## B.Sc. II YEAR Computer Science (For Regular Students)

| Paper | Subjects | Internals | | | Theory | Total (B) | Total A+B | Practical | Grand Total |
|---|---|---|---|---|---|---|---|---|---|
| | | 3 Months | 6 Months | Total (A) | | | | | |
| I | Object Oriented Programming using C++ and Java | 10 | 10 | 20 | 40 | 80 | 100 | 50 | 150 |
| II | Data Structures and Software Engineering | | | | 40 | | | | |

# ST. ALOYSIUS' COLLEGE (AUTONOMOUS) JABALPUR M.P.

## B.Sc. II YEAR COMPUTER SCIENCE

## PAPER I

## OBJECT ORIENTED PROGRAMMING USING C++ AND JAVA

**Maximum Marks: 40** **Minimim Marks:13**

**Course Objective:** To understand concepts of Object-Oriented Programming Language like classes, objects, polymorphism, memory management and garbage collection, inheritance using C++. Course is further extended to understand the process management using thread and creating GUI application in platform independent environment using java.

**Course Outcome:** Student will be able to implement classes, object, inheritance, reusability, security and many more OOPS tricks for better and effective programing in C++ and Java. Multithreading gives an opportunity to learning process working. Applet provided students to create their internet and user-friendly application.

### UNIT-I

**Introduction to C++:**Programming paradigms, Key concepts of object-oriented programming, advantages of OOP's. **Input and output in C++:** pre-defined streams, formatted and unformatted console I/O operations, parts of C++ program, types of tokens: keywords, identifiers, data types, constants, operators, precedence of operators, referencing and dereferencing operators, scope access operator. **Control statements.**

### UNIT-II

**Functions:**main(), parts of function, passing. **Arguments:** value, address, reference, inline functions, **Function Overloading:** Principles, precautions, library functions. **Classes and objects:** declaring classes and objects, accessing class members, Visibility Controls, defining member function: member function inside the class, member function outside the class, static members, friend function, friend class, overloading member function. **Constructor:** characteristics, applications, types of constructor, constructor overloading , Destructor.

### UNIT-III

**Operator overloading:** overloading unary operator, binary operator. **Inheritance:** public,private and protected inheritance, types of inheritances: single, multiple, hierarchical, multilevel, hybrid, multipath, virtual base class. Pointers: pointer declaration, pointer arithmetic, pointer to class & object, Array: declarations & initialization, arrays of classes. Polymorphism: static (Early) binding, dynamic (Late) binding, virtual function, pure virtual function.

### UNIT-IV

C++ vs JAVA, JAVA environment, Structure of Java, JAVA virtual machine. Define a Class, Adding Variables and Methods, Creating Objects, Accessing Class Members, Constructors, Method Overloading, Static Members, Nesting of Methods. **Arrays:** One Dimensional & two Dimensional, Strings, Vectors, Wrapper Classes. Define Interface, Extend Interface, Implement Interface, Package: in-built and user defined packages.

### UNIT-V

**Inheritance:** Extending a Class, Overriding Methods, Final Variables and Methods, Final Classes, Finalize Methods, Abstract methods and Classes, Visibility Control. Multithreading: Thread class, Life cycle of a thread. Writing Applets, Applets Life Cycle, Creating an Executable Applet, Applet Tag, Adding Applet to HTML File, Running the Applet.

**Text Books:**
Object-Oriented programming with ANSI & turbo C++ by Ashok N. Kamthane.
Object-Oriented programming in C++ by E.Balagurusamy
E. Balaguruswamy, "Programming In Java", 2nd Edition, TMH Publications ISBN 0-07- 463542-5

**Reference Books:**
C+ Object-Oriented programming in C++ by Robert Lafore.
The complete reference by Herbert Schildt, TMH publication.

**Suggested Practical List:**

**C++**

1. Write a program to show the implementation of Call by reference, call by value, call by address.
2. Write program to implement inline function for calculating square and cube of given number.
3. Write program in C++ to show the working of access specifier public, private, protected.
4. Write program in c++ to show the working friend function.
5. Write program in c++ to show the working static function.
6. Write program in c++ to initialized class member variables using constructor and destroy the created space using destructor.
7. Write a program to implement function overloading using class.
8. Write a program in C++ to implement unary operator overloading and binary operator overloading.
9. Write a program to show the implementation of Single, Multi level, multiple and hieratical inheritance.
10. Write a program to implement virtual function and pure virtual function.

**Java**

1. Write a program in java to find the average and sum of the N number using command line argument. 2. Write a program in java to create class Interest, initialized class using constructor, create methods like simple_interest, compound_interest.
3. Write a program in java, create a class named area and calculate area of rectangle, square, circle, triangle.
4. Write a program in java, create class product as abstract class. Create another class fan which must inherits product class.
5. Write java program, create a class string_calculate and implement following methods: toLowerCase, toUpperCase, toString, substr, equals, equalsIgnoreCase, toCompare,chatAt, trim, indexOf.
6. Write a program in java to that show implementation of Interface.
7. Write a program in Java to show implementation of Applet.
8. Write a program in Java to show implementation of Thread.
9. Write a program in java to that show implementation type casting.
10. Write a program in java to that show implementation final class.

