Annexure A
Detailed Syllabus – Discipline Specific Core

DISCIPLINE SPECIFIC CORE COURSE – DSC-07: DATA STRUCTURES

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Data Structures	4	3	0	1	Class XII pass with Mathematics	DSC-01

Learning Objectives:

1. *To introduce the fundamentals of data structures*
2. *To get familiar with programming*

Learning Outcomes:

1. *Develop the ability to use basic data structures like array, stacks, queues, lists, trees and hash tables to solve problems.*
2. *Use well-organized data structures in solving various problems.*
3. *Differentiate the usage of various structures in problem solutions.*
4. *Implement algorithms to solve problems using appropriate data structures.*

Unit I **(5 hours)**

Arrays: Single and multi-dimensional arrays, analysis of insert, delete and search operations in arrays (both linear search and binary search), implementing sparse matrices, applications of arrays to sorting: selection sort, insertion sort, bubble sort, comparison of sorting techniques via empirical studies.

Unit II **(5 hours)**

Linked Lists: Singly- linked, doubly-linked and circular lists, analysis of insert, delete and search operations in all the three types, implementing sparse matrices.

Unit III **(10 hours)**

Queues: Array and linked representation of queue, de-queue, comparison of the operations on queues in the two representations. Applications of queues.

Unit IV **(15 hours)**

Stacks: Array and linked representation of stacks, comparison of the operations on stacks in the two representations, implementing multiple stacks in an array; applications of stacks: prefix, infix and postfix expressions, utility and conversion of these expressions from one to another;

applications of stacks to recursion: developing recursive solutions to simple problems, advantages and limitations of recursion.

Unit V **(10 hours)**

Trees and Heaps: Introduction to tree as a data structure; binary trees, binary search trees, analysis of insert, delete, search operations, recursive and iterative traversals in binary search trees. Height-balanced trees (AVL), B trees, analysis of insert, delete, search operations on AVL and B trees. Introduction to heap as a data structure. Analysis of insert, extract-min/max and delete-min/ max operations, applications to priority queues.

Hash Tables: Introduction to hashing, hash tables and hashing functions -insertion, resolving collision by open addressing, deletion, searching and their analysis, properties of a goodhash function.

References

1. *Michael T. Goodrich, Roberto Tamassia and Michael H. Goldwasser (2013), Data Structures and Algorithms in Python, Wiley.*

2. *Rance D. Necaise, Data Structures and Algorithms Using Python, John Wiley & Sons, Inc.*

3. *Introduction to Algorithms, by Cormen, Leiserson, Rivest, and Stein, MIT Press, Third Edition, 2009.*

***List of Practical* (30 hours)**

A practical implementation of various data structure such as Array, Queues, Stacks, Linked List and Trees.

DISCIPLINE SPECIFIC CORE COURSE – DSC-08: Web Design and Development

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Web design and development	4	3	0	1	Class XII	NIL

Learning objectives:

1. To introduce the fundamentals of Internet, and the principles of web design.
2. To construct basic websites using HTML and Cascading Style Sheets.
3. To build dynamic web pages with validation using Java Script objects and by applying different event handling mechanisms.
4. To develop modern interactive web applications using PHP, XML and MySQL

Learning Outcomes:

1. *Structure and implement HTML/CSS.*
2. *Implement basic JavaScript.*
3. *Learn server side scripting language PHP and integration with database using MYSQL.*

UNIT-I **(5 Hours)**

Introduction to HTML & CSS:

HTML Basics, HTML Responsive, HTML Entities, HTML Forms, HTML5 Canvas, HTML5 SVG, HTML5 Data Storage, HTML5 Audio and Video , CSS Introduction, CSS Syntax, CSS Text, CSS Backgrounds, CSS Fonts, CSS Links, CSS Lists, CSS Tables, CSS Box Model, CSS Margins, Dimensions, Display, CSS Navigation Bar, CSS Attribute Selectors, CSS Rounded Corners, CSS Border Images, CSS Backgrounds, CSS Colors, CSS Animations.

