Shri Swami Vivekanand Shikshan Sanstha's

# VIVEKANAND COLLEGE (AUTONOMOUS), KOLHAPUR

DEPARTMENT OF COMPUTER SCIENCE

# Syllabus for the S.Y.B.Sc.
# Program: (Undergraduate) B.Sc.
# Course: Computer Science

# SYLLABUS OF COURSE TO BE OFFERED
Core Courses, Elective Courses & Ability Enhancement Courses

Credit Based Semester and Grading System with effect from the academic year **2022–2023**

# STRUCTURE OF COURSE

| Sr. No | Paper | Name of Paper | Marks | Internal Assessment |
|---|---|---|---|---|
| **SEMESTER-III** | | | | |
| 1 | DSC-1006C | Operating System and Object Oriented Programming | 70 | 30 |
| 2 | SEC-1006C | Introduction to HTML and CSS | | |
| **SEMESTER-IV** | | | | |
| 3 | DSC-1006D | Operating System and Data Structures | 70 | 30 |
| 4 | SEC-1006D | Introduction to JavaScript | | |
| 5 | Practical Paper -I | Practical's based on DSC-1006C, SEC-1006C | 50 | |
| 6 | Practical Paper -II | Practical's based on DSC-1006D, SEC-1006D | 50 | |

# Semester III (Theory)

| Course:<br>DSC-1006C | TOPICS (Credits : 4 Lectures/Week: 5) Study of<br>Operating System and Object Oriented Programming | |
|---|---|---|
| Objectives – This course provides an introduction to the concepts, theories and components that serve as the bases for the design of classical and modern operating systems and Object oriented concepts in programming.<br><br>Expected Course Outcomes of this course<br>Students should be able to ….<br>1) Learn and understand the basics of operating system and Object-Oriented programming concepts.<br>2) Learn and understand the basics of LINUX operating system and Object-Oriented programming concepts using Python.<br>3) Define, Discuss and Explain the concepts of OS and features of OOPs<br>4) Illustrate and construct Linux commands and basics of OOPs<br>5) Solve and Analyse problems with object-oriented concepts<br>6) Compile, Design and Construct case studies | | |
| **SECTION – I** | | |
| Unit I | **Introduction**<br>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 | 10L |
| Unit II | **Process Management**<br>**Processes**- Process Concept, Process Scheduling, Operations on Processes, Interprocess Communication, Examples of IPC Systems<br>**Thread**- Threads.<br>**CPU Scheduling**-Scheduling Criteria, Scheduling Algorithms (First-Come, First-Served Scheduling, Shortest-Job-First Scheduling, Priority Scheduling, Round-Robin Scheduling, Multilevel Queue Scheduling) | 10 L |
| Unit III | **Introduction to Linux**<br>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 | 10 L |

| SECTION – II | | |
|---|---|---|
| **Unit I** | **Introduction to Object Oriented Programming**<br>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. | 10 L |
| **Unit II** | **Classes and Objects**<br>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. | 10 L |
| **Unit III** | **Inheritance and Polymorphism**<br>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. | 10L |

**Text books:**

**Additional References:**

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

| Course:<br>**SEC-1006C** | **TOPICS (Credits : 2 Lectures/Week)**<br>**Introduction to HTML and CSS** | |
|---|---|---|
| **Objectives – This course provides an introduction to the basic concepts in the web development as Hyper text markup language and Cascading Style Sheets.**<br><br>**Expected Learning Outcomes of this course**<br>Students should be able to …<br>1) learn the basic tags used in HTML.<br>2) develop their own Cascading Sheets in order to design web pages.<br>3) develop Static web pages.<br>4) Develop attractive interface for web application | | |
| **Details** | | |
| **Unit I** | **HTML Basics**<br>Introduction, html editors, extensions, html elements, element syntax, HTML fundamental-(Heading, formatting, Paragraphs, Links, Images), comments, HTML attributes.<br><br>**HTML Fonts, Color, Tables, Lists**<br>Font tag, font attributes, Color attribute, color values, web standard color names, Creating HTML Tables ,Table Borders, Headings in a Table, Table with a Caption, Cells Spanning Multiple Columns, Tags Inside a Table, Cell Padding, Spacing, Table/Cell Background Colors and Images, HTML Lists, Unordered Lists, Ordered Lists, Definition Lists, Nested Lists.<br><br>**HTML Forms, Input Tag, Layout, Head and Meta element**<br>Form tag, form attributes, Input tag, Input tag attributes, Layout using Table, Frames-(frame tag, its attributes, frameset tag, its attributes), Head tag with its uses, Meta tag, its attributes, its uses. | 12L |
| **Unit II** | **CSS Basics**<br>Introduction, Syntax, Comments, Selectors, inserting style sheets (internal, external, inline), Multiple style sheets.<br>**CSS Styling**<br>Backgrounds, Texts, Fonts, Links, Lists, Tables, Box model (Border, Outline, Margin and Padding)<br><br>**CSS Advanced**<br>CSS Dimension, CSS Display, CSS Positioning, CSS Floating, CSS Align, CSS Pseudo-class, CSS Pseudo-element, CSS Navigation Bar, CSS Image Gallery, CSS Image Opacity, CSS Media Types, CSS Attribute Selectors, Building Forms. | 12 L |

