

# Lab Manual

Subject Code & Name (Pattern): 214447: Data  
Structure & Algorithms Lab (2019)

Class: SE

Semester: I

Compiled by:

Ms. J. H. Jadhav

## **VISION AND MISSION OF THE INSTITUTE**

### **Vision Statement:**

Pune Institute of Computer Technology aspires to be the leader in higher technical education and research of international repute.

### **Mission Statement:**

To be leading and most sought after Institute of education and research in emerging engineering and technology disciplines that attracts, retains and sustains gifted individuals of significant potential.

## **VISION AND MISSION OF THE DEPARTMENT**

### **Vision Statement:**

The department endeavors to be recognized globally as a centre of academic excellence & research in Information Technology.

### **Mission Statement:**

To inculcate research culture among students by imparting information technology related fundamental knowledge, recent technological trends and ethics to get recognized as globally acceptable and socially responsible professionals.

### **PEO's OF THE DEPARTMENT**

1. To produce graduates who would have developed a strong background in basic science and mathematics and ability to use these tools in their chosen fields of specialization.
2. To produce graduates who have the ability to demonstrate technical competence in the fields of information Technology and develop solutions to the problems.
3. To produce graduates who would attain professional competence through life-long learning such as advanced degrees, professional registration, and other professional activities.
4. To produce graduates who function effectively in a multi-disciplinary environment and individually, within a global, societal, and environmental context.
5. To produce graduates with ethical and moral behavior

### **PSO's OF THE DEPARTMENT**

1. Apply core aspects of Information Technology and programming paradigms in order to facilitate processing, storage, retrieval, transmission and exchange of information.
2. Design, document, and develop robust applications by considering human, financial and environmental factors using cutting edge technologies to address individual and organizational needs.
3. Work in teams to manage complex IT projects using suitable project management techniques by utilizing high level interpersonal skills.

## **PROGRAM OUTCOMES**

- 1. Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
- 2. Problem analysis:** Identify, formulate, research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
- 3. Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
- 4. Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
- 5. Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
- 6. The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
- 7. Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
- 8. Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
- 9. Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
- 10. Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
- 11. Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
- 12. Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

## Course Objectives

1. To study data structures and their implementations and applications.
2. To learn different searching and sorting techniques.
3. To study some advanced data structures such as trees, graphs and tables.
4. To learn different file organizations.
5. To learn algorithm development and analysis of algorithms.

## Course Outcomes

### **Below CO Given in SPPU Syllabus**

1. Analyze algorithms and to determine algorithm correctness and time efficiency class.
2. Implement abstract data type (ADT) and data structures for given application.
3. Design algorithms based on techniques like brute -force, divide and conquer, greedy, etc.).
4. Solve problems using algorithmic design techniques and data structures.
5. Analyze of algorithms with respect to time and space complexity.

### CO-PO-PSO Mapping

CO NO.	Course Outcomes	Program Outcomes												PSO		
		1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
CO 1	Student will be able to implement and demonstrate linear search, binary search, insertion sort, bubble sort, quick sort algorithms and analyse the same with respect to worst, best and average case behaviour of the each of the above algorithms, and select appropriate searching, sorting techniques for a given data set.	3	3	3	2	2	-	-	2	2	1	-	3	3	1	-
CO 2	Students will be able to execute and implement an application of expression conversion and evaluations by employing linear data structure Stack as an ADT	3	3	3	2	2	-	-	2	2	1	-	3	3	1	-
CO 3	Students will be able to implement and illustrate the use of linear data structure queue, circular queue.	3	3	3	2	2	-	-	2	2	1	-	3	3	1	-
CO 4	Student will able to write and implement a program to construct a nonlinear data structure threaded binary tree, expression tree, binary search tree and perform the various operations like traversals, search, delete, insert, depth of a tree ,mirror of a tree.	3	3	3	2	2	-	-	2	2	1	-	3	3	1	-
CO 5	Students will be able to create and construct a graph for real time application and implement a code to crate minimum spanning tree calculate the cost of minimum spanning tree using prims, krusklas algorithm, find the shortest path using Dijiktra's graph algorithm.	3	2	3	2	2	-	-	2	2	1	-	3	3	1	-
CO 6	Student will able to write and implement a code to create a binary heap tree and perform a heap sort on it.	3	2	3	2	2	-	-	2	2	1	-	3	3	1	-
CO 7	Students will be able implement algorithms to create and manipulate database using a file data structure	3	1	3	2	2	-	-	2	2	1	-	3	3	1	-

