

**NORTH MAHARASHTRA UNIVERSITY,
JALGAON**



**Syllabus for
S. Y. B. Sc.
(Semester Pattern)
Computer Science
(w. e. f. June 2016)**

SCIENCE FACULTY
North Maharashtra University, Jalgaon
S. Y. B. Sc.(Computer Science)
(Semester Pattern)
(w.e.f. June 2016)

Course	Title	Periods	Marks: Ext. Int.	
COMP 211	: Data Structure-I	60	60	40
COMP 212	: OOAD & Introduction to C++	60	60	40
COMP 213	: Practical Course		60	40
COMP 221	: Data Structure – II	60	60	40
COMP 222	: Programming in C++	60	60	40
COMP 223	: Practical Course		60	40

**NORTH MAHARASHTRA UNIVERSITY,
JALGAON**

Syllabus for

S. Y. B. Sc.

(Semester Pattern)

Computer Science

(w. e. f. June 2016)

SCIENCE FACULTY

North Maharashtra University, Jalgaon

S. Y. B. Sc.(Computer Science)

(Semester Pattern)

(w.e.f. June 2016)

Note :-

- 1. Each period is of 48 minutes duration.**
- 2. Each course is having weightage of four periods per week.**
- 3. Each practical course is having weightage of four periods per week.**
- 4. Question paper will be of 90 marks, students have to attempt 60 marks.**

Chairman B. O. S.

**Dean Sci. Faculty
North Maharashtra University, Jalgaon**

North Maharashtra University, Jalgaon
S. Y. B. Sc. (Computer Science)
w.e.f. June- 2016
Sem - I Paper - I

COMP 211 : Data Structure – I

Unit 1. Introduction to Data Structure & Algorithm Notations (L :12, M: 18)

- 1.1 Introduction to Data Structure,
- 1.2 Types of data structure 1. Primitive 2. Non Primitive 3. Linear 4. Non linear
- 1.3 Need of data structure
- 1.4 Algorithm Notations.
 - a. Format Convention
 - b. Name of Algorithm
 - c. Introductory Comment
 - d. Steps
 - e. Comments
- 1.5 Data Structure
 - a. Arrays
 - b. Dynamic Storage allocation
 - c. Functions
 - d. Procedures

Unit 2. Introduction to Algorithm analysis for Time and Space Requirement (L: 6, M:12)

- 2.1 Rate of Growth
- 2.2 Basic time analysis of an algorithm
- 2.3 Order Notation
- 2.4 More timing Analysis
- 2.5 Space analysis of an algorithm

Unit 3. Stacks (L: 12, M:15)

- 3.1 Definition and concept
- 3.2 Representations – static
- 3.3 Operations – push, pop, peep, change
- 3.4 Applications – infix to postfix & prefix, postfix evaluation, Recursion using stack
- 3.5 Concept of Multiple stacks

Unit 4. Queues (L: 12, M :15)

- 4.1 Definition and Concept
- 4.2 Representation – static
- 4.3 Operations- Insert, Delete
- 4.4 Circular queue : Concept, Operations – insert, delete
- 4.5 DeQue : Concept
- 4.6 Priority queues : Concept
- 4.7 Concept of Multiple Queues

Unit 5. Linked List**(L: 14, M: 21)**

5.1 Introduction to Linked list

5.2 Implementation of List – Dynamic representation.

5.3 Types of Linked List

a. Singly Linked list

b. Circular linked list

c. Doubly linked linear list

5.4 Operations on linked list – Insert, delete, search, copy, replace, reverse

5.5 Applications of linked list – polynomial manipulation

5.6 Generalized list – Concept & representation

Unit 6 File Structure**(L :4 , M :9)**

6.1 Introduction to file

6.2 Sequential File concept

6.3 Index Sequential File concept

6.4 Direct file concept

References :

1. Jean-Paul Trembley, Paul. G. Soresan, An introduction to data structures with applications, Mc-Graw Hill International Editions
2. Horowitz, Sahani, Data Structures : Galgotia publication
3. Aho, Hopcroft, Ullman, Data Structures and Algorithms
4. Niklaus wirth, Algorithms- Data Structures = Programs.
5. Tannenbaum, Data Structures using C and C++; PHI.
6. Thoms Horbron, -File systems – Structures and algorithms; PHI. I
7. Donald Knuth, - Art of Computer Programming Vol. I;

North Maharashtra University, Jalgaon
S. Y. B. Sc. (Computer Science)
w.e.f. June- 2016
Sem – II Paper – I
Comp-221: Data Structure – II

Unit 1. Tree **(L: 14, M :21)**

- 1.1 Definition and Concept
- 1.2 Binary tree
- 1.3 Storage representation and Manipulation of Binary trees
 - a.Sequential Storage representation of Binary Tree
 - b.Linked Storage representation of Binary Tree
 - c.Threaded storage representation of Binary Tree
- 1.4 Operations on Binary tree - Traversing
- 1.5 Operations & Algorithms on BST – Create, Insert, Delete
- 1.6 Concept : AVL tree. B- tree

