"Dissemination of Education for Knowledge, Science and Culture" -Shikshanmaharshi Dr. Bapuji Salunkhe



Shri Swami Vivekanand Shikshan Sanstha's VIVEKANAND COLLEGE, KOLHAPUR (AN EMPOWERED AUTONOMOUS INSTITUTE)

DEPARTMENT OF COMPUTER SCIENCE Three/Four- Years UG Programme Department/Subject Specific Core or Major (DSC)

NEP- 2.0

Curriculum, Teaching and Evaluation Structure

(as per NEP-2020 Guidelines)

for

B.Sc.-II Computer Science

Semester-III & IV

(From the academic year 2025-26)

Department of Computer Science

B.Sc.: Program Outcomes (POs):

- **PO 1: Disciplinary Knowledge:** Graduates will gain in-depth understanding in their specific major or discipline, mastering the foundational principles and theories, as well as advanced concepts. Execute theoretical and practical knowledge developed from the specific curriculum.
- **PO 2: Problem-Solving Skills:** Graduates will learn to use their knowledge to identify, analyze and solve problems related to their field of study.
- **PO 3: Analytical Skills:** Graduates will gain the ability to collect, analyze, interpret, and apply data in a variety of contexts. They might also learn to use specialized software or equipment.
- **PO 4: Research Skills and Scientific Temper:** Graduates might learn how to design and conduct experiments or studies, analyze results and draw conclusions. They might also learn to review and understand academic literature.
- **PO 5: Environment and Sustainability:** Possess a sympathetic awareness of the environment while conducting research and scientific studies and focus on sustainable social development.

B.Sc. in Computer Science: Program Specific Outcomes (PSOs):

- **PSO1:** To obtain sound knowledge in the theory, principles and applications of Computer Science.
- **PSO2:** Develop and build strong problem solving, analyzing and decision-making skills and abilities to solve multidisciplinary problems of various domains associated with Computer Science.
- **PSO3:** Apprised with current trends and concepts of Computer Science and its applications in order to get competence for getting professional status.
- **PSO4:** Ability to understand the computing needs of multidisciplinary problems and to formulate and solve these by applying programming skills, development tools and environments.
- **PSO5:** Integrate knowledge of Computer Science with associated subjects like mathematics, statistics, electronics etc. to build and explore problem solving concepts.
- **PSO6:** Develop skills and knowledge (communication, problem solving, interviewing, creativity, team work, group discussion, aptitude etc.) to make or to build competence among the stakeholders to achieve the career in different fields and at a different level (Industry personnel, academician, researcher, entrepreneur etc).
- **PSO7:** Understand professional and ethical responsibilities in order to work at different positions in organizations and at a societal context.

Department of Computer Science Departmental Teaching and Evaluation scheme Second Year Semester-III & IV

Sr. No.	Course Abbr.	Course code	Course Name	Teac Sch Hours	ching eme s/week	Examination Scheme an Marks		e and	Course Credits	
				TH	PR	ESE	CIE	PR	Marks	
			Semester	Durse Name Teaching Scheme Hours/week Examination Scheme and Marks Course Course (redits) Semester-III PR ESE CIE PR Marks Course (redits) rating System-I 2 - 40 10 - 50 2 opject Oriented rogramming 2 - 40 10 - 50 2 ction to Operating System-I 2 - 40 10 - 50 2 of Object-Oriented rogramming 2 - 40 10 - 50 2 a Script Lab-I - 4 - - 25 25 2 omputer Sci Lab-3 - 8 - - 50 50 4 omputer Sci Lab-3 - 40 100 - 50 2 a Script Lab-II 2 - 40 100 - 50 2 tating System-II 2 - 40 10 - 50						
1	DSC-V	2DSC03CSC31	Operating System-I	2	-	40	10	-	50	2
2	DSC-VI	2DSC03CSC32	Object Oriented Programming	2	-	40	10	-	50	2
3	MIN-V	2MIN03CSC31	Introduction to Operating System-I	2	-	40	10	I	50	2
4	MIN-VI	2MIN03CSC32	Basics of Object-Oriented Programming	2	-	40	10	-	50	2
5	VSC-PR-II	2VSC03CSC39	Java Script Lab-I	-	4	-	-	25	25	2
6	DSC-PR-III	2DSC03CSC39	DSC- Computer Sci Lab-3	-	8	-	-	50	50	4
7	MIN-PR-III	2MIN03CSC39	MIN- Computer Sci Lab-3	-	4	-	-	25	25	2
Semester –III Total			8	16	160	40	100	300	16	
Semester-IV										
1	DSC-VII	2DSC03CSC41	Operating System-II	2	-	40	10	-	50	2
2	DSC-VIII	2DSC03CSC42	Algorithm Analysis and Data Structures	2	-	40	10	-	50	2
3	MIN-VII	2MIN03CSC41	Introduction to Operating System-II	2	-	40	10	-	50	2
4	MIN-VIII	2MIN03CSC42	Introduction to Data Structures	2	-	40	10	_	50	2
5	VSC-PR-III	2VSC03CSC49	Java Script Lab-II	-	4	-	-	25	25	2
6	DSC-PR-IV	2DSC03CSC49	DSC- Computer Sci Lab-4	-	8	-	-	50	50	4
7	MIN-PR-IV	2MIN03CSC49	MIN- Computer Sci Lab-4	-	4	-	-	25	25	2
		Semester –IV	' Total	8	16	160	40	100	300	16