## INDEX OF LAB EXPERIMENTS

LAB EXPT NO	PROBLEM STATEMENT
1	<p>Consider a student database of SEIT class (at least 15 records). Database contains different fields of every student like Roll No, Name and SGPA. (array of structure)</p> <ol style="list-style-type: none"> <li>Design a roll call list, arrange list of students according to roll numbers in ascending order <b>(Use Bubble Sort)</b></li> <li>Arrange list of students alphabetically. <b>(Use Insertion sort)</b></li> <li>Arrange list of students to find out first ten toppers from a class. <b>(Use Quick sort)</b></li> <li>Search students according to SGPA. If more than one student having same SGPA, then print list of all students having same SGPA.</li> <li>Search a particular student according to name using binary search without recursion.</li> </ol> <p>(All the student records having the presence of search key should be displayed)</p> <p>(Note: Implement either Bubble sort or Insertion Sort.)</p>
2	<p>Implement stack as an abstract data type using singly linked list and use this ADT for conversion of infix expression to postfix, prefix and evaluation of postfix and prefix expression.</p>
3	<p>Implement Circular Queue using Array. Perform following operations on it.</p> <ol style="list-style-type: none"> <li>Insertion (Enqueue)</li> <li>Deletion (Dequeue)</li> <li>Display</li> </ol> <p>(Note: Handle queue full condition by considering a fixed size of a queue.)</p>
4	<p>Construct an expression tree for postfix expression and perform recursive and non- recursive Inorder, Preorder and Postorder traversals.</p>
5	<p>Implement binary search tree and perform following operations:</p> <ol style="list-style-type: none"> <li>Insert <b>(Handle insertion of duplicate entry)</b></li> <li>Delete</li> <li>Search</li> <li>Display tree (Traversal)</li> <li>Display - Depth of tree</li> <li>Display - Mirror image</li> <li>Create a copy</li> </ol>

	<p>h) Display all parent nodes with their child nodes</p> <p>i) Display leaf nodes</p> <p>j) Display tree level wise</p> <p><b>(Note: Insertion, Deletion, Search and Traversal are compulsory, from rest of operations, perform any three)</b></p>
6	Implement In-order Threaded Binary Tree. Traverse the implemented tree in Pre-order too.
7	<p>Represent a graph of your college campus using adjacency list /adjacency matrix.</p> <p>Nodes should represent the various departments/institutes and links should represent the distance between them. Find minimum spanning tree using</p> <p>a) Using Kruskal's algorithm.</p> <p>b) Using Prim's algorithm.</p> <p><b>Analyze above two algorithms for space and time complexity.</b></p>
8	Represent a graph of city using adjacency matrix /adjacency list. Nodes should represent the various landmarks and links should represent the distance between them. Find the shortest path using Dijkstra's algorithm from single source to all destination. Analyze the implemented algorithm for space and time complexity
9	Implement <b>Heap sort</b> to sort given set of values using max or min heap.
10	<p>Department maintains student's database. The file contains roll number, name, division and address. Write a program to create a sequential file to store and maintain student data.</p> <p>It should allow the user to</p> <p>a) Create a student database</p> <p>b) Add a information of student</p> <p>c) Delete information of student</p> <p>d) Search and Display information of particular student.</p> <p>i. If record of student does not exist an appropriate message is displayed.</p> <p>ii. If student record is found it should display the student details.</p> <p>e) Display Record of All students in tabular form</p>

---

**Subject Coordinator**

Mrs. J. H. Jadhav

---

**HOD IT**

Dr. A. S. Ghotkar



<b>TITLE</b>	<b>1. Sorting and searching</b>
<b>PROBLEM STATEMENT /DEFINITION</b>	<p>Database contains different Fields of every student like Roll No, Name and SGPA. (array of structure)</p> <p>a. Design a roll call list, arrange list of students according to roll numbers in ascending order (Use Bubble Sort)</p> <p>b. Arrange list of students alphabetically. (Use Insertion sort)</p> <p>c. Arrange list of students to find out first ten toppers from a class. (Use Quick sort)</p> <p>d. Search students according to SGPA. If more than one student having same SGPA, then print list of all students having same SGPA.</p> <p>e. Search a particular student according to name using binary search without recursion.</p>
<b>OBJECTIVE</b>	To learn the concept of searching, sorting.
<b>S/W PACKAGES AND HARDWARE APPARATUS USED</b>	<p>Platform: Linux OS/Ubuntu/Fedora</p> <p>Tools: GEditor, Eclipse, CDT compiler.</p> <p>PC with the configuration as: Pentium IV 1.7 GHz. 128M.B RAM, 40 G.B HDD, 15’’Color Monitor, Keyboard, Mouse</p>
<b>REFERENCES</b>	<ol style="list-style-type: none"> <li>1. Bruno R Preiss, “Data Structures and Algorithms with Object-Oriented Design Patterns in C++”, Wiley India Edition</li> <li>2. E. Balaguruswamy, “Object Oriented Programming Using C++”, Tata McGrawHill</li> </ol>
<b>STEPS</b>	Refer to student activity flow chart
<b>INSTRUCTIONS FOR WRITING JOURNAL</b>	<p><b>Title:</b></p> <p><b>Aim:</b></p> <p><b>Objectives:</b></p> <p><b>Theory:</b></p> <ol style="list-style-type: none"> <li>1. Definition</li> <li>2. Explanation (features, syntax)</li> <li>3. Diagram</li> <li>4. Example</li> <li>5. Working</li> <li>6. Advantages and disadvantages</li> <li>7. Applications</li> <li>8. Time and space complexity, if applicable</li> </ol> <p><b>Input &amp;Output:</b></p> <p><b>Validation:</b></p> <p><b>Test cases:</b></p>

