

**SYLLABUS STRUCTURE (A.Y.2020-21)**

<b>Class :F.Y.B.Sc. (Computer Science)</b>			
<b>Semester I</b>		<b>Semester II</b>	
CSCO 1101	Basic Programming using C	CSCO 1201	Advanced Programming using C
CSCO1102	DBMS-I	CSCO 1202	DBMS-II
<b>Practicals</b>			
CSCO 1203	Lab Course I : Basics of C &Advanced C		
CSCO 1204	Lab Course II : DBMS I & II		

<b>Class :S.Y.B.Sc. (Computer Science)</b>			
<b>Semester I</b>		<b>Semester II</b>	
CSCO 2101	Data Structures using C	CSCO 2201	Object Oriented Concepts using Java
CSCO2102	Introduction to Web Technology	CSCO2202	Software Engineering
<b>Practicals</b>			
CSCO2203	Lab Course I based On CSCO2101 & 2201		
CSCO2204	Lab Course II: based On CSCO2102 & CSCO2202 with Mini Project		

**- S.Y.B.Sc. (Computer Science) Credit Structure -**

Subject	Semester I		Semester II		Paper III Practical	Paper IV Practical	Total
	Paper I	Paper II	Paper I	Paper II			
Computer Science	3	3	3	3	4	<b>Grade</b>	16
Mathematics	3	3	3	3	4	--	16
Electronics	3	3	3	3	4	--	16
English	3	--	3	--	--	--	6
Evs.	-----		----		--	--	4
Certificate Courses	2		2		--	--	4

**Total Number of Credits = 54 (Core) + 4 (EVS) + 4 (Certificate Courses) = 62**

**1 Theory Credit = 16 Lectures**

**SYLLABUS (CBCS) FOR S. Y. B. Sc. (Computer Science)**  
**(w.e.f from June, 2020)**  
**Academic Year 2020-2021**

**Class: S.Y. B. Sc.(Computer Science) (Semester- I)**

**Subject:** Computer Science

**Paper Code :** CSCO 2101

**Title of Paper:** Data Structure Using C

**Paper:** I

**Credit:** 3 (4 Lectures/Week)

**No. of lectures:** 48

**Prerequisites:**

- Basic knowledge of algorithms and problem solving.
- Knowledge of C Programming Language.

**Objective:**

1. To learn the systematic way of solving problem
2. To understand the different methods of organizing large amount of data
3. To efficiently implement the different data structures
4. To efficiently implement solutions for specific problems

**Learning Outcomes:** On completion of the course, student will be able to

1. Use well-organized data structures in solving various problems.
2. Differentiate the usage of various structures in problem solution.
3. Implementing algorithms to solve problems using appropriate data structures.

**1. Introduction to data structures**

[2]

- 1.1 Concept
- 1.2 Data type, Data object, ADT
  - 1.2.1 Data Type
  - 1.2.2 Data Object
  - 1.2.3 ADT -Definition, Operation, examples on rational number
- 1.3 Need of Data Structure
- 1.4 Types of Data Structure

**2. Algorithm analysis**

[3]

- 2.1 Algorithm – definition, characteristics
- 2.2 Space complexity, time complexity
- 2.3 Asymptotic notation (Big O, Omega  $\Omega$ , Theta Notation  $\Theta$ )

**3. Linear data structures**

[6]

- 3.1 Introduction to Arrays - array representation
- 3.2 Sorting algorithms with efficiency - Bubble sort, Insertion sort, Merge sort, Quick Sort
- 3.3 Searching techniques –Linear Search, Binary search

**4. Linked List**

[8]

- 4.1 Introduction to Linked List
- 4.2 Implementation of Linked List – Static & Dynamic representation,
- 4.3 Types of Linked List
- 4.4 Operations on Linked List - create, display, insert, delete, reverse, search, sort, concatenate &merge
- 4.5 Applications of Linked List – polynomial manipulation
- 4.6 Generalized linked list – Concept and Representation

