

4. Implement circular linked lists.
5. Implement stack data structure and its operations using arrays.
6. Implement stack data structure and its operations using linked lists.
7. Convert Prefix expression to Infix and Postfix expressions, and evaluate.
8. Implement queue data structure and its operations using arrays.
9. Implement queue data structure and its operations using linked lists.
10. Implement Binary Trees and its traversals.

### GENERIC ELECTIVES (GE-4b): Introduction to Web Programming

#### 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
		Lecture	Tutorial	Practical/ Practice		
<b>GE4b: Introduction to Programming</b>	<b>4</b>	<b>3</b>	<b>0</b>	<b>1</b>	Pass in Class XII	NIL

#### Learning Objectives

The course aims at introducing the basic concepts and techniques of client side web programming. The student shall be able to develop simple websites using HTML, CSS and Javascript.

#### Learning outcomes

On successful completion of the course, students will be able to :

- Build websites using the elements of HTML.
- Build dynamic websites using the client side programming techniques with CSS, Javascript and jQuery.
- Learn to validate client-side data.

#### SYLLABUS OF GE4b

##### Unit 1 (5 hours)

**Introduction:** Introduction to Internet and web design. Basic concepts of web architecture.

##### Unit 2 (12 hours)

**HTML:** Introduction to hypertext mark-up language (html), creating web pages, lists,

hyperlinks, tables, web forms, inserting images, frames.

### **Unit 3 (8 hours)**

Cascading style sheet (CSS): Concept of CSS, creating style sheet, Importing style sheets, CSS properties, CSS styling (background, text format, controlling fonts), CSS rules, Style Types, CSS Selectors, CSS cascade, working with block elements and objects, working with lists and tables, CSS id and class, box model (introduction, border properties, padding properties, margin properties).

### **Unit 4 (10 hours)**

Javascript: Document object model, data types and variables, functions, methods and events, controlling program flow, JavaScript object model, built-in objects and operators, validations.

### **Unit 5 (10 hours)**

jQuery and JSON: Introduction to jQuery, syntax, selectors, events. JSON file format for storing and transporting data.

### **Essential/recommended readings**

1. Nixon, R. *Learning PHP, MySQL & JavaScript with jQuery, CSS and HTML5*, O'Reilly, 2018.
2. Powell, T.A. *HTML & CSS: The Complete Reference, 5<sup>th</sup> edition*, Tata McGrawHill, 2010.
3. Duckett, J. *JavaScript and JQuery: Interactive Front-End Web Development*, Wiley, 2014.

### **Additional References**

1. Minnick, J. *Web Design with HTML5 and CSS3*, 8<sup>th</sup> edition, Cengage Learning, 2015.
2. Boehm, A., & Ruvalcaba, Z. *Munarch's HTML5 and CCS*, 4<sup>th</sup> edition, Mike Murach & Associates, 2018.
3. J. A. Ramalho *Learn Advanced HTML 4.0 with DHTML*, BPB Publications, 2007.
4. Ivan Bayross *Web Enabled Commercial Application Development Using Html, Dhtml, Javascript, Perl CGI*, BPB Publications, 2009.

### **Suggested Practical List (If any): (30 Hours)**

#### **Practical exercises such as**

#### **HTML**

1. Create an HTML document with following formatting – Bold, Italics, Underline, Colors, Headings, Title, Font and Font Width, Background, Paragraph, Line Brakes, Horizontal Line, Blinking text as well as marquee text.
2. Create an HTML document with Ordered and Unordered lists, Inserting Images, Internal and External linking
3. Create an HTML displaying this semester's time table.
4. Create a website with horizontal and vertical frames. Top horizontal frame showing

your college's name and logo. Bottom horizontal frame split into two vertical frames. The left frame with hyperlinks to pages related to faculty, courses, student activities, etc. The right frame showing corresponding pages based on the link clicked on the left frame.

5. Create a student registration form using HTML which has the following controls:
  - I. Text Box
  - II. Dropdown box
  - III. Option/radio buttons
  - IV. Check boxes
  - V. Reset and Submit button

## CSS

Create a webpage for your department with drop down navigation menu for faculty, courses, activities, etc.. Implement the webpage using styles, rules, selectors, ID, class.