---

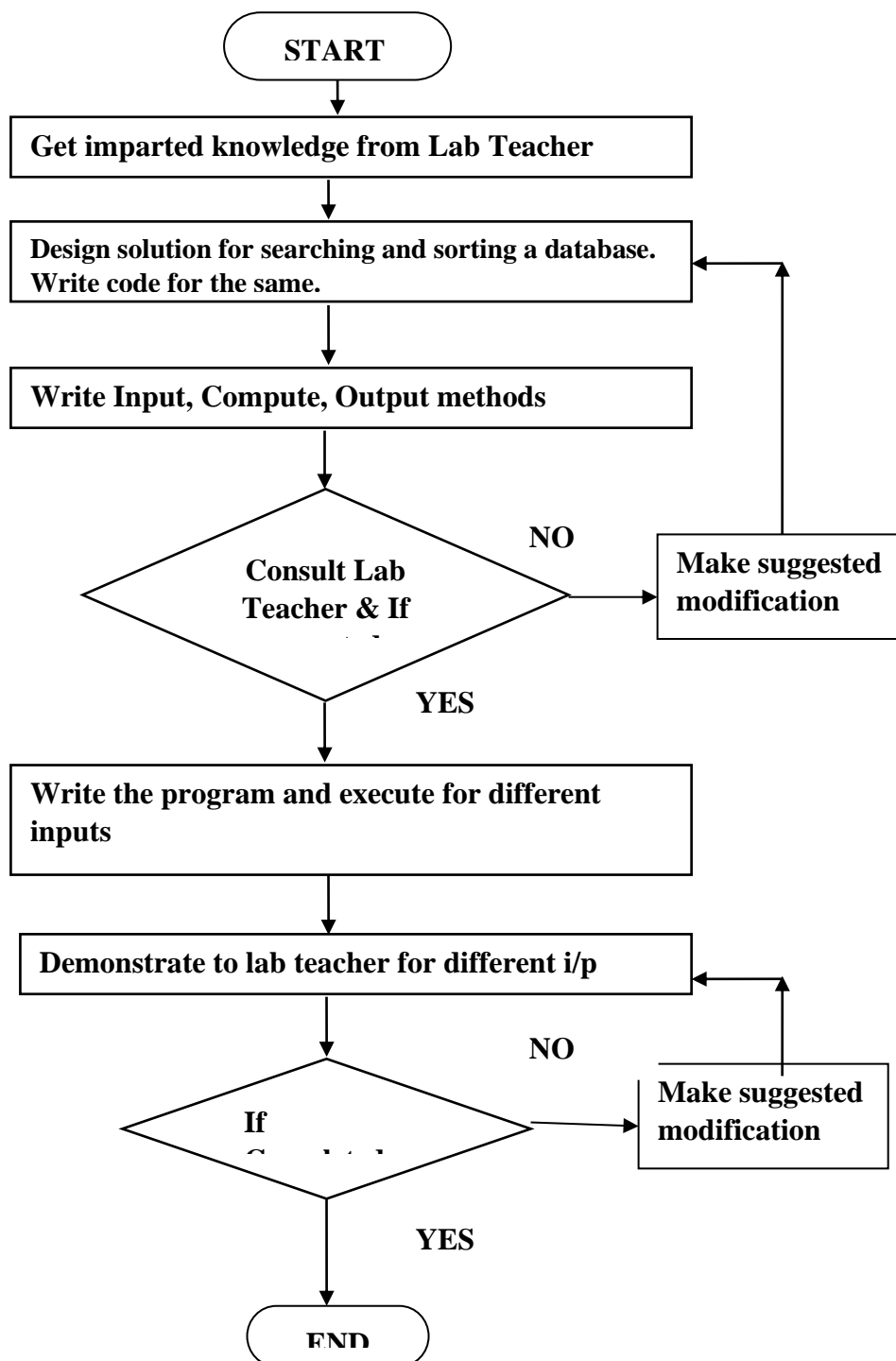
**Subject Coordinator**

Mrs. J. H. Jadhav

---

**HOD IT**

Dr. A. S. Ghotkar



<b>TITLE</b>	<b>2. Stack as ADT</b>
<b>PROBLEM STATEMENT /DEFINITION</b>	Implement stack as an abstract data type using singly linked list and use this ADT for conversion of infix expression to postfix, prefix and evaluation of postfix and prefix expression.
<b>OBJECTIVE</b>	To learn the concept stack and ADT using SLL.
<b>S/W PACKAGES AND HARDWARE APPARATUS USED</b>	Platform: Linux OS/Ubuntu/Fedora Tools: GEditor, Eclipse, Eclipse, CDT compiler. PC with the configuration as: Pentium IV 1.7 GHz. 128M.B RAM, 40 G.B HDD, 15’’Color Monitor, Keyboard, Mouse
<b>REFERENCES</b>	1. Bruno R Preiss, “Data Structures and Algorithms with Object-Oriented Design Patterns in C++”, Wiley India Edition 2. E. Balaguruswamy, “Object Oriented Programming Using C++”, Tata McGrawHill
<b>STEPS</b>	Refer to student activity flow chart
<b>INSTRUCTIONS FOR WRITING JOURNAL</b>	<b>Title:</b> <b>Aim:</b> <b>Objectives:</b> <b>Theory:</b> <ol style="list-style-type: none"> <li>1. Definition</li> <li>2. Explanation (features, syntax)</li> <li>3. Diagram</li> <li>4. Example</li> <li>5. Working</li> <li>6. Advantages and disadvantages</li> <li>7. Applications</li> <li>8. Time and space complexity, if applicable</li> </ol> <b>Input &amp; Output:</b> <b>Validation:</b> <b>Test cases:</b>