# ST. ALOYSIUS' COLLEGE (AUTONOMOUS) JABALPUR M.P.

## B.Sc. II YEAR COMPUTER SCIENCE

## PAPER II

## DATA STRUCTURES AND SOFTWARE ENGINEERING

**Maximum Marks: 40** **Minimim Marks:13**

**Course Objectives:** The course is designed to develop skills to design and analyze linear and nonlinear data structures. To understand the algorithm for solving a problem in real world. It strengthens the ability to the students to identify and apply the suitable data structure for the given real-world problem. It enables them to gain knowledge in practical applications of data structures. The course also introduces the concepts and methods required for the construction of large software intensive systems. It aims to develop a broad understanding of the discipline of software engineering.

**Course Outcome:** At the end of the course the students will be able to design and analyze the time and space efficiency of the data structures. Students will learn to apply various techniques for efficient data storage structure in real world problems. They will be able to analyze different algorithms and their correctness and compare different techniques linear and nonlinear data structures. Students will have basic knowledge and understanding of the analysis and design of complex systems. They develop the ability to apply software engineering principles and techniques and develop, maintain and evaluate large-scale software systems.

### UNIT-I

Concept of data structure and analysis of algorithm, abstract data structure, Introduction to stack and primitive operation on stack using array. Stack applications:-Infix Prefix, Postfix and Recursion, Introduction to queues, Primitive operation on queues using array, circular queue and applications of queue.

### UNIT-II

**LINKED LIST-** Introduction to Linked List, Types of Linked List (Singly, Circular Linked List, Doubly Linked List), Basic operations on Linked List (Insertion, Deletion & Traverse), Stacks, Queues, Circular Queue using Singly Linked List and Application of Linked List.

### UNIT-III

**TREES-**Basic terminology ,Binary Trees, Tree representations as array and Linked List, Basic operation Binary tree, Traversal of Binary trees:- In-order, Preorder, Post order, Application of Binary tree, Threaded Binary tree. **SEARCHING & SORTING:** Sequential Searching, Binary search, Insertion sort, Selection sort, Quick sort, Bubble sort, Heap sort, Comparison of sorting methods. **GRAPH:** Introduction of graphs, definition, Terminology, Directed, Undirected and weighted graph, representation of graph, graph traversal-Depth first, breath first search, Spanning tree, Minimum spanning tree (kruskal , prim's), shortest path algorithm.

### UNIT-IV

**INTRODUCTION TO SOFTWARE ENGINEERING:** Software Engineering- A Layered Technology, The Software Process, Software Process Models, The Linear Sequential Model, The Prototyping Model, The RAD Model, The Incremental Model, The Spiral Model, Comparison of different life-cycle models. Requirement Analysis and Specification: Properties of a good SRS document. **DESIGN CONCEPTS AND PRINCIPLES:** Cohesion, Coupling and Functional Dependence, need of function independence, function-oriented design and object-oriented design, DFD model of a system.

## UNIT-V

**PROJECT ESTIMATION:** Project planning, Metrics for software project size estimation: Lines of Code (LOC), Function point (FP), Limitations of function point (FP) metric. **PROJECT ESTIMATION TECHNIQUES:** Empirical estimation techniques - COCOMO Model **SOFTWARE TESTING FUNDAMENTALS:** Black Box Testing - equivalence class partitioning, Boundary Value Analysis. White Box Testing, Control flow graph, Cyclomatic complexity. **SOFTWARE TESTING STRATEGIES:** Unit Testing, Regression Testing, System Testing: Alpha and Beta Testing, Integration testing, Recovery Testing, Stress Testing, Performance Testing.

### Text Books:

1. Software Engineering: A Practitioner's Approach, Author: Roger S. Pressman
2. Data Structures Through C (A practical Approach), Author: G.S. Baluja
3. An Integrated Approach to Software Engineering Author: Pankaj Jalote

### Reference Books

1. Data Structures using C++ Author: D.S. Malik Second Edition
2. Principles of Data Structures using C and C++ Author: Vinu V Das New Age International Publishers
3. A Practical Introduction to Data Structures and Algorithm Analysis Author: Clifford A. Snaffer Third Edition (Java)
4. Data Structures and Algorithms Made Easy Narasimha Karumanchi

### Practical List

1. Write a program to find the factorial of a given number using recursion
2. Write a program to find GCD using recursion
3. Write a program for bubble sorting
4. Write a program for linear search
5. Write a program for binary search
6. Write a program for selection sorting
7. Write a program for quick sorting
8. Write a program for insertion sorting
9. Write a program to print Fibonacci series using recursion
10. Write a program to perform insertion and deletion operation on stack
11. Write a program to perform insertion and deletion operation on queue using static implementation
12. Write a program to perform insertion and deletion operation on queue using dynamic implementation
13. Write a program to insert a node at the beginning in singly linked list.
14. Write a program to insert a node at the middle of singly linked list
15. Write a program to insert a node at the last of singly linked list
16. Write a program to delete a node from the beginning of singly linked list
17. Write a program to delete a node from the middle of singly linked list
18. Write a program to delete a node from last in the singly linked list.
19. Write a program to traverse all the nodes in singly linked list.

20. Write a program to insert a node in the beginning of circular linked list.
21. Write a program to insert a node in the last of circular linked list.
22. Write a program to evaluate postfix operation.