Abbreviations: TH-Theory, PR-Practical, PRO- Project, SEE- Semester End Examination, CIE-Continuous Internal Examination Note:

Minimum passing for 10 marks Internal evaluation

- = 04 marks
- Minimum passing for 40 marks Theory paper Minimum passing for 25 marks Practical
- = 16 marks
- = 10 marks

Passing percentage for Democracy, Election and Good Governance (DEGG) and Environmental Studies papers should be 40% Separate passing for each Head - SEE, CIE and Practicals

Semester -III

B. Sc. Part – II Semester -III COMPUTER SCIENCE DSC-V: 2DSC03CSC31: Operating System-I Credits: 02 Theory: 30hrs.

Course Outcomes: After the completion of the course the student will be able to:

CO1: Understand the Fundamentals of Operating Systems.

CO2: Analyze Process and CPU Scheduling Mechanisms.

CO3: Gain Proficiency in Linux Operating System.

CO4: Develop Hands-on Skills with Linux Commands and Utilities.

Unit -1: Introduction to Operating System:

What Operating Systems Do, Computer-System Organization, Computer- System Architecture, Operating-System Structure, Operating-System Operations: Process Management, Memory Management, Storage Management, Protection and Security Distributed Systems, Special-Purpose Systems, Computing Environments, Operating-System Services, User Operating-System Interface, System Calls, Types of System Calls, System Programs, Virtual Machines, Operating-System Generation, System Boot.

Unit -2: Process Management:

Processes- Process Concept, Process Scheduling, Operations on Processes, Interprocess Communication, Examples of IPC Systems

Thread- Threads.

CPU Scheduling-Scheduling Criteria, Scheduling Algorithms (First-Come,

First-Served Scheduling, Shortest-Job-First Scheduling, Priority Scheduling, Round-Robin Scheduling, Multilevel Queue Scheduling).

Unit -3: Introduction to Linux:

Linux History and architecture of Linux system, Shell, Types of Shell's, Kernel, Kernel shell relationship, Login, Logout, Remote login, GPU(General Purpose Utilities) clear, script, cal, who, bc, wc, head, tail, inodes, structure of regular file, file manipulation commands, change file access permissions with chmod command, directories, directory management commands- cd, mkdir, rmdir. Simple filters- cut, paste, sort, tr, Advanced filters-sed, grep, gawk.

Reference Books:

- 1. Peter Baer Galvin, Operating System Concepts 5th edition, Addison-Wesley
- 2. Gary Nutt, Operating Systems 3rd edition, Pearson, 2007
- 3. Maurice J. Bach: The Design of The Unix Operating System, Prentice Hall, 2010

(10 Lectures)

e,

(10 Lectures)

(10 Lectures)

Marks-50

B. Sc. Part – II Semester - III COMPUTER SCIENCE DSC-VI: 2DSC03CSC32: Object Oriented Programming Marks-50 Credits: 02 Theory: 30hrs.