---

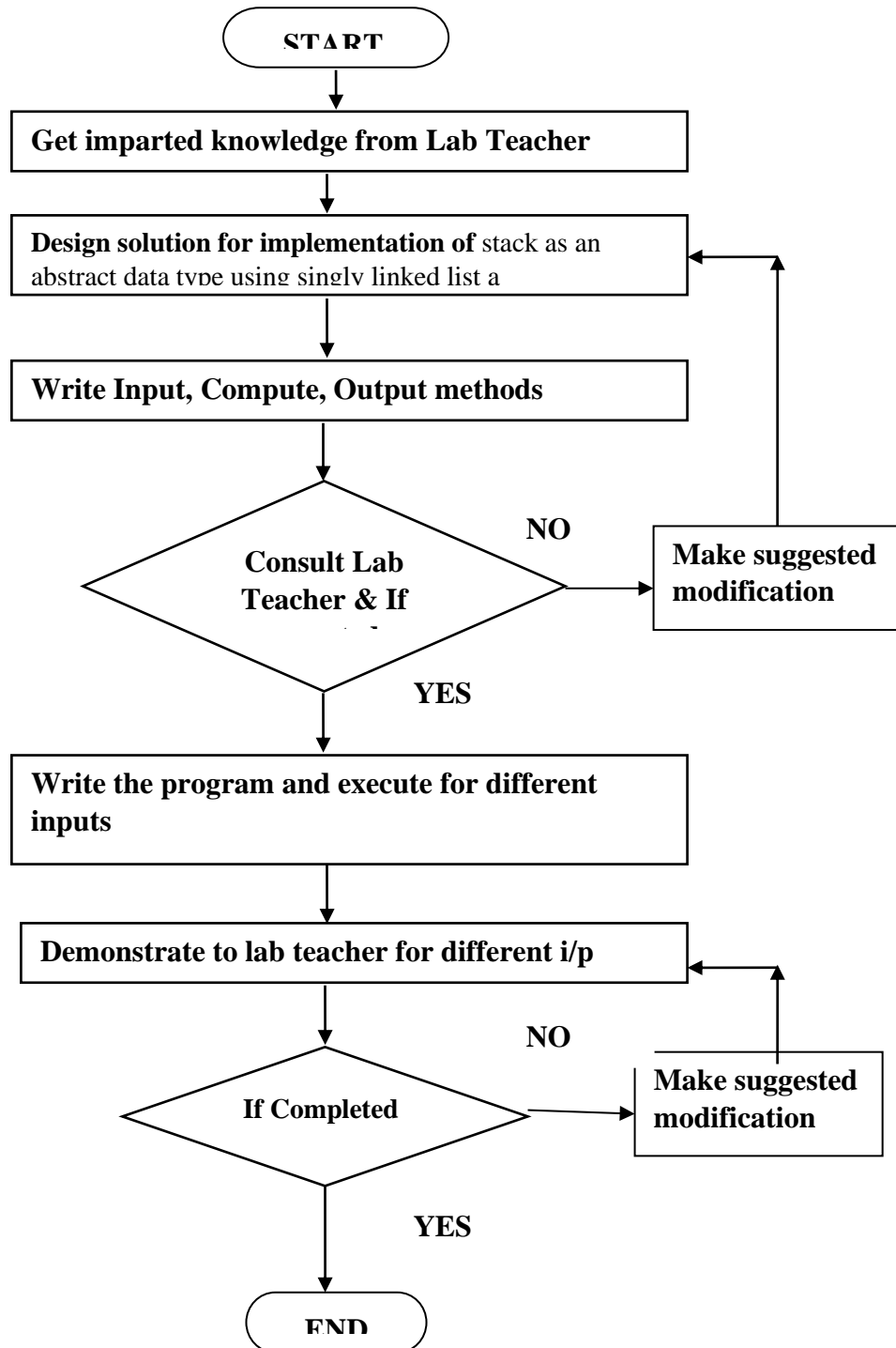
**Subject Coordinator**

Mrs. J. H. Jadhav

---

**HOD IT**

Dr. A. S. Ghotkar



<b>TITLE</b>	<b>3. Circular Queue using Array</b>
<b>PROBLEM STATEMENT /DEFINITION</b>	Implement Circular Queue using Array. Perform following operations on it. a) Insertion (Enqueue), b) Deletion (Dequeue), c) Display (Note: Handle queue full condition by considering a fixed size of a queue.)
<b>OBJECTIVE</b>	To Study concepts of Queue and types of Queue To implement circular Queue
<b>S/W PACKAGES AND HARDWARE APPARATUS USED</b>	Platform: Linux OS/Ubuntu/Fedora Tools: GEditor, Eclipse, Eclipse, CDT compiler. PC with the configuration as: Pentium IV 1.7 GHz. 128M.B RAM, 40 G.B HDD, 15’’Color Monitor, Keyboard, Mouse
<b>REFERENCES</b>	1 Bruno R Preiss, “Data Structures and Algorithms with Object-Oriented Design Patterns in C++”, Wiley India Edition 2 E. Balaguruswamy, “Object Oriented Programming Using C++” Tata McGrawHill
<b>STEPS</b>	Refer to student activity flow chart
<b>INSTRUCTIONS FOR WRITING JOURNAL</b>	<b>Title:</b> <b>Aim:</b> <b>Objectives:</b> <b>Theory:</b> <ol style="list-style-type: none"> <li>1. Definition</li> <li>2. Explanation (features, syntax)</li> <li>3. Diagram</li> <li>4. Example</li> <li>5. Working</li> <li>6. Advantages and disadvantages</li> <li>7. Applications</li> <li>8. Time and space complexity, if applicable</li> </ol> <b>Input&amp; Output:</b> <b>Validation:</b> <b>Test cases:</b>

