

## COMMON POOL OF GENERIC ELECTIVES (GE) COURSES

(For all the Generic Elective courses offered by your department, please put it in the format provided below)

### GENERIC ELECTIVES (GE-4a): Data Structures using C++

#### 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>GE4a Data Structures using C++</b>	<b>4</b>	<b>3</b>	<b>0</b>	<b>1</b>	Pass in Class XII	NIL

#### Learning Objectives

The course aims at developing the ability to use basic data structures like arrays, stacks, queues, lists, trees to solve problems. C++ is chosen as the language to understand implementation of these data structures.

#### Learning outcomes

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

- Compare two functions for their rates of growth.
- Understand abstract specification of data-structures and their implementation.
- Compute time and space complexity of operations on a data-structure.
- Identify the appropriate data structure(s) for a given application and understand the trade-offs involved in terms of time and space complexity.
- Apply recursive techniques to solve problems.

#### SYLLABUS OF GE-4a

##### Unit 1 (7 hours)

**Growth of Functions, Recurrence Relations:** Functions used in analysis, asymptotic notations, asymptotic analysis, solving recurrences using recursion tree, Master Theorem.

##### Unit 2 (14 hours)

**Arrays, Linked Lists, Stacks, Queues, Deques:** Arrays: array operations, applications, sorting, two-dimensional arrays, dynamic allocation of arrays; Linked Lists: singly linked lists, doubly linked lists, circularly linked lists, Stacks: stack as an ADT, implementing stacks using arrays, implementing stacks using linked lists, applications of stacks; Queues: queue as an ADT, implementing queues using arrays, implementing queues using linked lists, double-ended queue as an ADT. Time complexity analysis of operations on all data structures.

### **Unit 3 (4 hours)**

**Sorting:** Insertion Sort, Count Sort and their complexity analysis.

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

**Recursion:** Recursive functions, linear recursion, binary recursion.

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

**Trees, Binary Trees:** Trees: definition and properties, binary trees: definition and properties, traversal of binary trees and their time complexity analysis. Binary Search Trees: insert, delete (by copying), search operations, time complexity analysis of these operation.

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

**Binary Heap, Priority Queue:** Binary Heaps: motivation and introduction, application of heaps - Priority Queues.

### **Essential/recommended readings**

1. Goodrich, M.T, Tamassia, R., & Mount, D., *Data Structures and Algorithms Analysis in C++*, 2<sup>nd</sup> edition, Wiley, 2011.
2. Cormen, T.H., Leiserson, C.E., Rivest, R. L., Stein C., *Introduction to Algorithms*. 4<sup>th</sup> edition. Prentice Hall of India. 2022.
3. Drozdek, A., *Data Structures and Algorithms in C++*, 4<sup>th</sup> edition, Cengage Learning, 2012.

### **Additional References**

1. Sahni, S. *Data Structures, Algorithms and applications in C++*, 2<sup>nd</sup> edition. Universities Press, 2011.
2. Langsam Y., Augenstein, M. J., & Tanenbaum, A. M. *Data Structures Using C and C++*, 2<sup>nd</sup> edition, Pearson, 2009.

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

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

1. Perform matrix addition and multiplication.
2. Implement following recursive functions:
  - a. Factorial of a number
  - b. N<sup>th</sup> fibonacci number
  - c. Power function:  $x^y$
3. Implement singly linked lists.
3. Implement doubly linked lists.

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'Rielly, 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