3 Primitive Operations on stack
- 4.4 Application of Stack
  - 4.4.1 Evaluation of postfix and prefix
  - 4.4.2 Conversion of Infix, prefix, postfix
  - 4.4.3 Recursion and Tower of Hanoi

**Unit V: Queue** (06)

- 5.1 Introduction
- 5.2 Representation - Static & Dynamic
- 5.3 Primitive Operations on Queue
- 5.4 Circular queue, priority queue
- 5.5 Concept of doubly ended queue

**Unit VI: Trees** (12)

- 6.1 Concept & Terminologies
- 6.2 Types of trees
- 6.3 Representation – Static and Dynamic
- 6.4 Operations on Binary Tree and Binary Search Tree – Create, Insert, Delete, , Counting leaf nodes and non-leaf nodes, total number of nodes
- 6.5 Tree Traversals (preorder, inorder, postorder)
- 6.6 Height balanced tree- AVL trees, Rotations, AVL tree examples.

**Unit VII: Graph** (09)

- 7.1 Concept & terminologies
- 7.2 Graph Representation – Adjacency matrix, adjacency list,
- 7.3 Graph Traversal – Breadth First Search and Depth First Search
- 7.4 Applications
  - 7.4.1 Activity on Vertex network – Topological sort
  - 7.4.2 Activity On Edge – critical Path

**Suggested Readings:**

1. Fundamentals of Data Structures ---By Horowitz Sahani (Galgotia)
2. Data Structures using C and C++ --By YedidyahLangsam, Aaron M. Tenenbaum, Moshe J. Augenstein
3. Introduction to Data Structures using C By Ashok Kamthane
4. Data Structures using C -- Bandopadhyay&Dey (Pearson)
5. Data Structures using C --By Srivastava BPB Publication

**Ahmednagar Jilha Maratha Vidya Prasarak Samaj's**  
**New Arts, Commerce and Science College, Ahmednagar (Autonomous)**

**Syllabus of S. Y. B.C.A. Science**

**Under**

**Faculty of Science**

<b>Semester – III</b>	<b>Paper – III</b>
<b>Course Code: BCA-SC 303 T</b>	<b>Title of the Course: Advance Database Management Systems</b>
<b>Credits: 04</b>	<b>Total Lectures: 60 Hrs.</b>

**Course Outcomes (COs):**

- a. Formulate SQL queries using advanced SQL features.
- b. Perform Database operations using PL/PostgreSQL
- c. Compare and contrast different concurrency control and recovery techniques.
- d. Apply mechanisms for database security.
- e. Analyze various database system architectures.

**Detailed Syllabus:**

**Unit I: Relational Database Design (15)**

- 1.1. PL/Postgresql: Language Structure
- 1.2. Controlling the Program Flow, Conditional Statements, Loops
- 1.3. Views
- 1.4. Functions
- 1.5. Handling Errors and Exceptions
- 1.6. Cursors
- 1.7. Triggers

**Unit II: Transaction Concepts (10)**

- 2.1 Transaction, Properties of Transaction, States of Transactions
- 2.2 Concurrent Execution of Transactions and Conflicting Operations
- 2.3 Schedules, Types of Schedules, Concept of Serializability, Precedence Graph for Serializability

**Unit Iii: Concurrency Control (14)**

- 3.1 Ensuring Serializability by Locks, Different Lock Modes
- 3.2 2PL And Its Variations

3.3 Multiple Granularity Locking Protocol

3.4 Basic Timestamp Method for Concurrency, Thomas Write Rule

3.5 Locks with Multiple Granularity, Dynamic Database Concurrency (Phantom Problem)

3.6 Timestamps versus Locking

3.7 Optimistic Concurrency Control Algorithm, Multi Version Concurrency Control

3.8 Deadlock Handling Methods –

3.8.1 Detection And Recovery (Wait For Graph).

3.8.2 Prevention Algorithms (Wound-Wait, Wait-Die)

3.8.3 Deadlock Recovery Techniques (Selection of Victim, Starvation, Rollback)

#### **Unit IV: Crash Recovery**

**(10)**

4.1 Transaction Failure Classification

4.2 Recovery Concepts

4.3 Checkpoints

4.4 Recovery with Concurrent Transactions (Rollback, Checkpoints, Commit)

4.5 Log Base Recovery Techniques (Deferred and Immediate Update)

4.6 Buffer Management

4.7 Database Backup and Recovery from Catastrophic Failures

4.8 Shadow Paging

#### **Unit V: Database Security**

**(06)**

5.1 Introduction to Database Security Concepts

5.2 Methods for Database Security

5.3 Discretionary Access Control Method

5.4 Mandatory Access Control and Role Based Access Control for Multilevel Security

5.5 Use of Views in Security Enforcement

#### **Unit VI: Database System Architectures**

**(05)**

6.1 Centralized and Client – Server Architectures

6.2 Server System Architectures

6.3 Introduction to Parallel Systems

6.4 Introduction to Distributed Systems

6.5 Introduction to Object Based Databases

6.6 Introduction to Web based databases

### **Suggested Readings:**

1. Database System Concepts – Abraham Silberschatz, Henry F. Korth, S. Sudarshan, 6th edition McGraw-Hill
2. Fundamentals of Database Systems- Ramez Elmasri, Shamkant B. Navathe, 6th edition– Pearson.
3. Database Management Systems -Raghu Ramakrishnan, Johannes Gehrke, 3rd edition, Tata McGraw Hill
4. Introduction to Database Management System- Bipin Desai, 3rd edition, Galgotia Publication.
5. An Introduction to Database Systems - C.J. Date, 7 th edition, Addison-Wesley
6. Practical PostgreSQL- Joshua D. Drake, John C Worsley, O'Reilly Publications

## Ahmednagar Jilha Maratha Vidya Prasarak Samaj's

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

## Syllabus of S. Y. B.C.A. Science

Under

Faculty of Science

Semester – III	Paper – IV
Course Code: BCA-SC 304 P	Title of the Course: Object Oriented Programming Concept using C++( Lab)
Credits: 02	Total Lectures: 30 Hrs.

**Course Outcomes (COs):**

- Compare and contrast procedural and object oriented programming
- Apply principles of OOP
- Design and develop applications using object oriented programming language C++

**Detailed Syllabus: Lab Course Contents**

- Write a C++ program to prompt the user to input her/his name and print this name on the screen, as shown below. The text from keyboard can be read by using cin>> and to display the text on the screen you can use cout<<.
- Write a program in C++ to display various type or arithmetic operation using mixed data type
- Write a C++ program to prompt the user to input 3 integer values and print these values in forward and reversed order, as shown below.
- Write a program to find greatest from three numbers. 5. Write a program of student mark sheet where Physics, Chemistry and Maths are the subject as input.
- Define a class "Bank" with following data members Acc-no, Name of Depositor, A/C Balance, Write member function to perform following task. Accept Details Display Details.
- Write a program in C++ to swap two numbers.
- Write a program in C++ to swap two numbers using reference variable.
- Write a program in C++ to test the Type Casting
- Write a C++ program to read an integer n and prints the factorial of n.
- Write a C++ program to compute the sum of the specified number of Prime numbers.
- Write a program in C++ to demonstrate the manipulators.
- Write a C++ program to demonstrate the use of Friend function in class.
- Write a C++ program to sort an Array in Ascending order.
- Write a C++ program to create a class Person which contains data members as P\_Name, P\_City, P\_Contact\_Number. Write member functions to accept and display five Persons information.
- Write a C++ program to create a class Student which contains data members as Roll\_Number, Stud\_Name, Marks in five subjects. Write member functions to accept

- Student information. Display all details of student along with a percentage and class obtained depending on percentage. (Use array of objects)
16. Create a C++ class for a student object with the following attributes—roll no, name, number of subjects, marks of subjects. Write member function for accepting marks and display all information of student along with total and Percentage. Display marklist with Use of manipulators.
  17. Write a menu driven C++ program using class to perform all arithmetic operation (+, -, \*, /) (use inline function).
  18. Write a C++ program to find volume of cube, cylinder and rectangle using function overloading
  19. Write a C++ program to find area of triangle, circle, and rectangle using function overloading.
  20. Create a class student containing data members: a. Roll\_no b. name c. marks1, marks2, marks3 Write necessary member functions: a. to accept details of all students b. to display details of one student c. to display details of all students (Use Function overloading).
  21. Write a C++ program using class to check maximum of two integer numbers using Inline function and conditional operator
  22. Create two classes 'dist1 (meters, centimeters) and dist2 (feet, inches). Accept two distances from the user, one in meters and centimeters and the other in feet and inches. Find the sum and difference of the two distances. Display the result in both (meters and centimeters) as well as feet and inches (use friend function).
  23. Write a C++ program using class which contains two data members of type integer. Create and initialize the object using default constructor, parameterized constructor and parameterized constructor with default value. Write a member function to display maximum from given two numbers for all objects
  24. Create a class time that contains hours, minute and seconds as data members. Write the member function to overload operator '+' to add two object of type time, (Use Parameterized constructor to accept values for time).
  25. Write a program in C++ to overload the binary operator '+'. Overload an operator binary '+'. Overload operator binary '+' (addition).
  26. Create class "multi" with 2 data members x and y. Class has multiple constructor and i.e. one default constructor and another parameterized, which receives value of x and y. show() display the value accordingly.
  27. WAP to print user-defined in formations using formatting functions. Input or assign strings and numbers and define field width for them. Fill the empty spaces with character and also define size of numerical float values
  28. Create base class called shape. Use this class to store two double type values that could be used to compute the area of figures. Derive two specific classes called cylinder and rectangle from the base shape. Add to the base class, a member function get\_data(), print\_data() to initialize base class data members and display\_area(), display\_perimeter() to compute and display area and perimeter of shape
  29. Implement multiple and hierarchical inheritance. The class Allrounder derives information from both Bowler and Batsman classes which in turn derive information from the class Cricketer. Define all four classes and write a program to create, update and display information contained in Allrounder objects display the object having the highest number of runs.
  30. Create a base class Conversion. Derive three different classes Weight (Gram, Kilogram), Volume (Milliliter, Liter), Currency (Rupees, Paise) from Conversion class. Write a C++ program to perform read, convert and display operations. (Use