<b>5. Stacks</b>	<b>[6]</b>
5.1 Introduction	
5.2 Representation- Static & Dynamic	
5.3 Operations – Create , Init , Push , Pop & Display	
5.4 Application - infix to postfix, infix to prefix, Evaluation of Expression	
5.5 Simulating recursion using stack	
<b>6. Queues</b>	<b>[4]</b>
6.1 Introduction	
6.2 Representation - Static & Dynamic	
6.3 Operations – Create , Init , Insert , Remove & Display	
6.4 Circular queue, priority queue (with implementation)	
6.5 Concept of doubly ended queue (Dequeue)	
<b>7. Trees</b>	<b>[12]</b>
7.1 Concept & Terminologies	
7.2 Binary tree, binary search tree	
7.3 Representation – Static and Dynamic	
7.4 Operations on BST & Heap Tree – create, Insert, delete, traversals (preorder, inorder, postorder), counting leaf, non-leaf & total nodes , non recursive inorder traversal	
7.5 Application - Heap sort	
7.6 Height balanced tree- AVL trees- Rotations, AVL tree examples.	
<b>8. Graph</b>	<b>[7]</b>
8.1 Concept & terminologies	
8.2 Graph Representation – Adjacency matrix, adjacency list, inverse Adjacency list, adjacency multi list, orthogonal list	
8.3 Traversals – BFS and DFS	
8.4 Spanning Tree	
8.5 Applications – AOV network – topological sort, AOE network – critical path	

**References:**

1. Fundamentals of Data Structures ---- By Horowitz Sahani (Galgotia)
2. Data Structures using C and C++ --- By Yedidyah Langsam, Aaron M. Tenenbaum, Moshe J. Augenstein
3. Introduction to Data Structures using C---By Ashok Kamthane
4. Data Structures using C --- Bandopadhyay & Dey (Pearson)
5. Data Structures using C ---By Srivastav

**Class: S.Y. B. Sc.(Computer Science) (Semester- I)**

**Subject:** Computer Science

**Paper Code :** CSCO 2102

**Title of Paper:** Introduction to Web Technology

**Paper:** II

**Credit:** 3 (4 Lectures/Week)

**No. of lectures:** 48

**OBJECTIVES:** Students successfully completing this course will be able:

1. To understand different Web technologies.
2. To keep pace with the rapidly changing landscape of web application development.
3. To Design dynamic and interactive web pages.

<b>Title and Contents</b>		<b>No. of Lectures</b>
<b>Unit 1</b>	<b>Basics of Web Design</b> 1.1 History of the Internet 1.2 World Wide Web Consortium (W3C) 1.3 Personal, Distributed and Client/Server Computing 1.4 Key Software Trend: Object Technology 1.5 Browser Portability 1.6 Software Technologies 1.7 Client Server Architecture	04
<b>Unit 2</b>	<b>Introduction to HTML5</b> 2.1 Difference between HTML & HTML5 2.2 HTML Document and Basic Structure 2.3 Working with HTML Text, Heading, Paragraph, formatting 2.4 HTML Color, Link, Image 2.5 HTML Lists, Tables and Frames 2.6 HTML Forms Block, Layout	16
<b>Unit 3</b>	<b>Specific Elements of HTML5</b> 3.1 Header & Footer 3.2 Navigation Section 3.3 Article & Aside 3.4 The Meter Element 3.5 Working with Hyperlinks and Multimedia 3.6 Working with Forms and controls.	8
<b>Unit 4</b>	<b>The Basic of CSS</b> 4.1 Intro. Concepts of CSS & Creating of CSS, Using Inline CSS,CSS Color 4.2 Using Internal CSS, Using ID's and Classes, Creating External CSS. 4.3 Linking to External CSS, Inefficient Selectors and Efficient Selectors. 4.4 HTML Elements State, the CSS Box Model, Fonts	12
<b>Unit 5</b>	<b>Main CSS3.0 Specific Properties</b> 5.1 Introduction to CSS 3.0 5.2 Alpha Color Space, Opacity 5.3 Box Shadow & Border Radius	6
<b>Unit 6</b>	<b>Emerging Trends in Web Technologies</b> 6.1 Introduction to – 6.1.1) CMS-Wordpress/Drupal/Joomla 6.1.2 ) jQuery 6.1.3) AngularJS 6.1.4) Bootstrap	2