---

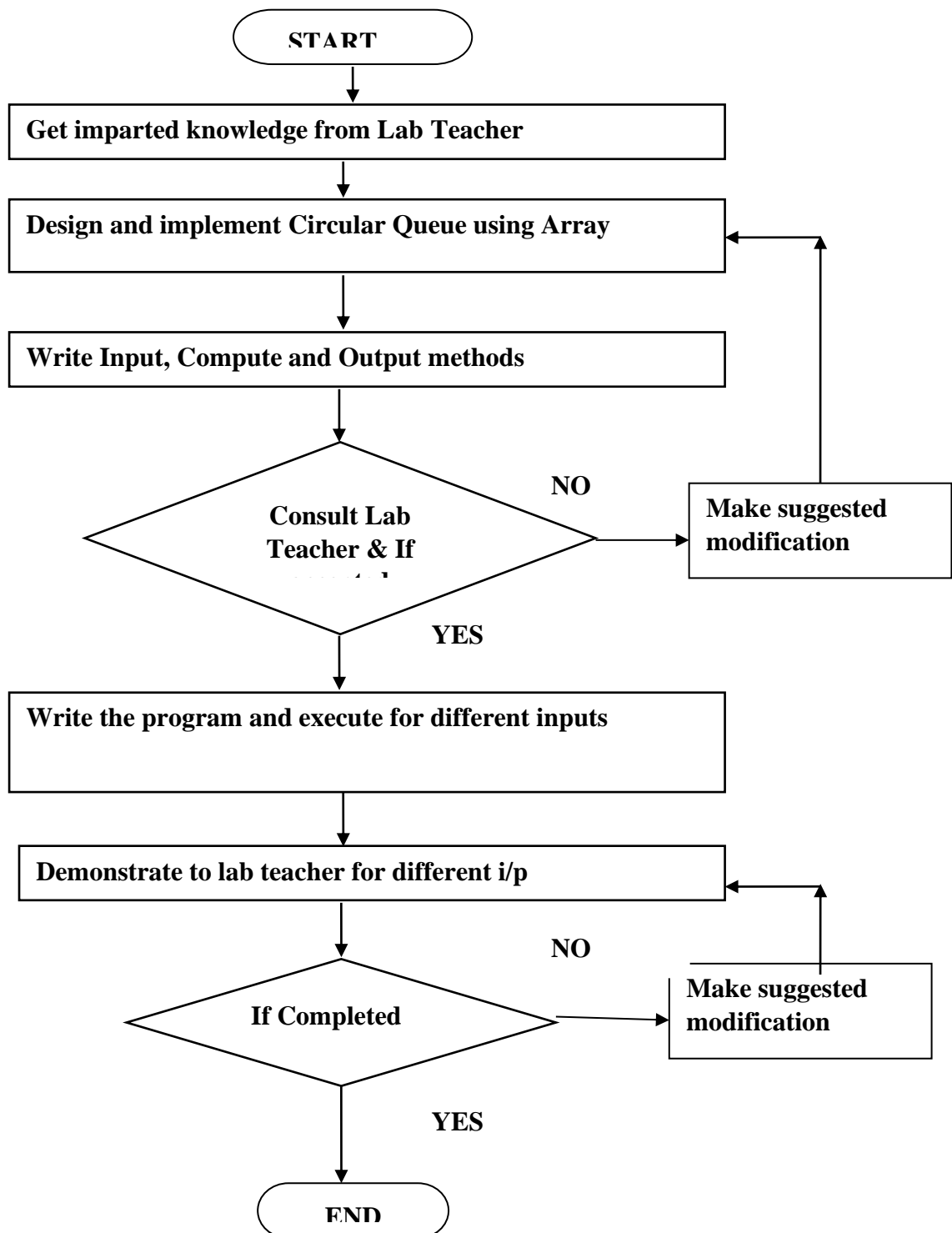
**Subject Coordinator**

Mrs. J. H. Jadhav

---

**HOD IT**

Dr. A. S. Ghotkar



<b>TITLE</b>	<b>4. Expression tree for postfix expression</b>
<b>PROBLEM STATEMENT /DEFINITION</b>	Construct an expression tree for postfix expression and perform recursive and non- recursive Inorder, Preorder and Postorder traversals.
<b>OBJECTIVE</b>	To understand the concept of expression tree. To implement expression tree for postfix expression and perform different traversals.
<b>S/W PACKAGES AND HARDWARE APPARATUS USED</b>	Platform: Linux OS/Ubuntu/Fedora Tools: GEditor, Eclipse, CDT compiler. PC with the configuration as: Pentium IV 1.7 GHz. 128M.B RAM, 40 G.B HDD, 15’’Color Monitor, Keyboard, Mouse
<b>REFERENCES</b>	1 Bruno R Preiss, “Data Structures and Algorithms with Object-Oriented Design Patterns in C++”, Wiley India Edition 2 E. Balaguruswamy, “Object Oriented Programming Using C++” Tata McGrawHill
<b>STEPS</b>	Refer to student activity flow chart
<b>INSTRUCTIONS FOR WRITING JOURNAL</b>	<p><b>Title:</b></p> <p><b>Aim:</b></p> <p><b>Objectives:</b></p> <p><b>Theory:</b></p> <ol style="list-style-type: none"> <li>1. Definition</li> <li>2. Explanation (features, syntax)</li> <li>3. Diagram</li> <li>4. Example</li> <li>5. Working</li> <li>6. Advantages and disadvantages</li> <li>7. Applications</li> <li>8. Time and space complexity, if applicable</li> </ol> <p><b>Input&amp; Output:</b></p> <p><b>Validation:</b></p> <p><b>Test cases:</b></p>