Pure virtual function) Create a base class Media. Derive two different classes Book (Book\_id, Book\_name, Publication, Author, Book\_price) and CD (CD\_title, CD\_price,CD\_capacity) from Media. Write a C++ program to accept and display information of both Book and CD. (Use pure virtual function)

31. Create a base class Media. Derive two different classes Book (Book\_id, Book\_name, Publication, Author, Book\_price) and CD (CD\_title, CD\_price,CD\_capacity) from Media. Write a C++ program to accept and display information of both Book and CD. (Use pure virtual function)
32. Write a C++ program to read the contents of a text file. Count and display number of characters, words and lines from a file. Find the number of occurrences of a given word present in a file.

**Ahmednagar Jilha Maratha Vidya Prasarak Samaj's  
New Arts, Commerce and Science College, Ahmednagar (Autonomous)**

**Syllabus of S. Y. B.C.A. Science**

**Under**

**Faculty of Science**

<b>Semester – III</b>	<b>Paper – V</b>
<b>Course Code: BCA-SC 305 P</b>	<b>Title of the Course: Data Structures Using C (Lab)</b>
<b>Credits: 02</b>	<b>Total Lectures: 30 Hrs.</b>

### Course Outcomes (COs):

- a. To understand the concepts of ADTs
- b. To understand sorting, searching and hashing algorithms
- c. To learn linear data structures – lists, stacks, and queues
- d. To apply Tree and Graph structures

### Detailed Syllabus: Lab Course Contents

#### Assignment No1. Basic Concept and Introduction to Data Structure (06)

- 1) Write a 'C' program to evaluate a given polynomial using function. (Use array).
- 2) Write a 'C' program that create a 2-D table of integers whose size will be specified at run
  - a. time. (Dynamic Memory Allocation)
- 3) Write a 'C' program for addition of two polynomials using array.
- 4) Write a C program that create a 1-D table of integers whose size will be specified at run time. (Dynamic Memory Allocation)
- 5) Write a 'C' program to accept two polynomials and add these two polynomials using function. Display the result (Use array).
- 6) Write a 'C' program to accept the details of employees from user and display it on the screen using Dynamic Memory Allocation

#### Assignment No 2: Linear data structures (06)

- 1) Write a 'C' program to sort array elements using Bubble sort method.
- 2) Write a 'C' program for implementing Linear Search method using function.
- 3) Write a 'C' program to search given elements into the list using Non-Recursive Binary Search Method.
- 4) Write a 'C' program to search given element into the list using Recursive Binary search method.

- 5) Write a 'C' program to sort array elements using Insertion sort method.
- 6) Write a 'C' program to sort array elements in ascending order using Selection sort method.
- 7) Write a 'C' program to accept n student names from user and store it in an array. Write a function to search given student name into the array using Linear search method.
- 8) Write a 'C' program to create a random array of n integers. Sort the array using bubble sort. Accept a value x from user and use linear search algorithm to check whether the number is present in array or not.
- 9) Write a 'C' program to sort the array elements in ascending order using Merge sort a. method.
- 10) Write a 'C' program to sort the element using Quick sort (recursive) method.

**Assignment No 3: Stack and Queue****(06)**

- 1) Write a 'C' program to reverse a string using Static implementation of Stack.
- 2) Write a 'C' program to accept an infix expression, convert it into its equivalent postfix expression and display the result.

3) Write a menu driven program in 'C' for static implementation of Circular Queue for integers. The menu includes

Insert

Delete

Display

Exit

4) Write a menu driven program using 'C' for Dynamic implementation of Queue for integers. The menu includes

Insert

Delete

Display

Exit

5) Write a 'C' program to read a postfix expression, evaluate it and display the result.

(Using two variable)

6) Write a 'C' program to accept an infix expression, convert it into its equivalent prefix expression and display the result.

7) Write menu driven program using 'C' for Dynamic implementation of Stack. The menu

includes following operations:

push

pop

display

exit

- 8) Write a 'C' program which accept the string and reverse each word of the string using stack.

**Assignment No 4: Linked List**

**(06)**

- 1) Write a menu driven program using 'C' for singly linked list-

- To create linked list.
- To display linked list
- To insert node at last position of linked list.
- To delete node from specific position of linked list.
- 

- 2) Write a 'C' program to create linked list with given number in which data part of each node contains individual digit of the number.

(Ex. Suppose the number is 584 then the nodes of linked list should contain 5, 8, 4)

- 3) Write a 'C' program to create a singly linked list and count total number of nodes in it and display the result.

- 4) Write a menu driven program using 'C' for singly linked list -

- To create linked list.
- To display linked list
- To search node in linked list.
- Insert at last position

- 5) Write a 'C' program to create two singly linked lists and perform the union of two lists and display it.

- 6) Write a 'C' program to create a singly linked list, reverse it and display both the list.

- 7) Write a 'C' program to create two singly linked lists and perform the intersection operations on two lists and display the resultant list.

- 8) Write a 'C' program to create Circular Singly Link list and display it.

- 9) Write a menu driven program using 'C' for implementation of Singly linked list. Menu includes –

- Create.
- Display
- Insert node at specific position
- Search a given element in list

- 10) Write a menu driven program using 'C' for implementation of Singly linked list. Menu includes –

- Create
- Display
- Insert node at first position

- 11) Write a 'C' program to sort elements of a singly linked list in ascending order and display the sorted List.
- 12) Write a menu driven program using 'C' for implementation of Singly linked list. Menu includes –
- Create
  - Display
  - Insert Node at particular position
- 13) Write a 'C' program to read n integers and create two lists such that all positive numbers are in one list and negative numbers are in another list. Display both the lists.
- 14) Write a 'C' program to create two singly linked lists and concatenate one list at the end of another list.
- 15) Write a 'C' program to remove last node of the singly linked list and insert it at the beginning of list.
- 16) Write a 'C' program to count all non-zero elements, odd numbers and even numbers in the singly linked list.
- 17) Write menu driven program using 'C' for Circular doubly linked list. The menu includes
- Create
  - Display
  - Exit
- 18) Write a 'C' program to create doubly link list and display nodes having even value.
- 19) Write a 'C' program to create a Circular doubly Link list and display it.
- 20) Write a 'C' program to swap  $m^{\text{th}}$  and  $n^{\text{th}}$  element of singly linked list.
- 21) Write a 'C' program to create a singly Link list and display its alternative nodes
- 22) Write a 'C' program to create to a Singly linked list. Accept the number from user, search the number in the list .If the number is present delete the node from the list and display the list .If node not present print the message "Node not Found".
- 23) Write menu driven program using 'C' for Singly linked list. The menu includes
- Create
  - Display
  - Search particular node in list and display its position.

**Assignment No 5: Tree and Graph****(06)**

- 1) Write menu driven program using 'C' for Binary Search Tree. The menu includes
- Create a BST
  - Insert element in a BST

- Display
- 2) Write a 'C' program for Binary Search Tree. The menu includes
  - Create a BST
  - Display
  - Delete a given element from BST
- 3) Write a 'C' program to create a Binary tree, traverse it using recursive operations like inorder, preorder and postorder and display the result of each one separately.
- 4) Write a 'C' program to read 'n' integers and store them in a binary tree structure and count the following and display it.
  - Number of nodes
  - Degree of tree
  - Leaf nodes
- 5) Write a 'C' program to create binary search tree and display its leaf nodes.
- 6) Write menu driven program using 'C' for Binary Search Tree. The menu includes -
  - Create a BST
  - Traverse it by using Inorder traversing technique
  - Search a given value in BST
- 7) Write a 'C' program to read 'n' integers and store them in a Binary search tree and display the nodes level wise.
- 8) Write menu driven program using 'C' for Binary Search Tree. The menu includes
  - Create a BST
  - Traverse it by using Preorder traversing technique
  - Search a given value in BST
- 9) Write a 'C' program to read 'n' integers and store them in binary search tree. Display mirror image of tree. (using recursive function)
- 10) Write a 'C' program to count indegree and outdegree of each node in graph.
- 11) Write a 'C' program to read the adjacency matrix of directed graph and convert it into adjacency list.
- 12) Write a 'C' program to read an adjacency matrix of a directed graph and traverse it using DFS.
- 13) Write a 'C' program to read an adjacency matrix of a directed graph and traverse using BFS.