*Note: Regular Practical assignment on HTML5 are 50% and one mini Project.*

**References:**

- 1) Html & CSS: The Complete Reference, Fifth Edition by Thomas A. Powell and published by McGraw Hill.
- 2) HTML 5 in simple steps by Kogent Learning Solutions Inc., Publisher Dreamtech Press
- 3) Head First HTML with CSS & XHTML Book by Elisabeth Freeman and Eric Freeman.
- 4) The Essential Guide to CSS and HTML Web Design Book by Craig Grannell.
- 5) Beginning XML by Joe Fawcett, Liam R.E. Quin & Danny Ayers  
Published by John Wiley & Sons, Inc.

**Class: S.Y. B. Sc.(Computer Science) (Semester- I)**

**Subject:** Computer Science

**Paper Code :** CSCO 2103

**Title of Paper:** Lab I based On Data structure

**Paper:** III

**Credit:** 2 (3 Hour Practical /batch/Week)

**No. of Practical's:** 12

**Prerequisites:**

- Basic knowledge of algorithms and problem solving.
- Knowledge of C Programming Language.
  
- Assignment 1 – Sorting Algorithms
  - Bubble Sort
  - Insertion Sort
  - Quick Sort
  - Merger Sort
  
- Assignment 2 - Recursive Sorting Algorithms
  - Quick sort,
  - Merge Sort
  
- Assignment 3 - Searching Method
  - Linear search,
  - Binary search
  
- Assignment 4 - Stack
  - Static Stack Implementation
  - Dynamic Stack Implementation
  
- Assignment 5 Queue
  - Static and Dynamic Implementation
  - Linear Queue,
  - Circular queue
  
- Assignment 6 Linked List –
  - Dynamic Implementation of Singly Linked List
  - Dynamic Implementation of Doubly Linked List
  - Dynamic Implementation of Circular Linked List.
  
- Assignment 7 Tree –
  - Binary Search Tree Traversal: Create, add, delete, and display nodes.
  
- Assignment 8 Graph –
  - Adjacency matrix to adjacency list conversion, in degree, out degree

**Class: S.Y. B. Sc.(Computer Science) (Semester- I)**

**Subject:** Computer Science

**Paper Code :** CSCO 2102

**Title of Paper:** Lab II: based On Web Technology

**Paper:** IV (Grade)

**Credit:** 2 (3 Hour Practical/Week/batch)

**No. of Practical:** 13

**OBJECTIVES:** Students successfully completing this course will be able:

1. To understand different Web technologies.
2. To keep pace with the rapidly changing landscape of web application development.
3. To Design dynamic and interactive web pages.

**Learning Outcome:**

1. To understand different Web technologies.
2. To keep pace with the rapidly changing landscape of web application development.
3. To Design dynamic and interactive web pages.

<b>Assignment on Web Technology Using (HTML5, CSS &amp; Java Script)</b>	
<b>Sr. No.</b>	<b>Assignment Name</b>
1	Be acquainted with elements, Tags and basic structure of HTML files.
2	Develop the concept of basic and advanced text formatting.
3	Practice the use of multimedia components in HTML documents.
4	Designing of webpage-Document Layout, Working with List, Working with Tables.
5	Practice Hyper linking, Designing of webpage-Working with Frames, Forms and Controls.
6	Prepare creating style sheet, CSS properties, Background, Text, Font and styling etc.
7	Working with List, HTML elements box, Positioning and Block properties in CSS.
8	Designing with cascading style sheet-Internal and External style sheet.
9	Working with CSS 3.0,Alpha Color Space, Opacity
10	Practice the use JavaScript Basics Programs Data Types, Control Structure
11	Develop the concept of basic and advance using JavaScript Functions
12	Practice the use JavaScript events (onClick, onMousemove and onMouseover events of button object)
13	Designing event driven JavaScript program and use Popup boxes
14	Working with JavaScript Objects