---

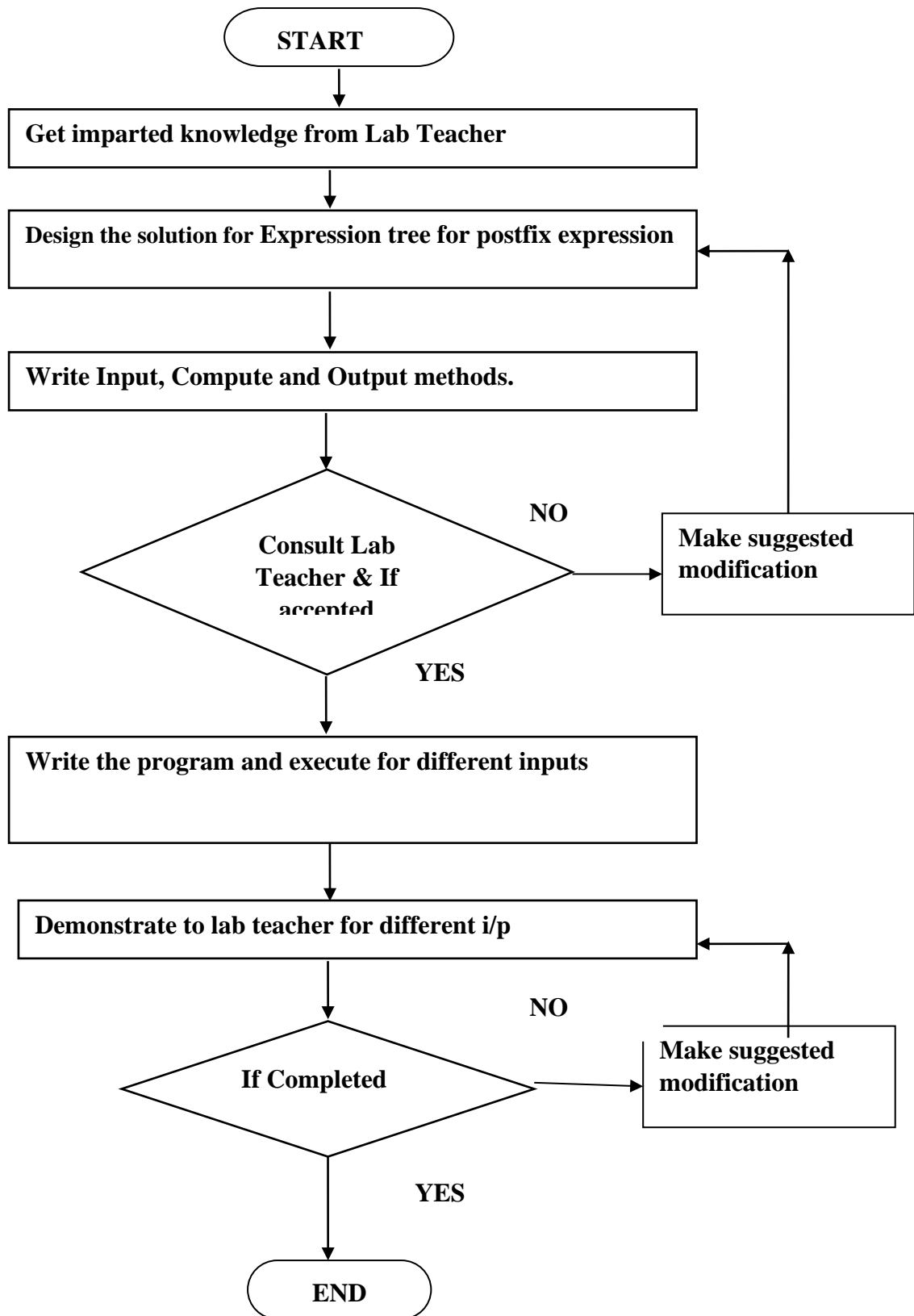
**Subject Coordinator**

Mrs. J. H. Jadhav

---

**HOD IT**

Dr. A. S. Ghotkar





<b>TITLE</b>	<b>5. Binary search tree</b>
<b>PROBLEM STATEMENT /DEFINITION</b>	<p>Implement binary search tree and perform following operations:</p> <ol style="list-style-type: none"> <li>Insert (<b>Handle insertion of duplicate entry</b>)</li> <li>Delete</li> <li>Search</li> <li>Display tree (Traversal)</li> <li>Display - Depth of tree</li> <li>Display - Mirror image</li> <li>Create a copy</li> <li>Display all parent nodes with their child nodes</li> <li>Display leaf nodes</li> <li>Display tree level wise</li> </ol> <p><b>(Note: Insertion, Deletion, Search and Traversal are compulsory, from rest of operations, perform Any three)</b></p>
<b>OBJECTIVE</b>	To understand concept of binary search tree and operations on it.