**Ahmednagar Jilha Maratha Vidya Prasarak Samaj's**  
**New Arts, Commerce and Science College, Ahmednagar (Autonomous)**

**Syllabus of S. Y. B.C.A. Science**

**Under**

**Faculty of Science**

<b>Semester – III</b>	<b>Paper – VI</b>
<b>Course Code: BCA-SC 306 P</b>	<b>Title of the Course: Advance Database Management Systems (Lab)</b>
<b>Credits: 02</b>	<b>Total Lectures: 30 Hrs.</b>

### **Course Outcomes (COs):**

- a. Formulate SQL queries using advanced features
- b. Write stored procedures, cursors and triggers using PL/PostgreSQL.
- c. Design a database using database normalization technique

### **Detailed Syllabus: Lab Course Contents**

**Assignment 1:** Designing a Database using normalization theory for the given application/database design

**Assignment 2:** Simple and Nested Queries

**Assignment 3:** Views Creation

**Assignment 4:** Stored Functions

- 1) A Simple Stored Function
- 2) A Stored Function that returns
- 3) A Stored Function recursive

**Assignment 5:** Cursors

- 1) Simple Cursor
- 2) Parameterize Cursor

**Assignment 6:** Error and Exception handling

- 1) Simple Exception- Raise Debug Level Messages
- 2) Simple Exception- Raise Notice Level Messages
- 3) Simple Exception- Raise Exception Level Messages

**Assignment 7: Triggers**

- 1) Before Triggers (insert, update, delete)
- 2) After Triggers (insert, update, delete)



**Ahmednagar Jilha Maratha Vidya Prasarak Samaj's  
New Arts, Commerce and Science College, Ahmednagar (Autonomous)**

**Syllabus of S. Y. B.C.A. Science**

**Under**

**Faculty of Science**

<b>Semester – III</b>	<b>Paper – VII</b>
<b>Course Code: BCA-SC 307 T</b>	<b>Title of the Course: React JS</b>
<b>Credits: 02 Credits</b>	<b>Total Lectures: 30 Hrs.</b>

### **Course Outcomes (COs):**

- a. To build user interfaces specifically for single-page applications.
- b. To enable fast and quality assured application development that in turn saves time for both - clients and developers.
- c. To create reusable UI components and hence reduce the code.

### **Detailed Syllabus:**

#### **Unit I: Introduction to ReactJS (04)**

- 1.1 What is ReactJS?
- 1.2 History of ReactJS.
- 1.3 Why to learn ReactJS?
- 1.4 Features of ReactJS.
- 1.5 Advantages of ReactJS.
- 1.6 Aspects of ReactJS.

#### **Unit II: ReactJS Installation (06)**

- 2.1 Setting up Environment for ReactJS
- 2.2 NodeJS Installation
- 2.3 Downloading and adding ReactJS Libraries.
- 2.4 Creating your first ReactJS project.
- 2.5 Use of create-react-app
- 2.6 ReactJS Project Structure

#### **Unit III: ReactJS ES6 (06)**

- 3.1 Classes
- 3.2 Arrow Functions
- 3.3 Variables
- 3.4 Array Methods
- 3.5 DeStructuring
- 3.6 Modules
- 3.7 Ternary Operators and Spread Operators.

**Unit IV: React JS Fundamentals****(08)**

- 4.1 React JSX
- 4.2 React Component
  - 4.2.1 Function Based Component
  - 4.2.2 Class Based Component
- 4.3 React Component API
- 4.4 Component Lifecycle
- 4.5 React State
- 4.6 React Props
- 4.7 React State vs React Props

**Unit V: Form Handling in ReactJS****(06)**

- 5.1 React Forms
- 5.2 React Events
- 5.3 React Lists
- 5.4 React Keys
- 5.5 React Refs
- 5.6 React Fragments

**Suggested Readings:**

1. “ FullStack React” The Complete Guide to ReactJS and Friends
2. The Road to Learn React: Your journey to master plain yet pragmatic React.js by Robin Wieruch
3. React in Action by Mark Tielens Thomas
4. Learning React, 2nd Edition

## Ahmednagar Jilha Maratha Vidya Prasarak Samaj's

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

## Syllabus of S. Y. B.C.A. Science

Under

Faculty of Science

<b>Semester – III</b>	<b>Paper – VIII</b>
<b>Course Code: BCA-SC 308 P</b>	<b>Title of the Course: React JS (Lab)</b>
<b>Credits: 02 Credits</b>	<b>Total Lectures: 30 Hrs.</b>

**Course Outcomes (COs):**

- To build user interfaces specifically for single-page applications.
- To enable fast and quality assured application development that in turn saves time for both - clients and developers.
- To create reusable UI components and hence reduce the code.

**List of Assignments:**

- Setup the environment for ReactJS.
  - Install NodeJS.
  - Download ReactJS Libraries
  - Create ReactJS App using create-react-app
- Create a simple ReactJS app and display "Hello World" message on webpage.
- Create a list of programming languages you know and display the list on webpage.
- Create a class-based Component Car and display the car details.
- Create a function-based Component and display your own details like Name, DOB, Contact Number, Address etc.
- Add an attribute to pass a colour to the Car Component and use it render() function to display the Colour of Car.
- Create a class Component Car and display Car Component in root.
- Create a Constructor function in the Car Component and add a Colour property. Use the Colour property in render function to display it.
- Create a class Component called as Garage. Use the Car Component inside the Garage Component.
- Specify the state object in the constructor method. The state object can contain as many properties as you like. Add all properties of your Car Component like brand, model, colour, year, etc.
- Refer to the state object and display Car details using render() function.
- Add a button with an onClick() event that will change the colour property.
- Implement lifecycle methods of Component.
- Add a Brand attribute to the Car element. Use the props object to access brand attribute in Component.
- Send this Brand property from Garage Component to Car Component.

16. Create a Football Component. Put a shoot() function in it. Call this shoot() on a button click.
17. Create a list of Car brands in Garage Component. Render all the Car from Garage.
18. Add keys to above list items of Brands in Garage Component. Render all the Car from Garage.
19. Add an attribute that allows React to keep track of elements in list.
20. Create a form that allow suer to enter their name. Use onChange() to manage the input.
21. Add a submit button and an event handler in the onSubmit attribute.

**Ahmednagar Jilha Maratha Vidya Prasarak Samaj's  
New Arts, Commerce and Science College, Ahmednagar (Autonomous)**

**Syllabus of S. Y. B.C.A. Science**

**Under**

**Faculty of Science**

<b>Semester – III</b>	<b>Paper – IX</b>
<b>Course Code: 309 T</b>	<b>Title of the Course: Technical English (MIL)</b>
<b>Credits: 02 Credits</b>	<b>Total Lectures: 30 Hrs.</b>

**Course Outcomes (COs):**

- a. Acquaint and enlighten with the speaking skill in various contexts.
- b. Acquaint and familiarize with advanced writing skills in different contexts.
- c. Acquaint and familiarize with soft skills through listening and speaking practice.
- d. Minimize the gap between the existing communicative skills and acquire the skills they require at professional level.
- e. Acquire the use of grammar effectively (vocabulary and so on) through extensive coursework on writing reports and reading comprehensions, articles, essays, general discussion etc.

**Detailed Syllabus:**

**UNIT: I**

**(10)**

Grammar: Sentence and Sentence Construction

Vocabulary: Homophones, Homographs, Homonyms

Listening: Listening for gist and detailed meaning and to identify the attitudes and opinions of the speakers.

Speaking: Mini-presentations on a business theme and giving information and expressing opinions.

Reading: Reading for detailed comprehension of detailed material; Skimming and Scanning.

Writing: Writing to deal with requests, giving information about a product.

**UNIT: II**

**(10)**

Grammar: Concord, Modal Auxiliary, Question Tags.

Vocabulary: Business Vocabulary.

Listening: Answering multiple choice questions on short conversations or monologues.

Speaking: Expressing opinions, Agreeing and Disagreeing, Talking about oneself, one's

current situations and plans.

Reading: Reading for Understanding Vocabulary and grammar in a short text

Writing: Writing for functional/ communicative task- e.g., Re-arranging appointments, asking for permission, giving instructions, apologizing and offering compensation.

### **UNIT:III INTERVIEW TECHNIQUES**

**(10)**

1. Job Application Letter
2. Resume Writing
3. . GDPI
4. Presentations

\*Practice and Discussion Sessions

#### **Suggested Readings:**

1. Whitby, N., Business Benchmark. Cambridge English,2013.
2. Hughes, J. and Newton, J., Business results – Intermediate,2021
3. Frank, M. Writing as Thinking: A Guided Process Approach. Prentice Hall Reagents.
4. Hamp-Lyons, L. and B. Heasley, Study Writing; A Course in Written English for Academic and Professional Purposes. Cambridge UP.
5. Quirk, R. S., Greenbaum, G. Leech and J. Svartik, A Comprehensive Grammar of the English Language. Longman.
6. Riordan, Daniel G. and Steven A., Panley. Technical Report Writing Today. Biztaantra.
7. Gerson, S., Gerson, S., Technical Writing: Process and Product. Pearson, 2011.
8. Board of Editors, Horizons: English in Multivalent Context, Orient Black Swan.