Course Outcomes: After the completion of the course the student will be able to:

CO1: Understand Object-Oriented Concepts

CO2: Apply Object-Oriented Techniques in Python

CO3: Demonstrate Inheritance and Polymorphism

CO4: Implement Advanced OOP Features

Unit -1: Introduction to Object Oriented Programming:

Programming Paradigms, What Is Object-Oriented Programming?, Features of OOP, Advantages and disadvantage of OOP, Function Overloading, Operator Overloading, Static and Dynamic Binding, Constructors and Destructors, Techniques of Object-Oriented Programming, When to use OOP?, Applications of OOP.

Unit -2: Classes and Objects:

Python Classes, Objects, Specifying attributes and behaviors, instance methods, instance attributes, static methods, constructor, types of constructors(default, parameterized), class methods as alternative constructor, constructor overloading, method overloading.

Unit -3: Inheritance and Polymorphism

Inheritance in Python (Syntax, Advantages,), Access Modifiers in Python, Types of Inheritance (single, multiple, multilevel, hierarchical and hybrid), Polymorphism-Method Overriding, magic methods and Operator Overloading.

Reference Books:

- 1. Michael H Goldwasser & David Letscher: Object-Oriented Programming in Python 1st Edition, Prentice Hall, 2007.
- 2. Dusty Phillips: Python 3 Object Oriented Programming, PACKT Publishing, 2010
- 3. David Ascher, Alex Martelli: Python Cookbook, Oreilly

(10 Lectures)

(10 Lectures)

B. Sc. Part – II Semester - III COMPUTER SCIENCE

MIN-V: 2MIN03CSC31: Introduction to Operating System-I Theory: 30hrs. Marks-50 Credits: 02

Course Outcomes: After the completion of the course the student will be able to:

CO1: Understand the Fundamentals of Operating Systems.

CO2: Analyze Process and CPU Scheduling Mechanisms.

CO3: Gain Proficiency in Linux Operating System.

CO4: Develop Hands-on Skills with Linux Commands and Utilities.

Unit -1: Introduction to Operating System:

What Operating Systems Do, Computer-System Organization, Computer- System Architecture, Operating-System Structure, Operating-System Operations: Process Management, Memory Management, Storage Management, Protection and Security Distributed Systems, Special-Purpose Systems, Computing Environments, Operating-System Services, User Operating-System Interface, System Calls, Types of System Calls, System Programs, Virtual Machines, Operating-System Generation, System Boot.

Unit -2: Process Management:

Processes- Process Concept, Process Scheduling, Operations on Processes, Interprocess Communication, Examples of IPC Systems

Thread- Threads.

CPU Scheduling-Scheduling Criteria, Scheduling Algorithms (First-Come,

First-Served Scheduling, Shortest-Job-First Scheduling, Priority Scheduling, Round-Robin Scheduling, Multilevel Queue Scheduling).

Unit -3: Introduction to Linux:

Linux History and architecture of Linux system, Shell, Types of Shell's, Kernel, Kernel shell relationship, Login, Logout, Remote login, GPU(General Purpose Utilities) clear, script, cal, who, bc, wc, head, tail, inodes, structure of regular file, file manipulation commands, change file access permissions with chmod command, directories, directory management commands- cd, mkdir, rmdir. Simple filters- cut, paste, sort, tr, Advanced filters-sed, grep, gawk.

Reference Books:

- 1. Peter Baer Galvin, Operating System Concepts 5th edition, Addison-Wesley
- 2. Gary Nutt, Operating Systems 3rd edition, Pearson, 2007
- 3. Maurice J. Bach: The Design of The Unix Operating System, Prentice Hall, 2010

(10 Lectures)

(10 Lectures)

B. Sc. Part – II Semester -III COMPUTER SCIENCE MIN-VI: 2MIN03CSC32: Basics of Object Oriented Programming Credits: 02 Theory: 30hrs. Marks-50

Course Outcomes: After the completion of the course the student will be able to:

CO1: Understand Object-Oriented Concepts

CO2: Apply Object-Oriented Techniques in Python

CO3: Demonstrate Inheritance and Polymorphism

CO4: Implement Advanced OOP Features

Unit -1: Introduction to Object Oriented Programming:

Programming Paradigms, What Is Object-Oriented Programming?, Features of OOP, Advantages and disadvantage of OOP, Function Overloading, Operator Overloading, Static and Dynamic Binding, Constructors and Destructors, Techniques of Object-Oriented Programming, When to use OOP?, Applications of OOP.