# SEMESTER – IV (Theory)

| Course: DSC-1006D | TOPICS (Credits : 4 Lectures/Week: 5)<br>Operating System and Data Structures | |
|---|---|---|
| **Objectives – The objective of this course is to impart the basic concepts of data structures and understanding about writing algorithms and step by step approach in solving problems with the help of fundamental data structures.**<br><br>**Expected Course Outcomes: Through this course there will be an enhancement to**<br>1) Learn and understand the related and extended concepts of OS and basics of Data Structures.<br>2) Learn and understand the concepts of LINUX scripting and data structures.<br>3) Illustrate and construct Linux shell scripts and different types of data structures in Python<br>4) Solve and Analyze problems using shell scripts and different types of data structures<br>5) Explain and Determine scope of shell scripting and data structures in different applications<br>6) Construct and Build case studies of shell scripting and Data Structures | | |
| **SECTION - I** | | |
| **Unit I** | **Memory Management**<br>**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 | 10 L |
| **Unit II** | **Storage Management**<br>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 | 10 L |
| **Unit III** | **Linux Scripting**<br>Writing and running the shell script,<br>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. | 10 L |

| SECTION - II | | |
|---|---|---|
| **Unit I** | **Abstract Data Type**<br>**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 | 15 L |
| **Unit II** | **Linked Structure**<br>**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 | 15 L |
| **Unit III** | **Stacks**<br>**The Stack ADT:** Implementing the stack, using a python list, using a linked list, Stack Appliactions: Balanced Delimiters, Evaluating Postfix Expression;<br>**Applicactions:** Solving a Maze: Backtracking, Designing a solution, The Maze ADT, Implementation<br><br>**Queues**<br>**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 | 15 L |

**Text books:**

**Additional References:**

1. Peter Baer Galvin, *Operating System Concepts 5$^{th}$ edition, Addison-Wesley*
2. Gary Nutt, *Operating Systems – 3$^{rd}$ edition,* Pearson, 2007
3. Maurice J. Bach*: The Design of The Unix Operating System,* Prentice Hall, 2010
4. Rance D. Necaise: *Data structures and algorithms using python,*Wileys, 2016
5. Richard Petersen, *Linux: The Complete Reference*, Sixth Edition, McGrawHill, 2008

| Course:<br>**SEC-1006D** | **TOPICS (Credits : 2 Lectures/Week)**<br>**Introduction to JavaScript** | |
|---|---|---|
| **Objectives – This course provides an introduction to JavaScript in order to enhance the developed web contents.**<br><br>**Expected Learning Outcomes of this course**<br>1) Students should to learn the basic concepts of JavaScript.<br>2) Student should use their own JavaScript in order to develop web pages. |
| **Details** |
| **Unit I** | **Introduction to JavaScript**<br>Definition, Uses, Introduction to JavaScript and Events, Version, Embedding JavaScript within HTML, from external files, Execution, syntax and statements details(case sensitivity , comments, use of semicolon, reserved words, statement definition, types ).<br><br>**Building Blocks**<br>Data types (number, string, boolean, null, undefined ), Variables(variable declaration, types, scope), operators(Arithmetic , Assignment, Comparison, logical, bitwise, special).<br><br>**Control Structure**<br>If, else if, else (statements), switch statement, ternary conditional, while (while, do-while) statement, for loop ( for, for…in, for each…in), use of break, continue.<br><br>**Dialog Boxes and Functions**<br>Dialog boxes (alert(), prompt(), confirm()), Function- Definition, Declaration and invocation, argument passing, return statement, function calling-( in body section, from another function, with parameters or without parameters) | 12L |
| **Unit II** | **Introduction to Objects**<br>Definition, creating object, adding method, Arrays-creating array object, accessing array element, properties, methods of an array , String- creating string object, properties and methods of String object, Math- creating Math object, properties and methods of Math object, Number - creating Number object, properties and method of Numbers object and Date- creating Date object, properties and methods of Date object<br><br>**Event handing, Document Object and Window Object**<br>Event handing Introduction, uses, locations, use on html element, use in java script, learning events(change, focus ,click, load, unload, key events, mouse events), Document Object- Definition, Properties(color, anchor, forms, all, URL), Methods(getElementById, getElementsByTagName, getElementByName, getElementsByClassName, open, close, write, writeln), Window Object- Introduction, Properties(closed, frames, innnerHeight, innerWidth, length, location, name), Methods(alert, prompt, confirm, print ,find, home, open, close,resizeBy, resizeTo). | 12 L |