#### **WEB REFERENCES:**

1. [www.cambridgeenglish.org](http://www.cambridgeenglish.org)

# SEMESTER IV

**Ahmednagar Jilha Maratha Vidya Prasarak Samaj's**  
**New Arts, Commerce and Science College, Ahmednagar (Autonomous)**

**Syllabus of S. Y. B.C.A. Science**

**Under**

**Faculty of Science**

<b>Semester – IV</b>	<b>Paper – I</b>
<b>Course Code: BCA-SC 401 T</b>	<b>Title of the Course: Core Java Programming</b>
<b>Credits: 04</b>	<b>Total Lectures: 60 Hrs.</b>

**Course Outcomes (COs):**

- a. Identify classes, objects, class members and relationships for a given problem.
- b. Design end to end applications using object oriented constructs.
- c. Apply collection classes for storing java objects.
- d. Use Java APIs for program development.
- e. Handle abnormal termination of a program using exception handling.

**Detailed Syllabus:**

**Unit I: Introducing Java (10)**

- 1.1 A Short History of Java
- 1.2 Features of Java
- 1.3 Java Environment – Compiler, Interpreter, JVM
- 1.4 Simple Java Program
- 1.5 Types of Comments
- 1.6 Declaring Single and Multi-Dimensional Arrays
- 1.7 Accepting Input Using Command Line Arguments
- 1.8 Accepting Input from Console (Using BufferedReader and Scanner Class)

**Unit II Classes and Objects (10)**

- 2.1 Defining Your Own Classes
- 2.2 Access Specifiers (Public, Protected, Private, Default)
- 2.3 Array of Objects
- 2.4 Constructor, Overloading Constructors and Use of “This” Keyword



- 
- 2.5 Static Blocks, Static Fields, Static Methods, Static Class
  - 2.6 Predefined Classes – Object Class Methods (equals(), toString(), hashCode())
  - 2.7 String class(basic methods) , String Buffer class , StringBuilder class
  - 2.8 Inner classes : static inner class , anonymous inner class.
  - 2.8 Garbage Collection (finalize() Method)

**Unit III :Inheritance (8)**

- 3.1 Inheritance Basics (Extends Keyword) and Types of Inheritance
- 3.2 Superclass, Subclass and Use of Super Keyword
- 3.3 Method Overriding and method overloading
- 3.4 Use of Final Keyword Related To Variable, Method and Class
- 3.5 Use of Abstract Class and Abstract Methods

**Unit IV:Interface and Packages (7)**

- 4.1 Defining and Implementing Interfaces
- 4.2 Runtime polymorphism using interface
- Packages
- 4.3 Creating, Accessing and using Packages

**Unit IV : Collections (10)**

- 5.1 Wrapper Classes
- 5.2 Introduction to the Collection framework
- 5.3 List – ArrayList, LinkedList and Vector
- 5.4 Set - HashSet, TreeSet, and LinkedHashSet
- 5.5 Map – HashMap, LinkedHashMap, Hashtable and TreeMap
- 5.6 Queue – PriorityQueue
- 5.7 Interfaces such as Iterators, ListIterators, Enumerations

**Unit V:Exception Handling and I/O Handling (05)**

- 5.1 Exception class, Checked and Unchecked exception
- 5.2 Catching exception and exception handling – try, catch, finally, throw and throws, multiple catch block

5.3 Creating user defined exception

I/O

5.4 Introduction to Java IO package.

5.5 Streams: Byte Stream and Character Stream

5.6 File class

5.7 File IO Basics: FileReader and FileWriter class

5.8 Object Serialization and Deserialization

## **Unit VI Swing**

**(10)**

6.1 What is Swing?

6.2 The MVC Architecture and Swing

6.3 Container in Swing : JFrame, JPanel

6.4 JComponent class– JLabel, JButton, JText, JTextArea, JCheckBox, JRadioButton, JList, JComboBox, JMenu and JPopupMenu Class, JMenuItem

6.5 Layout Manager – FlowLayout,LinearLayout,TabLayout,GridLayout,BorderLayout

6.6 Dialogs (Message, confirmation, input), JFileChooser

6.7 Event Handling: Event sources, Listeners – ActionListener, ItemListener

6.8 Mouse and Keyboard Event Handling

6.9 Adapters – MouseAdapter, KeyAdapter

### **Suggested Readings:**

1) Core Java Volume I - Fundamentals By Cay S. Horstmann, 11th Edition, Prentice Hall, ISBN 978-0-13-516630-7

2) The Complete Reference By Herbert Schildt, 11th Edition, McGraw Hill Education, ISBN 978-260-44023-2

3) Java Beginners Guide By Herbert Schildt, 8 th Edition, McGraw-Hill Education ISBN 978-1- 260-44021-8

4) Core Java Volume II – Fundamentals By Cay S. Horstmann, 11th Edition, Prentice Hall, ISBN 978-013-516631-4

5) Java 2 Programming Black Book By Steven Holzner, DreamTech Press, ISBN 978-93-5119-953-4

**Ahmednagar Jilha Maratha Vidya Prasarak Samaj's  
New Arts, Commerce and Science College, Ahmednagar (Autonomous)**

**Syllabus of S. Y. B.C.A. Science**

**Under**

**Faculty of Science**

<b>Semester – IV</b>	<b>Paper – II</b>
<b>Course Code: BCA-SC 402 T</b>	<b>Title of the Course: Python Programming</b>
<b>Credits: 04</b>	<b>Total Lectures: 60 Hrs.</b>

**Course Outcomes (COs):**

- a. To introduce various concepts of programming to the students using Python.
- b. Students should be able to apply the problem solving skills using Python.
- c. To acquire object oriented skill in python

**Detailed Syllabus:**

**Unit I: Introduction to Python Scripting (04)**

- 1.1. Why Scripting is Useful in Computational Science
- 1.2. Classification of Programming Languages
- 1.3. Productive Pairs of Programming Languages
- 1.4. Gluing Existing Applications
- 1.5. Scripting Yields Shorter Code, Efficiency
- 1.6. Type-Specification (Declaration) of Variables
- 1.7. Flexible Function Interfaces
- 1.8. Interactive Computing
- 1.9. Creating Code at Run Time
- 1.10. Nested Heterogeneous Data Structures
- 1.11. GUI Programming
- 1.12. Mixed Language Programming
- 1.13. When to Choose a Dynamically Typed Language
- 1.14. Why Python? Script or Program?
- 1.15. Application of Python
- 1.16. Concept (immutable)

**Unit II: Basic Python (06)**

- 2.1 Python identifiers and reserved words
- 2.2 Lines and indentation, multi-line statements
- 2.3 Comments Input/output with print and input functions
- 2.4 Command line arguments and processing command line arguments
- 2.5 Standard data types - basic, none, Boolean (true & False), numbers