Unit 2. Graph **(L: 12, M:21)**

- 2.1 Definition and Concept
- 2.2 Matrix representation of graph
- 2.3 List Structures
- 2.4 Multi list representation of Graph
- 2.5 Traversal of graph : Breadth First Search and Depth First search
- 2.6 Applications of graph

Unit 3. Sorting **(L :18, M :24)**

- 3.1 Introduction
- 3.2 Sorting Techniques :
 - 3.2.1 Selection Sort
 - 3.2.2 Insertion sort
 - 3.2.3 Bubble Sort
 - 3.2.4 Merge Sort
 - 3.2.5 Tree Sort
 - 3.2.6 Quick Sort
 - 3.2.7 Radix Sort
 - 3.2.8 Sorting Method Comparison on Time and space Complexity attribute

Unit 4. Searching Techniques **(L:10, M:15)**

- 4.1 Sequential Searching
- 4.2 Binary searching
- 4.3 Hash Table Method
 - 4.3.1 Introduction
 - 4.3.2 Hashing Function
 - 4.3.3 Collision Resolution Technique

Unit 5. Introduction to Algorithm Design Techniques **(L:6, M:9)**

- 5.1 Divide and Conquer technique
- 5.2 Greedy techniques
- 5.3 Dynamic Programming

5.4 Back Tracking.

References :

1. Jean-Paul Trembley, Paul. G. Soresan, An introduction to data structures with applications, Mc-Graw Hill International Editions
2. Horowitz, Sahani, Data Structures; Galgotia publication
3. Aho, Hopcroft, Ulman, Data Structures and Algorithms
4. Niklaus wirth, Algorithms- Data Structures = Programs.
5. Tannenbaum, Data Structures using C and C++; PHI.
6. Thoms Horbron, -File systems – Structures and algorithms; PHI. I
7. Donald Knuth, - Art of Computer Programming Vol. I;

North Maharashtra University, Jalgaon
S. Y. B. Sc. (Computer Science)
w.e.f. June- 2016
Sem – I Paper – II
COMP-212 : Object Oriented Analysis & Design (OOAD) & Introduction to C++

- Unit 1 Introduction to OOAD** (8L,14M)
- 1.1 Object Oriented Development
 - 1.2 Benefits and applications of OOP
 - 1.3 Characteristics of Object: Identity, classification, polymorphism, Inheritance, encapsulation, abstraction.
 - 1.4 Difference between Structure oriented programming and object oriented programming.
- 2 Object Modeling & Analysis** (12L,16M)
- 2.1 Abstract & Object Modeling Technique
 - 2.2 Object & classes
 - 2.3 Advanced Link & Association Concept
 - 2.4 Generalization & Inheritance, Grouping Constraints
 - 2.5 Aggregation
 - 2.6 Metadata
 - 2.7 Concepts & Types of Models
- 3. Object Oriented Programming styles** (8L,12M)
- 3.1 Basics of Object oriented Programming styles
 - 3.2 Reusability
 - 3.3 Extensibility
 - 3.4 Robustness
- 4. Introduction to C++** (10L, 16M)
- 4.1 Basics of C++, Structure of C++ Program, keywords in C++,Data types hierarchy in C++ ,
 - 4.2 Operators in C++:
 - 4.2.1 Scope resolution operator
 - 4.2.2 Insertion and Extraction operator
 - 4.2.3 New and Delete operators, reference operators.
 - 4.3 Manipulators function: endl, setw, setfill, setprecision.
- 5 Classes and objects** (10L,16M)
- 5.1 Classes, object, Specifying a class, Access specifiers, Class members
 - 5.2 Defining member functions: Inside and Outside the class definition
 - 5.3 Creating objects, passing object as an argument
 - 5.4 Array of objects
- Unit 6 Constructors and Destructors** (12L,16M)
- 6.1 Memory allocation and static data members
 - 6.2 What is constructor?
 - 6.3 Types of constructors:
 - 6.3.1 Default Constructor

- 6.3.2 Parameterized Constructor
- 6.3.3 Copy Constructor
- 6.3.4 Overloaded constructors in a class
- 6.3.5 Constructor with default arguments
- 6.4 Destructors

Reference Books:

1. Object oriented modeling & design By James Rum Baugh, Michel Blaha, PHI, 2005
2. Introduction to object oriented analysis & Design with UML & the unified Process By Stephen R. Schach, TMH, 2003
3. Object Oriented analysis & Design with Applications, 2nd edition by Grady Boch, Pearson Education, 2004
4. Object oriented programming with C++, E Balgurusamy, 2nd edition,
5. Programming with C++ D Ravichandran
4. Programming in C++ by John H Hubbard
5. Mastering C++ by K Venugopal, Rajkumar, T Ravishankar

North Maharashtra University, Jalgaon
S. Y. B. Sc. (Computer Science)
w.e.f. June- 2016
Sem – II Paper – II
COMP-222: Programming in C++

Unit 1 Function Overloading (12L, 18M)

- 1.1 Basics of function and its need? returning objects from functions,
- 1.2 Inline functions, friend functions
- 1.3 Functions with default arguments
- 1.4 Function overloading, function overloading with arguments
- 1.5 Scoping rules & features of function overloading.