UNIT-II **(10 Hours)**

Introduction to JavaScript:

JavaScript Introduction, JavaScript Output, JavaScript Variables, JavaScript Operators, JavaScript Arithmetic, JavaScript Data Types, JavaScript Assignment, JavaScript Functions, JavaScript Objects, JavaScript Scope, JavaScript Events, JavaScript Strings and String Methods, JavaScript Numbers and Number Methods, JavaScript Math, JavaScript Dates: Formats and Methods, JavaScript Booleans , JavaScript Comparisons, JavaScript Conditions, JavaScript Switch, JavaScript Loops, JavaScript Break, JavaScript Type, JavaScript Forms (API and Validation), JavaScript Objects, JavaScript Functions, JavaScript DOM, JavaScript Browser BOM, JavaScript Frameworks

UNIT-III **(15 Hours)**

Introduction to Bootstrap: Bootstrap Introduction, Bootstrap Components, Bootstrap Plugins, Bootstrap Grids, Bootstrap JS , PHP Introduction-Installing PHP, PHP Syntax, PHP Variables, PHP Data Types, PHP Strings, PHP Constants, PHP Operators, PHP Programming Loops, PHP Functions, PHP Arrays, PHP Super-global, PHP Forms and XML- PHP Form Handling, PHP Form Validation (Server side).

UNIT-IV **(15 Hours)**

PHP with MySQL: PHP MySQL Database, PHP Connecting to Database, PHP Creating Records, PHP Selecting Records, PHP Deleting Records, PHP Updating Records, PHP Limit Data, PHP Insert Multiple.

References

1. Learning PHP, MySQL & JavaScript: With JQuery, CSS & HTML5 by Robin Nixon , O'Reilly Media, Inc.

2. PHP and MySQL for Dynamic Web Sites: Visual QuickPro Guide by Larry Ullman , Fifth Edition.

List of Practicals (30 hours)

1. Design a home page which displays information about your college department using headings, HTML entities and paragraphs.
2. Implement different types of list tags, hyperlinks, marquee tag and HTML formatting tags in the college department homepage.
3. Create a web page having two frames, Frame 1 containing links and another with contents of the link. When a link is clicked appropriate contents should be displayed on Frame 2. Also, insert an iframe in the same page.
4. Design your course timetable and display it in tabular format.
5. Design an admission form for any course in your college with text, password fields, drop-down list, check-boxes, and radio buttons, submit and reset button etc. with proper CSS formatting.
6. Create a website for online book stores with Home, Login, Catalogue, Registration page with links to all these pages in a menu on top of every page. Embed heading, paragraph, images, video, .iframe, form controls, table, and list in this website. Use both Internal and external CSS in this.
7. Write a JavaScript program to display the current day and time.
8. Write a JavaScript program to
   a) Remove a character at the specified position of a given string and return the new string.
   b) Change the case of a string. (I.e. upper case to lower case and vice-versa).
9. Write a JavaScript program to compute the sum of elements of a given array of integers.
10. Develop and demonstrate a HTML file that includes JavaScript script for taking full name in a text field and display first, middle, last name *in 3 different labels. Middle and last name may be optional, thus messages like "NA" should be displayed in corresponding labels. If input contains 2 words, then they should be considered as first and last names.
11. Design HTML form for keeping student record, apply JavaScript validation for restriction of mandatory fields, numeric field, email-address field, specific value in a field etc.
12. Write a JavaScript code that displays text "Bigger Text" with increasing font size in the interval of 10ms in red color, when the font size reaches 50 pt. it displays "Smaller Text" in green color. Then the font size should decrease to 5pt and then stop.
13. Write a PHP script that removes the whitespaces from a string.
14. Create a login page having user name and password. On clicking submit, a welcome message should be displayed if the user is already registered (i.e.name is present in the database) otherwise error message should be displayed.
15. Create a simple 'birthday countdown' script, the script will count the number of days between current day and birth day.