<b>S/W PACKAGES AND HARDWARE APPARATUS USED</b>	<p>Platform: Linux OS/Ubuntu/Fedora</p> <p>Tools: GEditor, Eclipse, CDT compiler.</p> <p>PC with the configuration as: Pentium IV 1.7 GHz. 128M.B RAM, 40 G.B HDD, 15’’Color Monitor, Keyboard, Mouse</p>
<b>REFERENCES</b>	<ol style="list-style-type: none"> <li>Bruno R Preiss, “Data Structures and Algorithms with Object-Oriented Design Patterns in C++”, Wiley India Edition</li> <li>E. Balaguruswamy, “Object Oriented Programming Using C++” Tata McGrawHill</li> </ol>
<b>STEPS</b>	Refer to student activity flow chart
<b>INSTRUCTIONS FOR WRITING JOURNAL</b>	<p><b>Title:</b></p> <p><b>Aim:</b></p> <p><b>Objectives:</b></p> <p><b>Theory:</b></p> <ol style="list-style-type: none"> <li>Definition</li> <li>Explanation (features, syntax)</li> <li>Diagram</li> <li>Example</li> <li>Working</li> <li>Advantages and disadvantages</li> <li>Applications</li> <li>Time and space complexity, if applicable</li> </ol> <p><b>Input&amp; Output:</b></p> <p><b>Validation:</b></p> <p><b>Test cases:</b></p>