2.6 Data type conversion

2.7 Python basic operators (Arithmetic, comparison, assignment, bitwise logical)

2.8 Python membership operators (in & not in)

2.9 Python identity operators (is & is not)

2.10 Operator precedence

2.11 Control Statements, Python loops, Iterating by subsequence index, loop control statements (break, continue, pass)

2.12 Mathematical functions and constants (import math), Random number functions

### **Unit III: Python strings**

**(06)**

3.1 Concept, escape characters

3.2 String special operations

3.3 String formatting operator - Single quotes, Double quotes, Triple quotes

3.4 Raw String, Unicode strings, Built-in String methods.

3.5 Python Lists - concept, creating and accessing elements, updating & deleting lists, reversing a list

3.7 Indexing, slicing and Matrices

3.8 built-in List functions

3.9 Functional programming tools - filter(), map(), and reduce()

3.10 Using Lists as stacks and Queues, List comprehensions

### **Unit IV: Python tuples and sets**

**(06)**

4.1 Creating & deleting tuples

4.2 Accessing values in a tuple

4.3 Updating tuples, delete tuple elements

4.4 Indexing, slicing and Matrices, built- in tuple functions.

4.6 Sets - Concept, operations.

### **Unit V: Python Dictionary**

**(04)**

5.1 Concept (mutable)

5.2 Creating and accessing values in a dictionary

5.3 Updating dictionary, delete dictionary elements

5.4 Properties of dictionary keys

5.5 Built-in dictionary functions.

## **Unit VI: Functions**

**(10)**

6.1 Defining a function (def)

6.2 Calling a function

6.3 Function arguments - Pass by value, Keyword Arguments, default arguments

6.4 Scope of variable - basic rules

6.5 Documentation Strings

6.6 Variable Number of Arguments

6.7 Call by Reference

6.8 Order of arguments (positional, extra & keyword)

6.9 Anonymous functions

6.10 Recursion

6.11 Treatment of Input and Output Arguments

6.12 Unpacking argument lists

6.13 Lambda forms

6.14 Function Objects

6.15 Function Ducktyping & polymorphism

6.16 Generators (functions and expressions) and iterators, list comprehension

## **Unit VII: Files and Directories**

**(08)**

7.1 Creating files

7.2 Operations on files (open, close, read, write)

7.3 File object attributes, file positions, Listing Files in a Directory Testing File Types

7.4 Removing Files and Directories

7.5 Copying and Renaming Files

7.6 Splitting Pathnames

7.7 Creating and Moving to Directories

7.8 Traversing Directory Trees

7.9 Illustrative programs: word count, copy file

### **Unit VIII: Python Classes / Objects**

**(12)**

8.1 Object oriented programming and classes in Python - creating classes, instance objects, accessing members

8.2 Data hiding (the double underscore prefix)

8.3 Built-in class attributes

8.4 Garbage collection : the constructor

8.5 Overloading methods and operators

8.6 Inheritance - implementing a subclass, overriding methods

8.7 Recursive calls to methods

8.8 Class variables, class methods, and static methods

### **Unit IX: Python Exceptions**

**(04)**

9.1 Exception handling : assert statement

9.2 Except clause - with no exceptions and multiple exceptions

9.3 Try - finally, raising exceptions, user-defined exceptions.

### **Suggested Readings:**

1. Introducing Python- Modern Computing in Simple Packages – Bill Lubanovic, O,,Reilly Publication

2. Magnus Lie Hetland, Beginning Python: From Novice to Professional, Apress

3. Paul Gries, et al., Practical Programming: An Introduction to Computer Science Using Python 3, Pragmatic Bookshelf, 2/E 2014

4. Introduction to Computer Science Using Python- Charles Dierbach, Wiley Publication Learning with Python “, Green Tea Press, 2002

5. E-Books : python\_tutorial. pdf, python\_book\_01.pdf

6. Beginning Programming with Python for Dummies Paperback – 2015 by John Paul Mueller

7. A Beginner’s Python Tutorial: [http://en.wikibooks.org/wiki/A\\_Beginner%27s\\_Python\\_Tutorial](http://en.wikibooks.org/wiki/A_Beginner%27s_Python_Tutorial)

**Ahmednagar Jilha Maratha Vidya Prasarak Samaj's  
New Arts, Commerce and Science College, Ahmednagar (Autonomous)**

**Syllabus of S. Y. B.C.A. Science**

**Under**

**Faculty of Science**

<b>Semester – IV</b>	<b>Paper – III</b>
<b>Course Code: BCA-SC 403 T</b>	<b>Title of the Course: Operating System Concepts</b>
<b>Credits: 04</b>	<b>Total Lectures: 60 Hrs.</b>

### **Course Outcomes (COs):**

- a. Describe algorithms for process, memory and disk scheduling
- b. Apply technique for inter-process communication and Multithreading.
- c. Implement concept of critical-section
- d. Compare and contrast deadlock avoidance and prevention.
- e. Use functions for file system management

### **Detailed Syllabus:**

#### **Unit I: Introduction to Operating Systems (05)**

- 1.1 Operating Systems Overview
  - 1.1.1 Definition
  - 1.1.2 Role and objective of operating systems
  - 1.1.3 Operating system as User View and System View
  - 1.1.4 Functions of Operating system
- 1.2 Operating system structure
  - 1.2.1 Operating system Services
  - 1.2.2 System call and types
  - 1.2.3 System programs
  - 1.2.4 System boot

#### **Unit II : Process Management (12)**

- 2.1 Process Concept
  - 2.1.1 Lifecycle of process
  - 2.1.2 Process Control Block
- 2.2 Operations on Process – Process creation using fork(), Process termination

## 2.3 CPU Scheduling

2.3.1 Basic scheduling – CPU-I/O burst cycle, CPU Scheduler, Pre-emptive Scheduling and Dispatcher

2.3.2 Process Scheduling – Scheduling Criteria, Scheduling queues, Schedulers, context switch

2.3.3 Scheduling Algorithms – FCFS, SJF, Priority scheduling, Round-robin scheduling, multiple queue scheduling, multilevel feedback queue scheduling

2.4 Inter-process Communication – Shared memory system, Message passing systems

## 2.5 Multithreaded Programming

2.5.1 Overview, types of thread

2.5.2 Multithreading Models

2.5.3 Thread Scheduling

## **Unit III: Process Synchronization**

**(06)**

2.1 Background

2.2 Critical Section Problem

2.3 Semaphores: Usage, Implementation

2.4 Classic Problems of Synchronization – The bounded buffer problem, the reader writer problem, the dining philosopher problem

## **Unit III: Deadlocks**

**(12)**

3.1 System Model

3.2 Deadlock Characterization – Necessary Conditions, Resource Allocation Graph

3.3 Deadlock Prevention

3.4 Deadlock Avoidance - Safe state, Resource-Allocation-Graph Algorithm, Banker's Algorithm

3.5 Deadlock Detection

3.6 Recovery from Deadlock – Process Termination, Resource Preemption

## **Unit IV: Memory Management**

**(12)**

4.1 Background – Basic Hardware, Address Binding, Logical Versus Physical Address Space, Dynamic Loading, Dynamic Linking and Shared Libraries, Overlays



4.2 Swapping

4.3 Contiguous Memory Allocation – Memory mapping and protection, Memory allocation, Fragmentation

4.4 Paging – Basic Method, Hardware support, Protection, Shared Pages

4.5 Segmentation – Basic concept, Hardware

4.6 Virtual Memory Management – Demand paging, Performance of demand paging, Page replacement – FIFO, Optimal, LRU and Second Chance Algorithm

4.7 Thrashing – Cause of thrashing, Working-Set Model

### **Unit V: File System**

**(08)**

5.1 File Concept, File Attribute, File Operations, File Types, File Structure

5.2 Access Methods - Sequential Access Method, Direct Access Method, Other Access Methods

5.3 Directory overview, Single level directory, Two level directory, Tree structure directory, Acyclic graph directory, General graph directory

5.4 File System Structure and Implementation - Partitions and Mounting, Virtual File Systems

5.5 Allocation Methods - Contiguous allocation, Linked allocation, Indexed allocation

5.6 Free Space Management – Bit vector, Linked list, Grouping, Counting, Space maps

### **Unit VI: Disk Scheduling**

**(05)**

6.1 Overview

6.2 Disk Structure

6.3 Disk Scheduling, FCFS Scheduling, SSTF Scheduling, SCAN Scheduling, C-SCAN Scheduling, LOOK Scheduling

6.4 Disk Management

### **Suggested Readings:**

1. Operating System Concepts, Avi Silberschatz, Peter Galvin, Greg Gagne, 8th Edition, Wiley Asia
2. Operating Systems: Internals and Design Principles, William Stallings, Prentice Hall of India.
3. The Design of the UNIX Operating System By Maurice J. Bach., PHI publication

## Ahmednagar Jilha Maratha Vidya Prasarak Samaj's

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

## Syllabus of S. Y. B.C.A. Science

Under

Faculty of Science

<b>Semester – IV</b>	<b>Paper – IV</b>
<b>Course Code: BCA-SC 404 P</b>	<b>Title of the Course: Core Java Programming (Lab)</b>
<b>Credits: 02</b>	<b>Total Lectures:30 Hrs.</b>

**Course Outcomes:**

- Identify classes, objects, class members and relationships for a given problem.
- Design end to end applications using object oriented constructs.
- Apply collection classes for storing java objects.
- Use Java APIs for program development.
- Handle abnormal termination of a program using exception handling.

**Lab Course Contents****Assignment No 1. Basics of Java, Classes and Objects****Classes and objects, Array of objects, Static keyword,Constructor**

- Write a Java program to accept a number from command prompt and generate multiplication table of a number.
- Write a Java program to print the factors of a number.
- Write a Java program to display Fibonacci series using recursion.
- Write a Java program to accept a number from user and print all prime numbers upto that number(Use Buffered Reader class)
- Write a Java program to print the sum of elements of the array.
- Write a Java Program which define class Product with data member as id, name and price Store the information of 5 products and Display the name of product having minimum price(Use array of object).
- Write a Java Program which define class Employee with data member as id, name and salary Store the information of n employees and Display the name of employee having maximum salary (Use array of object).
- Define a class student having rollno, name and percentage. Define Default and parameterized constructor. Overload the constructor. Accept the 5 student details and display it.(usethis keyword).
- Define a class MyNumber having one private integer data member. Write a default constructor initialize it to 0 and another constructor to initialize it to a value. Write methods isNegative, isPositive, isOdd, iseven. Use command line argument to pass a value

to the object and perform the above tests.

10. Define a class CricketPlayer(name, no\_of\_innings, no\_of\_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 (). Handle appropriate exception while calculating average. Define static method “sortPlayer” which sort the array on the basis of average. Display the player details in sorted order.

### Assignment No 2: Inheritance

(04)

#### Types of Inheritance, Method Overriding, Super keyword

1. Write a Java program to create a super class Vehicle having members Company and Price. Derive two different classes LightMotorVehicle (mileage) and HeavyMotorVehicle (capacity\_in\_tons). Accept the information for “n” vehicles and display the information in appropriate form. While taking data, ask user about the type of vehicle first.
2. Define a class Employee having members – id, name, department, salary. Define default and parameterized constructors. Create a subclass called Worker with private member bonus. Define methods accept and display in both the classes. Create “n” objects of the Worker class and display the details of the worker having the maximum total salary (salary + bonus).
3. 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.