Unit -2: Classes and Objects:

Python Classes, Objects, Specifying attributes and behaviors, instance methods, instance attributes, static methods, constructor, types of constructors(default, parameterized), class methods as alternative constructor, constructor overloading, method overloading.

Unit -3: Inheritance and Polymorphism

Inheritance in Python (Syntax, Advantages,), Access Modifiers in Python, Types of Inheritance (single, multiple, multilevel, hierarchical and hybrid), Polymorphism-Method Overriding, magic methods and Operator Overloading.

Reference Books:

- 1. Michael H Goldwasser & David Letscher: Object-Oriented Programming in Python 1st Edition, Prentice Hall, 2007.
- 2. Dusty Phillips: Python 3 Object Oriented Programming, PACKT Publishing, 2010
- 3. David Ascher, Alex Martelli: Python Cookbook, OReilly

(10 Lectures)

(10 Lectures)

B. Sc. Part – II Semester -III COMPUTER SCIENCE VSC-PR-II: 2VSC03CSC39: Java Script Lab-I Credits: 02 Practical: 30hrs.

Course Outcomes: After the completion of the course the student will be able to:

CO1: Understand and Apply JavaScript Syntax and Structure:

CO2: Effectively Embed and Execute JavaScript in Web Pages.

CO3: Utilize Variables and Data Types Proficiently.

CO4: Apply JavaScript Statements and Operators.

- 1. Write a JavaScript program to display "Hello, World!" in an alert box.
- 2. Create an external JavaScript file and embed it in an HTML page to display a welcome message.
- 3. Demonstrate the use of JavaScript comments (both single-line and multi-line).
- 4. Write a program to check JavaScript's case sensitivity using variable names.
- 5. Declare variables of different data types (number, string, boolean, null, undefined) and print their values.
- 6. Write a program to swap two numbers using JavaScript variables.
- 7. Demonstrate the use of different arithmetic operators (+, -, *, /, %, **).
- 8. Write a JavaScript program to compare two numbers using comparison operators and display the result.
- 9. Implement a program to check if a number is even or odd using logical operators.

Marks-25

B. Sc. Part – II Semester -III COMPUTER SCIENCE DSC-PR-III: 2DSC03CSC39: DSC- Computer Sci Lab-3 Credits: 04 Practical: 30hrs. Marks-50

- 1. Usage of following commands: ls, pwd, tty, cat, who, who am I, rm, mkdir, rmdir, touch, cd.
- 2. Usage of following commands: cal, cat(append), cat(concatenate), mv, cp, man, date.
- 3. Usage of following commands: chmod, grep, tput (clear, highlight), bc.
- 4. Write a shell script to check if the number entered at the command line is prime or not.
- 5. Write a shell script to modify "cal" command to display calendars of the specified months.
- 6. Write a shell script to modify "cal" command to display calendars of the specified range of months.
- 7. Write a shell script to accept a login name. If not a valid login name display message "Entered login name is invalid".
- 8. Write a shell script to display date in the mm/dd/yy format.
- 9. Write a shell script to display on the screen sorted output of "who" command along with the total number of users.
- 10. Write a shell script to display the multiplication table any number,
- 11. Program to demonstrate simple class and instance of a class.
- 12. Program to demonstrate use of class methods.
- 13. Program to demonstrate default constructor.
- 14. Program to demonstrate parameterized constructor.
- 15. Program to demonstrate constructor overloading.
- 16. Program to demonstrate destructor.
- 17. Program to demonstrate visibility modes.
- 18. Program to implement single inheritance.
- 19. Program to implement multiple inheritance.
- 20. Program to implement multilevel inheritance.
- 21. Program to implement hybrid inheritance.
- 22. Program to demonstrate method overriding.
- 23. Program to demonstrate operator overloading using magic method
- 24. Program to demonstrate method overloading.