DISCIPLINE SPECIFIC CORE COURSE – DSC-09: Operating Systems

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Operating Systems	4	3	0	1	Class XII	NIL

Learning objectives:

1. *Learn fundamental operating system abstractions such as processes, threads, files, semaphores, IPC abstractions, shared memory regions, etc.,*
2. *Learn how the operating system abstractions can be used in the development of application programs, or to build higher level abstractions,*
3. *Learn how the operating system abstractions can be implemented,*
4. *Learn the principles of concurrency and synchronization, and apply them to write correct concurrent programs/software,*
5. *Learn basic resource management techniques (scheduling, time management, space management) and principles and how they can be implemented. These also include issues of performance and fairness objectives, avoiding deadlocks, as well as security and protection.*

Learning Outcomes:

1. *Understand the need of an Operating System & Define Multiprogramming and multithreading concepts.*
2. *Implement Process Synchronization service (Critical Section, Semaphores), CPU scheduling service with various algorithms.*
3. *Learn Main memory Management (Paging, Segmentation) algorithms, Handling of Deadlocks*
4. *Identify and appreciate the File systems Services, Disk Scheduling service*

UNIT - I **(5 hours)**

Introduction: Operating Systems (OS) definition and its purpose, Multi-programmed and Time Sharing Systems, OS Structure, OS Operations: Dual and Multi-mode, OS as resource manager.

UNIT – II **(5 hours)**

Operating System Structures: OS Services, System Calls: Process Control, File Management, Device Management, and Information Maintenance, Inter-process Communication, and Protection, System programs, OS structure- Simple, Layered, Microkernel, and Modular.

UNIT - III **(10 hours)**

Process Management : Process Concept, States, Process Control Block, Process Scheduling, Schedulers, Context Switch, Operation on processes, Threads, Multicore Programming, Multithreading Models, Threads, Process Scheduling Algorithms: First Come First Served, Shortest-Job-First, Priority & Round-Robin, Process Synchronization: The critical-section problem and Peterson’s Solution, Deadlock characterization, Deadlock handling.

UNIT – IV **(10 hours)**

Memory Management: Physical and Logical address space, Swapping, Contiguous memory allocation strategies - fixed and variable partitions, Segmentation, Paging.

Virtual Memory Management: Demand Paging and Page Replacement algorithms: FIFO Page Replacement, Optimal Page replacement, and LRU page replacement.

UNIT – V **(15 hours)**

File System: File Concepts, File Attributes, File Access Methods, Directory Structure: Single-Level, Two-Level, Tree-Structured, and Acyclic-Graph Directories.

Mass Storage Structure: Magnetic Disks, Solid-State Disks, And Magnetic Tapes, Disk Scheduling algorithms: FCFS, SSTF, SCAN, C-SCAN, LOOK, and C-Look Scheduling.

References

1. *Silberschatz, A., Galvin, P. B., Gagne G. Operating System Concepts, 9th edition, John Wiley Publications, 2016*
2. *Dhamdhere, D. M. Operating Systems: A Concept-based Approach. 2nd edition, Tata McGraw-Hill Education, 2017*
3. *Kernighan, B. W., Rob Pike, R. The UNIX Programming Environment. Englewood Cliffs, NJ: Prentice-Hall, 1984*
4. *Stallings, W. Operating Systems: Internals and Design Principles. 9th edition, Pearson Education, 2018*
5. *Tanenbaum, A. S. Modern Operating Systems. 3rd edition, Pearson Education, 2007*

List of Practicals :(30 hours)

1. Write a program (using fork() and/or exec() commands) where parent and child execute:
   a) Same program, same code.
   b) Same program, different code.
   c) Before terminating, the parent waits for the child to finish its task.
2. Write a program to report behavior of Linux kernel including kernel version, CPU type and model. (CPU information)
3. Write a program to report behavior of Linux kernel including information on configured memory, amount of free and used memory. (Memory information)
4. Write a program to print file details including owner access permissions, file access time, where file name is given as argument.
5. Write a program to copy files using system calls.
6. Write a program to implement FCFS scheduling algorithm.
7. Write a program to implement Optimal scheduling algorithm.
8. Write a program to implement the SJF scheduling algorithm.
9. Write a program to implement a non-preemptive priority based scheduling algorithm.
10. Write a program to implement SRJF scheduling algorithm.
11. Write a program to calculate sum of n numbers using thread library.
12. Write a program to implement first-fit, best-fit and worst-fit allocation strategies.