## Javascript

1. Create event driven programs for the following:
  - e. Enter a number and on click of a button print its multiplication table.
  - f. Print the largest of three numbers entered by the user.
  - g. Find the factorial of a number entered by the user.
  - h. Enter a list of positive numbers using the prompt terminated by a zero. Find the sum and average of these numbers.
2. Create a student registration form using text, radio button, check box, drop down box, text field and all other required HTML elements. Customise the CSS and javascript to input and validate all data. Create functions to perform validation of each element, example:
  - d. Roll number is a 7-digit numeric value
  - e. Name should be an alphabetical value(String)
  - f. Non-empty and valid fields like DOB

## jQuery and JSON

1. Change text color and contents using button click events using jQuery
2. Select elements using ID, class, elements name, attribute name
3. Run code on click events in jQuery

(Computer Science Courses for Undergraduate Programme of study with **Computer Science** discipline as one of the **three** Core Disciplines)

**DISCIPLINE SPECIFIC CORE COURSE (DSC04): 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		
<b>DSC04: Operating Systems</b>	<b>4</b>	<b>3</b>	<b>0</b>	<b>1</b>	Pass in Class XII	DSC 01 Programming using Python/ A course in C/C++/Python at plus 2 level.

**Learning Objectives**

This course introduces the students to Operating Systems and its importance in computer systems. The focus is to explain the common services provided by an operating system like process management, memory (primary, secondary & virtual) management, I/O management, file management. The course talks about the various functional components of the operating and their design.

**Learning outcomes**

On successful completion of the course, students will be able to

- gain knowledge of different concepts of the operating System and its components.
- learn about shell scripts and would be able to use the system in an efficient manner.

**SYLLABUS OF DSC04**

**Unit 1 (4 hours)**

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

**Unit 2 (10 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 3 (9 hours)**

**Process Management:** Process Concept, States. Process Control Block, Context Switch, Process scheduling, Schedulers, Overview of threads and Scheduling Algorithms: First Come First Served, Shortest-Job-First, Priority & Round-Robin.

#### **Unit 4 (9 hours)**

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

#### **Unit 5 (8 hours)**

**File and Input / Output Device Management:** File Concepts, File Attributes, File Access Methods, Directory Structure: Single-Level, Two-Level, Tree-Structured, and Acyclic-Graph Directories, Magnetic Disks, Solid-State Disks, Magnetic Tapes.

#### **Unit 6 (5 hours)**

**Shell Scripting:** Shell variables, parameter passing conditional statements, iterative statements, writing and executing shell scripts, utility programs (cut, paste, grep, echo, pipe, filter etc.)

#### **Essential/recommended readings**

1. Galvin, S. P. B., Gagne, G., *Operating System Concepts*, 9<sup>th</sup> edition, John Wiley Publications, 2016.
2. G. Nutt, *Operating Systems*, Pearson, 2009
3. Das, S., *Unix: Concepts and Applications*, 4<sup>th</sup> edition, TMH, 2009.

#### **Additional References**

1. Dhamdhere, D. M., *Operating Systems: A Concept-based Approach*, 2<sup>nd</sup> edition, Tata McGraw-Hill Education, 2017.
2. Kernighan, B. W., Pike, R., *The Unix Programming Environment*, Englewood Cliffs, NJ: Prentice-Hall, 1984.
3. Stallings, W., *Operating Systems: Internals and Design Principles*, 9<sup>th</sup> edition, Pearson Education, 2018.
4. Tanenbaum, A. S., *Modern Operating Systems*. 3<sup>rd</sup> edition, Pearson Education, 2007.

#### **Suggested Practical List (If any): (30 Hours)**

##### **Practical exercises such as**

1. Execute various LINUX commands for:
  - i. Information Maintenance: wc, clear, cal, who, date, pwd
  - ii. File Management: cat, cp, rm, mv, cmp, comm, diff, find, grep
  - iii. Directory Management : cd, mkdir, rmdir, ls

2. Execute various LINUX commands for:
  - i. Process Control: fork, getpid, ps
  - ii. Communication: Input-output redirection, Pipe
  - iii. Protection Management: chmod, chown, chgrp
3. Write a program(using fork() and/or exec() commands) where parent and child execute:
  - i. same program, same code.
  - ii. same program, different code.
  - iii. before terminating, the parent waits for the child to finish its task.
4. Write a program to calculate sum of n numbers using Pthreads.
5. Write a program to generate a Fibonacci Series of numbers using Pthreads.
6. Write a program to implement best-fit and worst-fit allocation strategies
7. Write a program to copy files using system calls and using pthreads and compare timings.
8. Write a program to implement FCFS scheduling algorithm.
9. Write a program to implement SJF scheduling algorithm.
10. Write a program to implement non-preemptive priority based scheduling algorithm.