B. Sc. Part – II Semester -III COMPUTER SCIENCE MIN-PR-III: 2MIN03CSC39: MIN- Computer Sci Lab-3 Credits: 02 Practical: 30hrs. Marks-25

- 1. Usage of following commands: ls, pwd, tty, cat, who, who am I, rm, mkdir, rmdir, touch, cd.
- 2. Usage of following commands: cal, cat(append), cat(concatenate), mv, cp, man, date.
- 3. Usage of following commands: chmod, grep, tput (clear, highlight), bc.
- 4. Write a shell script to check if the number entered at the command line is prime or not.
- 5. Write a shell script to modify "cal" command to display calendars of the specified months.
- 6. Write a shell script to modify "cal" command to display calendars of the specified range of months.
- 7. Write a shell script to accept a login name. If not a valid login name display message "Entered login name is invalid".
- 8. Write a shell script to display date in the mm/dd/yy format.
- 9. Write a shell script to display on the screen sorted output of "who" command along with the total number of users.
- 10. Write a shell script to display the multiplication table any number,
- 11. Program to demonstrate simple class and instance of a class.
- 12. Program to demonstrate use of class methods.
- 13. Program to demonstrate default constructor.
- 14. Program to demonstrate parameterized constructor.
- 15. Program to demonstrate constructor overloading.
- 16. Program to demonstrate destructor.
- 17. Program to demonstrate visibility modes.
- 18. Program to implement single inheritance.
- 19. Program to implement multiple inheritance.
- 20. Program to implement multilevel inheritance.
- 21. Program to implement hybrid inheritance.
- 22. Program to demonstrate method overriding.
- 23. Program to demonstrate operator overloading using magic method
- 24. Program to demonstrate method overloading.

Semester -IV

B. Sc. Part – II Semester -IV COMPUTER SCIENCE DSC-VII: 2DSC03CSC41: Operating System-II Credits: 02 Theory: 30hrs.

Course Outcomes: After the completion of the course the student will be able to:

CO1: Understand Memory Management Techniques

CO2: Apply Storage Management Concepts.

CO3: Analyze I/O System and Kernel-Level Operations.

CO4: Develop Shell Scripts for System Automation.

Unit -1: Memory Management:

Main Memory-Swapping, Contiguous Memory Allocation, Paging, Structure of the Page Table, Segmentation, Example: The Intel Pentium, Virtual Memory-Demand Paging, Copy-on-Write, Page Replacement (FIFO, Optimal, LRU, MFU,LFU), Allocation of Frames, Thrashing, Memory-Mapped Files

Unit -2: Storage Management:

File-System Interface-File Concept, Access Methods, Directory Structure, File-System Mounting, File Sharing, Protection, File-System Structure, File-System Implementation, Directory Implementation, Allocation Methods, Free-Space Management, Efficiency and Performance, I/O Systems-I/O Hardware, Application I/O Interface, Kernel I/O Subsystem.

Unit -3: Linux Scripting:

Writing and running the shell script, read, echo, decisions and loop control structure, file tests, exit, command line arguments, exporting shell variable, arrays, shell function, writing data entry script to create data files, data validations before storing on hard disk.

Reference Books:

- 1. Peter Baer Galvin, Operating System Concepts 5th edition, Addison-Wesley
- 2. Gary Nutt, Operating Systems 3rd edition, Pearson, 2007
- 3. Maurice J. Bach: The Design of The Unix Operating System, Prentice Hall, 2010

Department of Computer Science

(10 Lectures)

(10 Lectures)

(10 Lectures)

Marks-50

B. Sc. Part – II Semester - IV COMPUTER SCIENCE DSC-VIII: 2DSC03CSC42: Algorithm Analysis and Data Structures Theory: 30hrs. Credits: 02 Marks-50

Course Outcomes: After the completion of the course the student will be able to:

CO1: Understand Abstract Data Types and Algorithm Efficiency

CO2: Implement Linked List Variants for Efficient Data Handling

CO3: Apply Stack Operations for Problem-Solving

CO4: Implement Queue and Priority Queue Structures

Unit -1: Abstract Data Type:

Introduction: Abstractions, Abstract Data Types, Data Structures, General Definitions; The Date Abstract Data Type: Defining the ADT, Using the ADT, Preconditions and Postconditions, Implementing the ADT; Bags: The Bag Abstract Data Type, Selecting a Data Structure, List-Based Implementation; Iterates: Designing an Iterator, Using Iterators; Application: Student Records, Designing a Solution, Implementation Algorithm Analysis: Complexity Analysis: Big-O Notation, Evaluating Python Code; Evaluating the Python List; Amortized Cost; Application: The Sparse Matrix, List-Based Implementation, Efficiency Analysis.