### Assignment No 3: Interfaces and Packages

(8)

1. Create an interface “CreditCardInterface” with methods: viewCreditAmount(), useCard(), payCard(), and increaseLimit(). Create a class “SilverCardCustomer” (name, cardnumber (16digit), creditamount-initialized to 0, creditLimit-set to 50,000) which implements above interface. Inherit class GoldCardCustomer from SilverCardCustomer having same methods but creditLimit of 1,00,000. Create an object of each class and perform operations. Display appropriate messages for success or failure of transaction. (Use method overloading)
  - a. useCard() method increase the creditAmount by a specific amount upto creditLimit.
  - b. payCredit() reduces the creditAmount by a specific amount.
  - c. increaseLimit() increases the creditLimit for GoldCardCustomers (only 3 times, not more than 5000 rupees each time.)
2. Create an abstract class Shape with methods area & volume. Derive two classes Sphere (radius), Cylinder(radius, height) from it. Calculate area and volume of both. (Use Method Overriding)
3. Define an Interface Shape with abstract method area(). Write a java program to calculate an area of Circle and Sphere. (use final keyword)
4. Create a package named Series having three different classes to print series:
  - a. Fibonacci series
  - b. Cube of numbers

- c. Square of numbers
- Write a java program to generate „n“ terms of the above series.
  - Write a package game which will have 2 classes Indoor & Outdoor. Use a function display() to generate the list of player for the specific game. Use default & parameterized constructor.

**Assignment No 3: Collections****(04)****List, Set, Map interface related classes,**

- Accept N integers from the user and store them in a collection. Display them in the sorted order. The collection should not accept duplicate elements. (Use a suitable collection).
- Construct a Linked List having names of Fruits: Apple, Banana, Guava and Orange and Display
  - the contents of the List using an Iterator;
  - the contents of the List in reverse order using a ListIterator;
- Create a Hash table containing Student Name and Percentage. Display the contents of the hash table. Also search for a specific Student and display his percentage.

**Assignment No 4: Exception and I/O handling****(04)****Exception handling-try and catch, User defined Exception, File class**

- Write a java program to accept a number from the user, if number is zero then throw user defined exception —Number is 0, otherwise check whether no is prime or not
- Write a java program to accept Doctor Name from the user and check whether it is valid or not. (It should not contain digits and special symbol ) If it is not valid then throw user defined Exception —Name is Invalid -- otherwise display it.
- Write a class Driver with attributes license\_no, name, address and age. Initialize values through the parameterized constructor. If age of Driver is less than 18 then user-defined exception should be generated —Age is below 18 years -.
- Write a java program that displays the number of characters, lines and words of a file. Write a java program to accept details of n customers (c\_id, cname, address, mobile\_no) from user and store it in a file ( Use DataOutputStream class). Display the details of customers by reading it from file.( use DataInputStream class)

**Assignment No 5: Swing****(04)****GUI designing, Event handling on GUI**

- Write a java program to design the following GUI using Swing components.



- Write a java program to design a following GUI using Swing. After submission, display the accepted details on the next page. (Use Event Handling)



3. Write a java program to design the following GUI using Swing components.

Submit

4. Write a program to create following GUI using Swing and check whether username and password is correct or not. (Use Event Handling)

Submit

**Ahmednagar Jilha Maratha Vidya Prasarak Samaj's**  
**New Arts, Commerce and Science College, Ahmednagar (Autonomous)**

**Syllabus of S. Y. B.C.A. Science**

under

**Faculty of Science**

<b>Semester – IV</b>	<b>Paper – V</b>
<b>Course Code: BCA-SC 405 P</b>	<b>Title of the Course: Python Programming (Lab)</b>
<b>Credits: 02</b>	<b>Total Lectures: 30 Hrs.</b>

**Course Outcomes (COs):**

- a. Write programs using Python programming constructs.
- b. Write programs using function concepts in python
- c. Develop applications using Python programming.

**Detailed Syllabus: Lab Course Contents**

- 1) Write a Python Program to Calculate the Average of Numbers in a Given List.
- 2) Write a program which accepts 6 integer values and prints “DUPLICATES” if any of the values entered are duplicates otherwise it prints “ALL UNIQUE”.

Example: Let 5 integers are (32, 10, 45, 90, 45, 6) then output “DUPLICATES” to be printed.

- 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 finds sum of digits of a number.
- 5) Write a program which prints Fibonacci series of a number.
- 6) 1. Write a program to replace all occurrences of ‘a’ with \$ in a String. (Ex. apple then output is \$pple).
- 7) Write a Python program 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}
- 8) Write a Python program to get a string made of the first 2 and the last 2 chars
- 9) 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'

Expected Result : Empty String

- 10) Write a Python program to calculate the Length of a String without using a Library Function.
- 11) 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.
- 12) Write a Python program to create a tuple with numbers and print one item.
- 13) Write a Python program to unpack a tuple in several variables.
- 14) Write a Python program to add an item in a tuple.
- 15) Copy element 44 and 55 from the following tuple into a new tuple
- 16) tuple1 = (11, 22, 33, 44, 55, 66)
- 17) Write a Python program to accept the strings which contains all vowels .
- 18) Write a Python program to create a union of sets.
- 19) Write a Python program to create an intersection of sets.
- 20) Write a Python program to find maximum and the minimum value in a set.
- 21) Write a Python program to create set difference and a symmetric difference
- 22) Write a Python program to find the length of a set.
- 23) Write a Python script to generate and print a dictionary that contains a number (Between 1 and n) in the form (x, x\*x).
- 24) Write a Python program to combine two dictionary adding values for common keys.
- 25) Sample Dictionary:
- 26) d1={'a':100,'b':200,'c':300}  
d2={'a':300,'b':200,'d':400}  
Sample output: Counter({'a': 400, 'b': 400, 'd': 400, 'c': 300})
- 27) Write a Python program to create a dictionary from a string.
- 28) Sample-String: 'W3resource'  
Expected output: {'3': 1, 's': 1, 'r': 2, 'u': 1, 'w': 1, 'c': 1, 'e': 2, 'o': 1}
- 29) Write a Python program to iterate over dictionaries using for loops.
- 30) Write a Python program to sum all the items in a dictionary.
- 31) Write a Python function that takes a number as a parameter and check the number is prime or not.
- 32) Write a generator function that reverses a given string.
- 33) Write a recursive function to calculate the sum of numbers from 0 to 10.
- 34) Write a Python program to filter a list of integers using Lambda
- 35) Write a Python program to accept and display n student's details such as roll no, name, marks in three subjects, using class. Display percentage of each student. Also, delete the object.
- 36) Write a Python program to create a class in which one method accepts a string from user and another method prints it. Define a class named Country, which has a method called printNationality. Define subclass named state from country, which has a method called printState. Write a method to print state, country and nationality.

- 37) Write a Python program to read the contents of a file in reverse order and display the size of file.
- 38) Write a Python program to 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 Shapes area is 0 by default.
- 39) Define a class Date (Day,month,year)with functions to accepts and display it. Accept date from user through user defined exception (Invalid date exception) if the date is invalid.
- 40) Write a text file named test.txt that contains integers, characters and float numbers. WAP to read the test.txt and print appropriate message using exception to print whether line contains integer, char or float value.



**Ahmednagar Jilha Maratha Vidya Prasarak Samaj's****New Arts, Commerce and Science College, Ahmednagar (Autonomous)****Syllabus of S. Y. B.C.A. Science****Under****Faculty of Science**

<b>Semester – IV</b>	<b>Paper – VI</b>
<b>Course Code: BCA-SC 406 P</b>	<b>Title of the Course: Operating System Concept (Lab)</b>
<b>Credits: 02</b>	<b>Total Lectures: 30 Hrs.</b>

**Course Outcomes (COs):**

- Implement algorithms for Process scheduling and Memory management
- Describe process synchronization and multithreading
- Compare and contrast the algorithms for memory management and its allocation policies.

**Detailed Syllabus: Lab Course Contents****Assignment 1: Operations on processes**

- Create a child process using fork (), display parent and child process id. Child process will display the message “Hello World” and the parent process should display “Hi”.
- Creating a child process using the command exec (). Note down process ids of the parent and the child processes, check whether the control is given back to the parent after the child process terminates. Write a similar program using execv() and execvp() and observe the differences in behaviors of the commands
- Creating a child process without terminating the parent process Write a program to create a child process using fork ().The parent should goto sleep state and child process should begin its execution. In the child process, use execl() to execute the “ls” command.
- Write a program to illustrate the concept of orphan process ( Using fork() and sleep())
- Write a program that demonstrates the use of nice () system call. After a child process is started using fork (), assign higher priority to the child using nice () system call.
- Write a program to find the execution time taken for execution of a given set of instructions (use clock() function)

**Assignment 2:FCFS, SJF, Priority Scheduling, Round-robin scheduling**

- Write a program to simulate FCFS CPU-scheduling algorithm. Accept number of Processes as input. Also accept arrival time and CPU burst time for each process as input.