# Semester III – Practicals

| DSC- 1006C (Lab-I) | Practicals of DSC 1006C (Credits: 2, Pract/Week: 4) |
|---|---|
| | 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. Write a shell script to compare two files and if found equal asks the user to delete the duplicate file. |
| | 12. Write a shell script to find the sum of digits of a given number. |
| | 13. Write a shell script to merge the contents of three files, sort the contents and then display them page by page. |
| | 14. Write a shell script to find the LCD (least common divisor) of two numbers. |
| | 15. Write a shell script to perform the tasks of basic calculator. |
| | 16. Write a data entry script for student database with all validations. |
| | 17. Write a menu driven shell script to display today's date, present working directory, names of user, directory listing. |
| | 18. Write a shell script to check whether the number is Armstrong or not. |
| | 19. Write a shell script to check whether the file have all the permissions or not. |
| | 20. Program to show the pyramid of special character "*". |

# Semester IV - Practicals

| DSC1006D (LAB-II) | Practicals of DSC1006D (Credits: 2, Pract/Week: 4) |
|---|---|
| | 1. Program to demonstrate singly linked list (for insert, delete, display).<br>2. Program to demonstrate doubly linked list (insert, delete, display).<br>3. Program to demonstrate application of linked list- addition of polynomials.<br>4. Program to implement stack.<br>5. Program to demonstrate expression conversion of stack.<br>6. Program to demonstrate recursion using stack.<br>7. Program to implement of Circular Queue.<br>8. Program to demonstrate simple class and instance of a class.<br>9. Program to demonstrate use of class methods.<br>10. Program to demonstrate method overloading.<br>11. Program to demonstrate default constructor.<br>12. Program to demonstrate parameterized constructor.<br>13. Program to demonstrate constructor overloading.<br>14. Program to demonstrate destructor.<br>15. Program to demonstrate visibility modes.<br>16. Program to implement single inheritance.<br>17. Program to implement multiple inheritance.<br>18. Program to implement multilevel inheritance.<br>19. Program to implement hybrid inheritance.<br>20. Program to demonstrate method overriding.<br>21. Program to demonstrate operator overloading using magic method. |

# Evaluation Scheme

**I. Internal Exam - Marks – 20 Marks for each paper**
**(i) Test – 10 Marks**
10 marks Test – Duration 20 mins
It will be conducted either using any open source learning management system such as Moodle (modular object-oriented dynamic learning environment) or a test based on an equivalent online course on the contents of the concerned course(subject) offered by or build using MOOC (Massive Open Online Course) platform.

**(ii) 10 Marks –** Active participation in routine class instructional deliveries, Overall conduct as a responsible student, Organizing co-curricular activities etc.

**II. External Exam - Marks –80 Marks for each paper**
1. Duration - 3 Hours.
2. Theory question paper pattern:-
All questions are compulsory.
Question Based on Marks
     **Section-I**
          Q.1 Unit I [10]
          Q.2 Unit II [15]
          Q.3 Unit III [15]
    **Section – II**
          Q.1 Unit I [10]
          Q.2 Unit II [15]
          Q.3 Unit III [15]

All questions shall be compulsory with internal choice within the questions. Each Question may be sub divided into sub questions as a, b, c, d & e, etc & the allocation of Marks depends on the weightage of the topic.

**III. Practical Exam – 100 (50+50) marks**
**(Certified Journal is compulsory for appearing for practical exam)**
Practicals based on DSC1006C – 40 marks + 5 marks (journal) + 5 marks (viva)
Practicals based on DSC1006D – 40 marks + 5 marks (journal) + 5 marks (viva)