Unit -2: Linked Structure:

The singly Linked List: Traversing the node, Searching for a node, Prepending Nodes, Removing Nodes; The Bag ADT Revisited: A linked List Implementation, Comparing Implementations, Linked list iterators; More Ways to Build a Linked List: Using a Tail Reference, The sorted linked list; The Sparse Matrix Revisited: An array of Lined list implementation, Comparing the Implementations; Applications: Polynomials, Polynomial Operations, The Polynomial ADT, Implementation. Advanced Linked List: The Doubly Linked List: Organization, List Operations; Circular Linked List: Organization, List Operation Multi-Linked Lists: Multiple Chains, The sparse Matrix; Complex Iterators; Application: Text Editor, Typical Editor Operations, The EDIT Buffer ADT, Implementation.

Unit -3: Stacks and Queues

The Stack ADT: Implementing the stack, using a python list, using a linked list, Stack Appliactions: Balanced Delimiters, Evaluating Postfix Expression; Applicactions: Solving a Maze: Backtracking, Designing a solution, The Maze ADT, Implementation

The Queue ADT;Implementing the Queue:Using a Python List, Using a Circular Array, Using a Linked List Priority Queues: The priority Queue ADT, Implementation: Unbounded Priority Queue, Implementation :Bounded Priority Queue ;Application : Computer Simulation : Airline Ticket Counter,

Implementation.

Reference Books:

- 1. Rance D. Necaise: Data structures and algorithms using python, Wileys, 2016
- 2. Richard Petersen, Linux: The Complete Reference, Sixth Edition, McGrawHill, 2008

(10 Lectures)

(10 Lectures)

B. Sc. Part – II Semester - IV COMPUTER SCIENCE MIN-VII: 2MIN03CSC41: Introduction to Operating System-II Theory: 30hrs. Marks-50 Credits: 02

Course Outcomes: After the completion of the course the student will be able to:

CO1: Understand Memory Management Techniques

CO2: Apply Storage Management Concepts.

CO3: Analyze I/O System and Kernel-Level Operations.

CO4: Develop Shell Scripts for System Automation.

Unit -1: Memory Management:

Main Memory-Swapping, Contiguous Memory Allocation, Paging, Structure of the Page Table, Segmentation, Example: The Intel Pentium, Virtual Memory-Demand Paging, Copy-on-Write, Page Replacement (FIFO, Optimal, LRU, MFU,LFU), Allocation of Frames, Thrashing, Memory-Mapped Files

Unit -2: Storage Management:

File-System Interface-File Concept, Access Methods, Directory Structure, File-System Mounting, File Sharing, Protection, File-System Structure, File-System Implementation, Directory Implementation, Allocation Methods, Free-Space Management, Efficiency and Performance, I/O Systems-I/O Hardware, Application I/O Interface, Kernel I/O Subsystem.

Unit -3: Linux Scripting:

Writing and running the shell script, read, echo, decisions and loop control structure, file tests, exit, command line arguments, exporting shell variable, arrays, shell function, writing data entry script to create data files, data validations before storing on hard disk.

Reference Books:

- 1. Peter Baer Galvin, Operating System Concepts 5th edition, Addison-Wesley
- 2. Gary Nutt, Operating Systems 3rd edition, Pearson, 2007
- 3. Maurice J. Bach: The Design of The Unix Operating System, Prentice Hall, 2010

Department of Computer Science

(10 Lectures)

(10 Lectures)

B. Sc. Part – II Semester - IV COMPUTER SCIENCE MIN-VIII: 2MIN03CSC42: Introduction to Data Structures Credits: 02 Theory: 30hrs. Marks-50

Course Outcomes: After the completion of the course the student will be able to:

CO1: Understand Abstract Data Types and Algorithm Efficiency

CO2: Implement Linked List Variants for Efficient Data Handling

CO3: Apply Stack Operations for Problem-Solving

CO4: Implement Queue and Priority Queue Structures

Unit -1: Abstract Data Type:

Introduction: Abstractions, Abstract Data Types, Data Structures, General Definitions; The Date Abstract Data Type: Defining the ADT, Using the ADT, Preconditions and Postconditions, Implementing the ADT; Bags: The Bag Abstract Data Type, Selecting a Data Structure, List-Based Implementation; Iterates: Designing an Iterator, Using Iterators; Application: Student Records, Designing a Solution, Implementation Algorithm Analysis: Complexity Analysis: Big-O Notation, Evaluating Python Code; Evaluating the Python List; Amortized Cost; Application: The Sparse Matrix, List-Based Implementation, Efficiency Analysis.

Unit -2: Linked Structure:

(10 Lectures) The singly Linked List: Traversing the node, Searching for a node, Prepending Nodes, Removing Nodes; The Bag ADT Revisited: A linked List Implementation, Comparing Implementations, Linked list iterators; More Ways to Build a Linked List: Using a Tail Reference, The sorted linked list; The Sparse Matrix Revisited: An array of Lined list implementation, Comparing the Implementations; Applications: Polynomials, Polynomial Operations, The Polynomial ADT, Implementation. Advanced Linked List: The Doubly Linked List: Organization, List Operations; Circular Linked List: Organization, List Operation Multi-Linked Lists: Multiple Chains, The sparse Matrix; Complex Iterators; Application: Text Editor, Typical Editor Operations, The EDIT Buffer ADT, Implementation.

Unit -3: Stacks and Queues

The Stack ADT: Implementing the stack, using a python list, using a linked list, Stack Appliactions: Balanced Delimiters, Evaluating Postfix Expression; Applicactions: Solving a Maze: Backtracking, Designing a solution, The Maze ADT, Implementation

The Queue ADT;Implementing the Queue:Using a Python List, Using a Circular Array, Using a Linked List Priority Queues: The priority Queue ADT, Implementation: Unbounded Priority Queue, Implementation :Bounded Priority Queue ;Application : Computer Simulation : Airline Ticket Counter, Implementation.

Reference Books:

- 1. Rance D. Necaise: Data structures and algorithms using python, Wileys, 2016
- 2. Data Structures and Algorithms in Python by Michael T. Goodrich

(10 Lectures)

B. Sc. Part – II Semester -IV COMPUTER SCIENCE VSC-PR-III: 2VSC03CSC49: Java Script Lab-II Credits: 02 Practical: 30hrs. Marks-25

Course Outcomes: After the completion of the course the student will be able to:
CO1: Implement and Manipulate JavaScript Objects and Arrays.
CO2: Apply JavaScript Event Handling Techniques for Interactive Web Applications.
CO3: Utilize the Document Object Model (DOM) for Dynamic Web Manipulation.
CO4: Work with the Window Object for Browser Control and User Interaction.

- 1. Implement a program to check if a number is even or odd using logical operators.
- 2. Write a JavaScript program to find the largest of three numbers using if-else statements.
- 3. Implement a switch-case statement to display the name of a day based on a number input (1-7).
- 4. Create a JavaScript program that prints numbers from 1 to 10 using a for loop.
- 5. Write a program to calculate the sum of even numbers from 1 to 50 using a while loop.
- 6. Implement a do-while loop that keeps asking for a number until a positive number is entered.
- 7. Write a JavaScript function to display an alert box when a button is clicked.
- Create a program that takes user input using the prompt() method and displays the entered text in a confirm() box.
- 9. Write a function that takes two numbers as parameters, adds them, and returns the sum.
- 10. Create a function that checks whether a given string is a palindrome or not.
- 11. Write a JavaScript program to create an object for a student with properties (name, age, marks) and display them.
- 12. Implement a program that creates an array of five numbers, sorts them in ascending order, and displays the result

B. Sc. Part – II Semester -IV COMPUTER SCIENCE DSC-PR-IV: 2DSC03CSC49: DSC- Computer Sci Lab-4 Credits: 04 Practical: 30hrs. Marks-50