- The output should generate a Gantt chart, turnaround time and waiting time for each process. Also display the average turnaround time and average waiting time.
2. Write a program to simulate Non-Pre-emptive Shortest Job First (SJF) CPU scheduling algorithm. Accept the number of Processes and arrival time and CPU burst time for each process as input. The output should generate a Gantt chart, turnaround time and waiting time for each process. Also display the average turnaround time and average waiting time.
  3. Write a program to simulate Non-Pre-emptive Priority and Round robin CPU scheduling algorithm. Accept the number of Processes and arrival time, CPU burst time and priority for each process as input. Priorities should be in High to Low order (Example 1 is High and 5 is Low). For Round robin, consider time slice is 2 units.

### Assignment 3: Deadlock detection and avoidance

1. Write a program for deadlock detection considering a single instance of each resource using wait for graph. Write a program which will traverse the graph to check for cycles. If a cycle is detected, print the list of processes that are involved in the deadlock.

#### 2. Deadlock avoidance using Banker's Algorithm

Write a program to implement a Banker's Algorithm. Accept the total number of processes (n) and resource types (m) as input. Also accept the number of instances for each resource type, Allocation and Max of size "n x m" as input and perform the

- a) Show the contents of Available array of size "m"
- b) Calculate and display the contents of Need matrix of size "n x m"

Using Safety and Resource-Request algorithm perform the following operations:

- a) Check whether the system is in safe state or not
- b) If a request of size "m" arrives from process  $P_i$ , can it be granted immediately by keeping the system in safe state? Where  $0 \leq i \leq m-1$ .

### Assignment 4: Page Replacement Algorithms: FIFO, Optimal, LRU

Consider the following page reference string:

1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3

How many page faults would occur for the following page replacement algorithms assuming four frames? All frames are initially empty.

1. LRU 2. Optimal 3. FIFO

2. Write a program to simulate demand paging using FIFO, LRU, optimal page replacement algorithms. Assume the memory of "n" frames. Show the contents of the page after every page replacement in a frame and at the end show the total number of page faults accordingly. Reference String : 3,4,5,4,3,4,7,2,4,5,6,7,2,4,6

**Ahmednagar Jilha Maratha Vidya Prasarak Samaj's  
New Arts, Commerce and Science College, Ahmednagar (Autonomous)**

**Syllabus of S. Y. B.C.A. Science**

Under

**Faculty of Science**

<b>Semester – IV</b>	<b>Paper – VII</b>
<b>Course Code: BCA-SC 407 T</b>	<b>Title of the Course: Cloud Computing</b>
<b>Credits: 02</b>	<b>Total Lectures: 30 Hrs.</b>

### Course Outcomes (COs):

- a. To understand the principles and paradigm of Cloud Computing
- b. To appreciate the role of Virtualization Technologies
- c. Ability to design and deploy Cloud Infrastructure
- d. Understand cloud security issues and solutions

### Detailed Syllabus:

#### Unit I: Introduction to Cloud Computing

(06)

- 1.1 Overview
  - 1.1.1 Roots of Cloud Computing
  - 1.1.2 Layers and Types of Cloud
  - 1.1.3 Desired Features of a Cloud
  - 1.1.4 Benefits and Disadvantages of Cloud Computing,
- 1.2 Cloud Infrastructure Management
  - 1.2.1 Infrastructure as a Service Providers
  - 1.2.2 Platform as a Service Providers
  - 1.2.3 Challenges and Risks.
- 1.3 Cloud-Enabling Technology
  - 1.3.1 Broadband Networks and Internet Architecture
  - 1.3.2 Data Center Technology
  - 1.3.3 Virtualization Technology
  - 1.3.4 Web Technology
  - 1.3.5 Multitenant Technology
  - 1.3.6 Service Technology.

#### Unit II: Architecture, Services and Applications

(06)

- 2.1 Exploring the Cloud Computing Stack
- 2.2 Connecting to the Cloud
- 2.3 Infrastructure as a Service, Platform as a Service, Saas Vs. Paas,

- 2.4 Using PaaS Application Frameworks
- 2.5 Software as a Service Cloud Deployment Models
- 2.6 Public vs Private Cloud
- 2.7 Cloud Solutions
- 2.8 Cloud ecosystem
- 2.9 Service management
- 2.10 Computing on demand
- 2.11 Identity as a Service, Compliance as a Service
- 2.12 Future of cloud computing and Emerging trends.

**Unit III: Abstraction and Virtualization****(05)**

- 3.1 Introduction to Virtualization Technologies
- 3.2 Load Balancing and Virtualization
- 3.3 Understanding Hypervisors
- 3.4 Understanding Machine Imaging
- 3.5 Porting Applications
- 3.6 Virtual Machines
  - 3.6.1 Provisioning and Manageability
  - 3.6.2 Virtual Machine Migration Services
  - 3.6.3 Virtual Machine Provisioning and Migration in Action
  - 3.6.4 Provisioning in the Cloud Context
  - 3.6.5 Virtualization of CPU, Memory, I/O Devices
  - 3.6.6 Virtual Clusters and Resource management
  - 3.6.7 Virtualization for Data Center Automation

**Unit IV: Programming, Environments and Applications****(06)**

- 4.1 Features of Cloud and Grid Platforms
- 4.2 Programming Support
  - 4.2.1 Programming Support of Google App Engine
  - 4.2.2 Programming on Amazon AWS
  - 4.2.3 Microsoft Azure
  - 4.2.4 Emerging Cloud Software Environments
- 4.3 Understanding Core OpenStack Ecosystem
- 4.4 Applications
  - 4.4.1 Moving application to cloud
  - 4.4.2 Microsoft Cloud Services
  - 4.4.3 Google Cloud Applications
  - 4.4.4 Amazon Cloud Services
  - 4.4.5 Cloud Applications (Social Networking, E-mail, Office Services, Google Apps, Customer Relationship Management).

**Unit V: Security in the Cloud****(07)**

- 5.1 Security Overview
  - 5.1.1 Cloud Security Challenges and Risks
  - 5.1.2 Software-as-a-Service Security
- 5.2 Security Governance
  - 5.2.1 Risk Management
  - 5.2.2 Security Monitoring
  - 5.2.3 Security Architecture Design
- 5.3 Securities
  - 5.3.1 Data Security
  - 5.3.2 Application Security
  - 5.3.3 Virtual Machine Security
- 5.4 Identity Management and Access Control
- 5.5 Autonomic Security
  - 5.5.1 Autonomic Security Storage Area Networks,
- 5.6 Disaster Recovery in Clouds.

**Suggested Readings:**

1. Cloud Computing: Technologies and Strategies of the Ubiquitous Data Center, Brian J.S. Chee and Curtis Franklin, CRC Press, ISBN:9781439806128
2. Rajkumar Buyya, Christian Vecchiola, S. Thamarai Selvi, Mastering Cloud Computing: Foundations and Applications Programming McGraw Hill, ISBN:9781259029950, 1259029956
3. Kai Hwang, Geoffrey C Fox, Jack G Dongarra, Distributed and Cloud Computing, From Parallel Processing to the Internet of Things, Morgan Kaufmann Publishers, 2012

**Ahmednagar Jilha Maratha Vidya Prasarak Samaj's  
New Arts, Commerce and Science College, Ahmednagar (Autonomous)**

**Syllabus of S. Y. B.C.A. Science**

**Under**

**Faculty of Science**

<b>Semester – IV</b>	<b>Paper – VIII</b>
<b>Course Code: BCA-SC 408 P</b>	<b>Title of the Course: Cloud Computing (Lab)</b>
<b>Credits: 02</b>	<b>Total Lectures: 30 Hrs.</b>

**Course Outcomes (COs):**

- a. Articulate the main concepts, key technologies, strengths, and limitations of cloud computing and the possible applications for cloud computing
- b. Explain the core issues of cloud computing such as security, privacy, and interoperability.
- c. Identify problems, and analyze, evaluate various cloud computing solutions according to the applications used.
- d. Create and deploy a cloud using google, Amazon, cloud platform.

**Assignments:**

1. Working and Implementation of Infrastructure as a service.
2. Working and Implementation of Software as a service.
3. Working and Implementation of Platform as a services.
4. Practical Implementation of Storage as a Service
5. Working of Google drive to make spreadsheet and notes.
6. Working and Implementation of identity management
7. Write a program for web feed
8. Execute the step to Demonstrate and implementation of cloud on single sign on
9. Practical Implementation of cloud security.
10. Installing and Developing Application Using Google App Engine
11. Implement VMWareESXi Server
12. Using OpenNebula to manage heterogeneous distributed data center Infrastructure.
13. Implementation of Cloud Failure Cluster.
14. Managing and working of cloud Xen server.
15. Working with Aneka and demonstrate how to Managing cloud computing Resources .
16. Installation and configuration of cloud Hadoop and demonstrate simple query.

17. Create a sample mobile application using Amazon Web Service (AWS) account as a cloud service. Also provide database connectivity with implemented mobile application.

