**S.Y.B.Sc. Computer Science Theory Paper I**
**Semester – 1**
**CS 211- DATA STRUCTURES USING 'C'**
**(Compulsory Course)**

**Total Lectures: 48**

**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

**Prerequisites**: Knowledge of C Programming Language

**1. Introduction to data structures [3]**

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 [2]**

2.1 Algorithm – definition, characteristics
2.2 Space complexity, time complexity
2.3 Asymptotic notation (Big O, Omega Ω)

**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

**5. Stacks [6]**

5.1 Introduction
5.2 Representation- Static & Dynamic
5.3 Operations
5.4 Application - infix to postfix,infix to prefix, postfix evaluation,
5.5 Simulating recursion using stack

**6. Queues [4]**

6.1 Introduction
6.2 Representation - Static & Dynamic
6.3 Operations
6.4 Circular queue, priority queue (with implementation)
6.5 Concept of doubly ended queue

**7. Trees [12]**

7.1 Concept & Terminologies
7.2 Binary tree, binary search tree
7.3 Representation – Static and Dynamic
7.4 Operations on BST – create, Insert, delete, traversals (preorder, inorder, postorder), counting leaf, non-leaf & total nodes , non recursiveinorder traversal
7.5 Application - Heap sort
7.6 Height balanced tree- AVL trees- Rotations, AVL tree examples.

**8. Graph [7]**

8.1 Concept & terminologies
8.2 Graph Representation – Adjacency matrix, adjacency list, inverse Adjacency list, adjacency multilist, orthogonal list
8.3 Traversals – BFS and DFS
8.4 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 YedidyahLangsam, 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 Srivastava BPB Publication.

**S.Y.B.Sc. Computer Science Theory Paper I**
**Semester II**

# CS 221 -Object Oriented Concepts using C++

**Total Lectures: 48**
**Objective:-**

1. Acquire an understanding of basic object oriented concepts and the issues involved in effective class design
2. Write C++ programs that use object oriented concepts such as information hiding, constructors, destructors, inheritance etc.

**Prerequisites:** Knowledge of C Programming Language

**1. Object oriented concepts [2]**

1.1 Object oriented concepts
1.2 Features, advantages and Applications of OOPS

**2. Introduction to C++ [6]**

2.1 Data types, new operators and keywords, using namespace concept
2.2 Simple C++ Program
2.3 Introduction to Reference variables
2.4 Usage of ‘this’ pointer
2.5 Classes and Objects
2.6 Access specifiers
2.7 Defining Data members and Member functions
2.8 Array of objects

**3. Function in C++ [8]**

3.1 Call by reference, Return by reference
3.2 Function overloading and default arguments
3.3 Inline function
3.4 Static class members
3.5 Friend Concept – Function, Class

**4. Constructors and destructor [4]**

4.1 Types of constructors
4.2 Memory allocation (new and delete)
4.3 Destructor

**5. Operator overloading [4]**

5.1 Overloading Unary and Binary operators
5.2 Overloading using friend function
5.3 Type casting and Type conversion

**6. Inheritance [8]**

6.1 Types of inheritance with examples
6.2 Constructors and destructor in derived classes
6.3 Virtual base classes, Virtual functions and Pure virtual function
6.4 Abstract base classes

**7. Managing Input and Output using C++ [4]**

7.1 Managing console I/O
7.2 C++ stream classes
7.3 Formatted and unformatted console I/O
7.4 Usage of manipulators

**8. Working with files [6]**

8.1 File operations – Text files, Binary files
8.2 File stream class and methods
8.3 File updation with random access
8.4 Overloading insertion and extraction operator

**9. Templates [4]**

9.1 Introduction to templates
9.2 Class templates, function templates and overloading of function templates
9.3 Templates with multiple parameters

**10. Exception Handling in C++ [2]**

10.1 try, catch and throw primitives

**Reference Books: -**

1. Object Oriented Programming with C++ by Robert Lafore
2. Object Oriented Programming with C++ by E. Balagurusamy
3. Object Oriented Modeling and Design by James Rumbough
4. The Complete Reference C++ by Herbert Schildt
5. Let us C++ by – YashwantKanitkar
6. Mastering C++ by Venugopal, T Ravishankar, RajkumarTHM Pub.
7.Trouble free C++ by HarimohanPande, ANE publication