Unit 2 Operator Overloading (12L, 18M)

- 2.1 Introduction to operator overloading, rules of operator overloading
- 2.2 Operator overloading:
 - 2.2.1 Unary and binary operators,
 - 2.2.2 Comparison, arithmetic assignment operator
 - 2.3.3. Overloading new & delete operators
- 2.3 Data conversion between objects and basic data types and vice-versa.

Unit 3 Pointers in C++ & Exception Handling (12L, 18M)

- 3.1 Concept of pointer, pointer operator
- 3.2 Pointer to function.
- 3.3 Pointer and object, array of pointer to object
- 3.4 Exception Handling mechanism, Concept of try, throw and catch with example

Unit 4 Inheritance and Extending Classes (12L,20M)

- 4.1 Introduction to Inheritance
- 4.2 Types of Inheritance:
 - 4.2.1 Single level
 - 4.2.2 Multiple inheritance,
 - 4.2.3 Multilevel inheritance
 - 4.2.4 Hierarchical inheritance
 - 4.2.5 Hybrid inheritance.
- 4.3 Derived Class Constructors
- 4.4 Benefits of inheritance in C++
- 4.5 Virtual function, this pointer.
- 4.6 Abstract class, Pure virtual function.

Unit5 Templates & Introduction to Standard Template Library (12L,16M)

- 5.1 Basic of templates, Function templates, Class templates
- 5.2 Templates with multiple parameter
- 5.3 Introduction to STL,
- 5.4 Components of STL, Containers (Sequence, Associative & Derived)

Reference Books:

- 1 Object oriented programming with C++, E Balgurusamy, 2nd edition,
- 2 Programming with C++ D Ravichandran
- 3 Programming in C++ by John H Hubbard
- 4 Mastering C++ by K Venugopal, Rajkumar, T Ravishankar

North Maharashtra University, Jalgaon
S. Y. B. Sc. (Computer Science)
w.e.f. June- 2016
Sem – I Paper – III
COMP 213 : PRACTICAL COURSE

PRACTICALS BASED ON DATA STRUCTURE : I

Note : Implement all practicals using 'C' Language

1. Write a program to implement Stack operations : push, pop, peep, change, Display
2. Write a program to convert given infix expression into postfix.
3. Write a program to implement Linear Queue operations : Insert, Delete, Change, Display
4. Write a program to implement Circular queue with its operations.
5. Write a program to implement singly linked list with operations.
i) create ii) insert iii) delete iv) find v) Search
6. Write a program to implement doubly linked list with operations.
i) create ii) insert iii) delete.

PRACTICALS BASED ON C++ PROGRAMMING

1. Write a program to demonstrate all manipulators in C++.
2. Demonstrate the memory management operators : new , delete
3. Write a program to demonstrate the simple class for following objects
i) Student Information ii) Employee Information
4. Write a C++ program to demonstrate the concept of defining member function (inside, outside).
5. Write a C++ program to demonstrate the array of objects.
6. Write a C++ program to demonstrate following constructors and destructor
i) Default constructor
ii) Parameterized constructor
iii) Copy Constructor

North Maharashtra University, Jalgaon
S. Y. B. Sc. (Computer Science)
w.e.f. June- 2016
Sem – II Paper – III
COMP 223 : PRACTICAL COURSE

PRACTICALS BASED ON DATA STRUCTURE : II

1. To Create a tree and Implement following Tree Traversal Techniques:
i)inorder ii)preorder iii)postorder.
2. Implement following Graph Search Techniques:
i) BFS
ii) DFS.
3. Implement Selection Sort technique.
4. Implement Bubble and Insertion Sort techniques.
5. Implement Quick Sort technique.
6. Implement: i)Linear Search ii) Binary Search

PRACTICALS BASED ON C++ PROGRAMMING

1. Write a C++ program to demonstrate :
i)function overloading. ii) operator overloading
2. Write a C++ program to demonstrate inline function
3. Write a C++ program to demonstrate friend function
4. Write a C++ program to demonstrate virtual function.
5. Write a C++ program to demonstrate all types of Inheritances.
6. Write a C++ program to demonstrate exception handling.
7. Write a C++ program to demonstrate :
i)function template ii)class template.