- 1. Write a shell script to compare two files and if found equal asks the user to delete the duplicate file.
- 2. Write a shell script to find the sum of digits of a given number.
- 3. Write a shell script to merge the contents of three files, sort the contents and then display them page by page.
- 4. Write a shell script to find the LCD (least common divisor) of two numbers.
- 5. Write a shell script to perform the tasks of basic calculator.
- 6. Write a data entry script for student database with all validations.
- 7. Write a menu driven shell script to display today's date, present working directory, names of user, directory listing.
- 8. Write a shell script to check whether the number is Armstrong or not.
- 9. Write a shell script to check whether the file have all the permissions or not.
- 10. Program to show the pyramid of special character "*".
- 11. Program to demonstrate singly linked list (for insert, delete, display).
- 12. Program to demonstrate doubly linked list (insert, delete, display).
- 13. Program to demonstrate application of linked list- addition of polynomials.
- 14. Program to implement stack.
- 15. Program to demonstrate expression conversion of stack.
- 16. Program to demonstrate recursion using stack.
- 17. Program to implement of Circular Queue.

B. Sc. Part – II Semester -IV COMPUTER SCIENCE MIN-PR-IV: 2MIN03CSC49: MIN- Computer Sci Lab-4 Credits: 02 Practical: 30hrs. Marks-25

- 1. Write a shell script to compare two files and if found equal asks the user to delete the duplicate file.
- 2. Write a shell script to find the sum of digits of a given number.
- 3. Write a shell script to merge the contents of three files, sort the contents and then display them page by page.
- 4. Write a shell script to find the LCD (least common divisor) of two numbers.
- 5. Write a shell script to perform the tasks of basic calculator.
- 6. Write a data entry script for student database with all validations.
- Write a menu driven shell script to display today's date, present working directory, names of user, directory listing.
- 8. Write a shell script to check whether the number is Armstrong or not.
- 9. Write a shell script to check whether the file have all the permissions or not.
- 10. Program to show the pyramid of special character "*".
- 11. Program to demonstrate singly linked list (for insert, delete, display).
- 12. Program to demonstrate doubly linked list (insert, delete, display).
- 13. Program to demonstrate application of linked list- addition of polynomials.
- 14. Program to implement stack.
- 15. Program to demonstrate expression conversion of stack.
- 16. Program to demonstrate recursion using stack.
- 17. Program to implement of Circular Queue.

Question Paper Format:

Seat No.

Ques. Paper Code

Marks : 40

VIVEKANAND COLLEGE, KOLHAPUR

(AN EMPOWERED AUTONOMOUS INSTITUTE)

B.Sc. Part- II (Computer Science) (Semester-III) Examination.....

Course Code and Name: DSC03CSC31: Operating System-I Time: 1.30 hours

Day:

Date: --/--/

Instructions:

1) All the questions are compulsory.

2) Figures to the right indicate full marks.

3) Draw neat labeled diagrams wherever necessary.

4) Use of log table/calculator is allowed.

Q. 1. Select correct alternative (One mark each):

i)	Xyzabcdefghijklmnop							
ii)	a) Xyzabcdefghijklmno	b) p	c)	d)				
iii)	a) Xyzabcdefghijklmno	b) p	c)	d)				
iv)	a) Xyzabcdefghijklmno	b) p	c)	d)				
v)	a) Xyzabcdefghijklmno	b) p	c)	d)				
vi)	a) Xyzabcdefghijklmno	b) p	c)	d)				
vii)	a) Xyzabcdefghijklmno	b) p	c)	d)				
viii)	a) Xyzabcdefghijklmno	b) p	c)	d)				
	a)	b)	c)	d)				

Q.2. Attempt any TWO (Eight marks each):

- i) Xyzabcdefghijklmnop.
- ii) Xyzabcdefghijklmnop.
- iii) Xyzabcdefghijklmnop.

[16]

[8]

Q.3. Attempt any FOUR (Four marks each):

- i) Xyzabcdefghijklmnop.
- ii) Xyzabcdefghijklmnop.
- iii) Xyzabcdefghijklmnop.
- iv) Xyzabcdefghijklmnop.
- v) Xyzabcdefghijklmnop.
- vi) Xyzabcdefghijklmnop.

.....

[16]