**Ahmednagar Jilha Maratha Vidya Prasarak Samaj's**  
**New Arts, Commerce and Science College, Ahmednagar (Autonomous)**

**Syllabus of S. Y. B.C.A. Science**

**Under**

**Faculty of Science**

<b>Semester – IV</b>	<b>Paper – IX</b>
<b>Course Code: 409 T(A)</b>	<b>Title of the Course: MIL: Foreign Language: Japanese</b>
<b>Credits: 02</b>	<b>Total Lectures: 30 Hrs.</b>

### Course Outcomes (COs):

- It will introduce students to the Japanese scripts – Hiragana, Katakana, and about 100 basic kanji's (Chinese characters).
- It will help students to learn the Japanese phonology.
- It will introduce students to the elementary Japanese structures containing about 700 words.
- It will help students to acquaint with Japan, Japanese Culture, Civilization etc.

### Detailed Syllabus:

#### UNIT I

(6)

- Introduction about Japan, geographical features, culture.
- Introduction and history of Japanese Script.
- Hiragana and Katakana Script.
- Greetings and classroom expressions.
- Self introduction.

#### UNIT II

(4)

- Introduction of demonstrative pronouns.
- Introduction of particles は, が, も, か.
- Introduction of verbs, their dictionary form, masu form, (simple present tense and simple future tense.)

**UNIT III****(8)**

- 3.1 Introduction of prepositions – なか, した, うえ, etc.
- 3.2 Introduction of basic sentence pattern i.e. ~に~が あります/います.( External pattern.)
- 3.3 Introduction of Japanese unique counting system, numerals.
- 3.4 Introduction of multiples of 100,1000,10,000.
- 3.5 Introduction of days, date, year, age, duration, clock.

**UNIT IV****(4)**

- 4.1 Introduction of い、な adjectives.
- 4.2 To make adverbs from adjectives.
- 4.3 Past tense, present tense, affirmation, negation in case of verb, naa an see adjective.

**UNIT V ORAL****(8)**

- 5.1 Reading/listening/speaking practice.
- 5.2 Speech, self-introduction in Japanese.

**Suggested Readings:**

Textbooks prescribed: (Pertinent portions)

1. Minna No Nihongo I –Author: 3A Corporation, Publisher: Goyal Pub.& Dist.
2. Nihongo shoho Vol. I – Publisher: Japan Foundation, Tokyo, Japan (paperback edition available with JALTAP, Pune) Author- Japan Foundation.
3. Kanji Picture book Publisher: Japan foundation. Author- Japan Foundation.
4. Sulabh Japani Vyakarana – Part I Author - Dr. V. N. Kinkar, Publisher: JALTAP, Pune.



**Ahmednagar Jilha Maratha Vidya Prasarak Samaj's**  
**New Arts, Commerce and Science College, Ahmednagar (Autonomous)**  
**Syllabus of S. Y. B.C.A. Science**  
**Under**  
**Faculty of Science**

<b>Semester – IV</b>	<b>Paper – IX</b>
<b>Course Code: 409 T(B)</b>	<b>Title of the Course: MIL: Foreign Language: French</b>
<b>Credits: 02</b>	<b>Total Lectures: 30 Hrs.</b>

### Course Outcomes (COs):

- a. Identify the main point and basic supporting details of level-appropriate authentic texts from different
- b. Talk about familiar topics related to the self, personal interest and everyday life, using linked sentences with a moderate level of accuracy both orally and in writing;
- c. Ask and answer questions about the self, personal interest, everyday life, and the immediate environment, orally and in writing;
- d. Combine sentences and phrases to express opinions on topics related to the self, personal interest, and everyday life;

### Detailed Syllabus:

#### FRENCH SYLLABUS - 30 HOURS COURSE

#### \*ENCHANTÉ 1- 15 HOURS\*

LEÇON	OBJECTIFS DE LA LEÇON	GRAMMAIRE
0.BONJOUR LA FRANCE !	*CONNAISSANCE GÉNÉRALE DE LA FRANCE  *DISCUTANT EN GÉNÉRALE DE LA	*LES NOMS

	FRANCE	
1.LES LETTRES ET LES NOMBRES	*EPELER LE NOM ET COMPTER LES NOMBRES  *APPRENDRE LES ALPHABETS	*LES ACCENTS ,LES FORMULES POUR DEMANDER LE NOM
2.LES SALUTATIONS	*SAVOIR SALUER, REMERCIER,S' EXCUSER ( ENTRE CAMARADES ET ENTRE ADULTES)	*FORMULES DE SALUTATIONS ( UNE CONVERSATION FORMELLE ET INFORMELLE)  * CONNAÎTRE LES FORMULES POUR PRENDRE CONGÉ
3.À LA CANTINE	*SAVOIR UTILISER LE VERBE ÊTRE.  *APPRENDRE LES PRONOMS PERSONNELS.	*CONJUGAISON DU VERBE ÊTRE, LES PRONOMS
4.LES OBJECTD ET LES GENS	*PRÉSENTER QUELQU'UN OU QUELQUE CHOSE.	*LES ADJECTIFS, EXPRESSIONS AVEC QU'EST-CE QUE.....? ET QUI EST -CE .....?
5.LA CHAMBRE D' ALAIN	*SAVOIR UTILISER LE VERBE AVOIR ,LES ARTICLES DÉFINIS ET LES NOMBRES JUSQU' À CENT.	*CONJUGAISON D' AVOIR, LES ARTICLES DÉFINIS.
6.LE CIEL BLEU,LE PARC VERT	*SAVOIR CONJUGUER LE VERBE ALLER ,ACCORDER LES ADJECTIFS DE COULEUR	*CONJUGAISON DU VERBE ALLER,LES ADJECTIFS LES COULEURS

7.C'EST LE WEEK-END !	*CONNAÎTRE LES JOURS DE LA SEMAINE ET LES MOIS DEMANDEZ LE JOUR	*VOCABULAIRE DES JOURS ET DES MOIS
8.UNE SORTIE ENTRE AMIS	*SAVOIR CONJUGUER LES VERBES EN- ER	*RÈGLES DE CONJUGAISONS DU PREMIER GROUPE DE VERBE.
9.OU HABITES - TU ?	*SAVOIR DIRE SA NATIONALITÉ ET SON PAYS.	*VERBE HABITER, ACCORD DES NATIONALITÉS
10.LA FAMILLE D ' ALAIN	*SAVOIR UTILISER LES ADJECTIFS POSSESSIFS	*LES ADJECTIFS POSSESSIFS

**\*ENCHANTÉ 2 - 15 HOURS\***

LEÇON	OBJECTIFS DE LA LEÇON	GRAMMAIRE
0.DÉCOUVRONS LA FRANCE !	*IDENTIFIER QUELQUES SYMBOLES DE FRANCE	*LA CULTURE FRANÇAISE
1.BELLES IMAGES	*L'ACCORD DES ADJECTIFS	*LE PLURIEL DES NOMS * LES ADJECTIFS
2.VOICI PAUL!	*DÉCRIRE QUELQU'UN	*LES PARTIES DU CORPS *LA POSITION DES ADJECTIFS
3.LES HABITS	*SAVOIR CONJUGUER LES TROIS GROUPES DE VERBE ( ER ,IR ,RE )	*LES VERBES *RÈGLES DES VERBES * LES VÊTEMENTS
4.LES ANIMAUX DU ZOO	*MONTRER SA POSSESSION	*LES ADJECTIFS POSSESSIFS *LES NOMBRES ORDINAUX

		*LES ANIMAUX *GER ENDINGS VERBS
5.QUEL TEMPS FAIT- IL?	*APPRENDRE LES EXPRESSIONS COURANTES AVEC FAIRE ET AVOIR	*LES SAISONS *LES EXPRESSIONS AVEC FAIRE ET AVOIR
6.AU TRAVAIL!	*PARLER DES PROFESSIONS RÉPONDRE AU NÉGATIF	*LA NEGATION *LES PROFESSIONS *LES MÉTIER
7.CHEZ CLARA	*FAITES DES PHRASES AVEC LES PRÉPOSITIONS	*LES PRÉPOSITIONS *LES VERBES IRRÉGULIERS - RE
8.LA MAISON DE MARIE	*DÉCRIRE LA MAISON	*VOCABULAIRE DE LA MAISON *LES ARTICLES CONTRACTÉS *LES VERBES EN- TIR
9.BON APPÉTIT	*EXPRIMER LA QUANTITÉ, DÉCRIRE LES REPAS D'UNE JOURNÉE.	*LES ARTICLES PARTITIFS *LES VERBES IRRÉGULIERS *LA NÉGATION AVEC L'ARTICLE PARTITIF
10.MA VILLE ,MON QUARTIER.	*APPRENDRE À POSER DES QUESTIONS, POSER LA QUESTIONS AVEC " EST- CE QUE ....?"	*LES VERBES IRRÉGULIER *EST - CE QUE .

**Suggested Readings:**

1. Enchanté 0,1,2,3,4

Méthode de Français,

Author: Archna Khurana

Rachna Sagar

2. Apprenons Le Français

Mahitha Ranjit

Saraswati House

3. Encore En Échanges

Neelima Raddi

Anjali Paranjpye

Oxford University Press

4. Langue et Civilisation Françaises

G.Mauger

Goyal Publishers