---

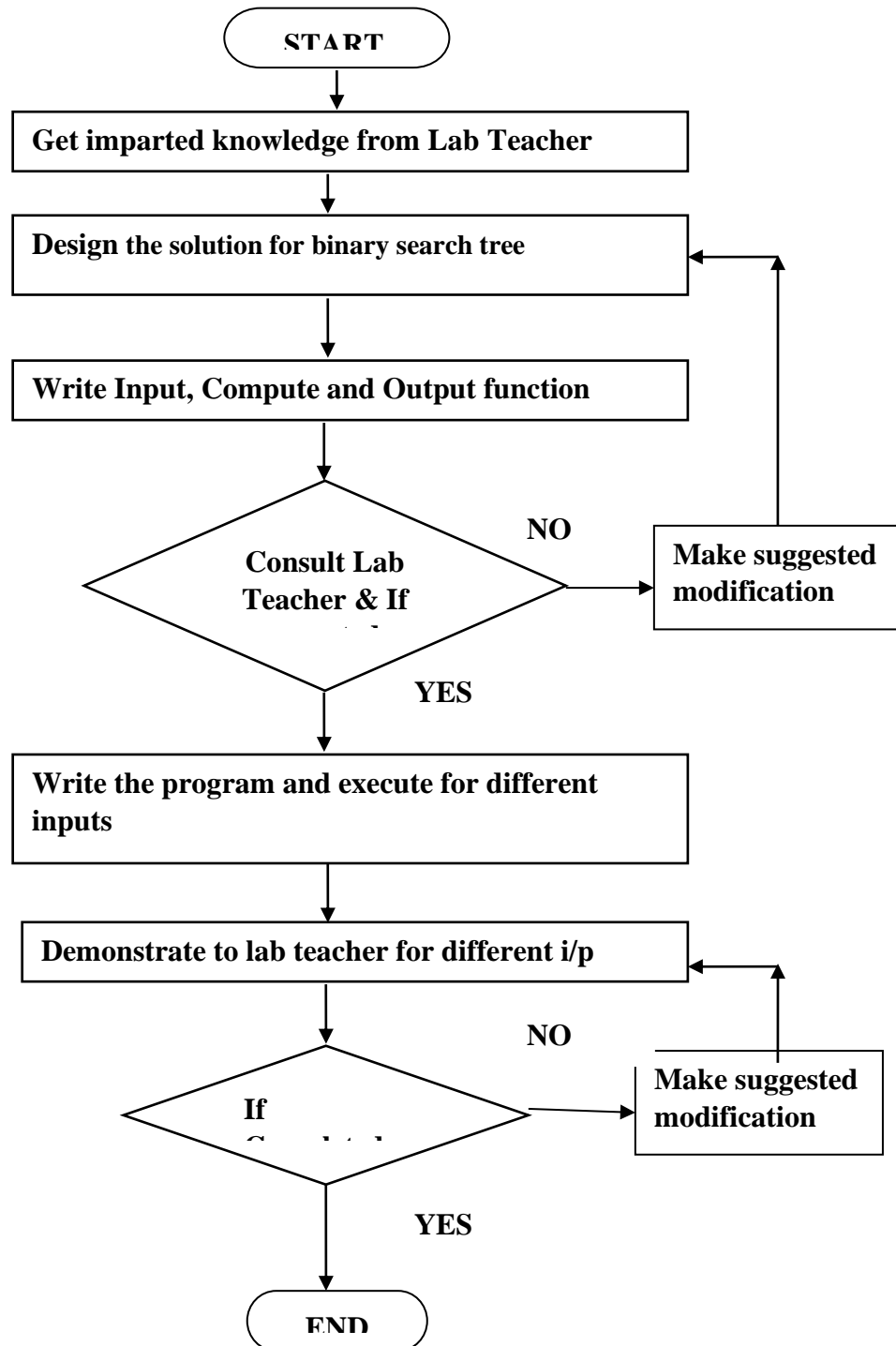
**Subject Coordinator**

Mrs. J. H. Jadhav

---

**HOD IT**

Dr. A. S. Ghotkar



<b>TITLE</b>	<b>6. In-order Threaded Binary Tree</b>
<b>PROBLEM STATEMENT /DEFINITION</b>	Implement In-order Threaded Binary Tree. Traverse the implemented tree in Pre- order too.
<b>OBJECTIVE</b>	To understand concepts of In-order Threaded Binary Tree.
<b>S/W PACKAGES AND HARDWARE APPARATUS USED</b>	Platform: Linux OS/Ubuntu/Fedora Tools: GEditor, Eclipse, CDTcompiler. PC with the configuration as: Pentium IV 1.7 GHz. 128M.B RAM, 40 G.B HDD, 15’’Color Monitor, Keyboard, Mouse
<b>REFERENCES</b>	1 Bruno R Preiss, “Data Structures and Algorithms with Object-Oriented Design Patterns in C++”, Wiley India Edition 2 E. Balaguruswamy, “Object Oriented Programming Using C++”Tata McGrawHill
<b>STEPS</b>	Refer to student activity flow chart
<b>INSTRUCTIONS FOR WRITING JOURNAL</b>	<p><b>Title:</b></p> <p><b>Aim:</b></p> <p><b>Objectives:</b></p> <p><b>Theory:</b></p> <ol style="list-style-type: none"> <li>1. Definition</li> <li>2. Explanation (features, syntax)</li> <li>3. Diagram</li> <li>4. Example</li> <li>5. Working</li> <li>6. Advantages and disadvantages</li> <li>7. Applications</li> <li>8. Time and space complexity, if applicable</li> </ol> <p><b>Input&amp; Output:</b></p> <p><b>Validation:</b></p> <p><b>Test cases:</b></p>

---

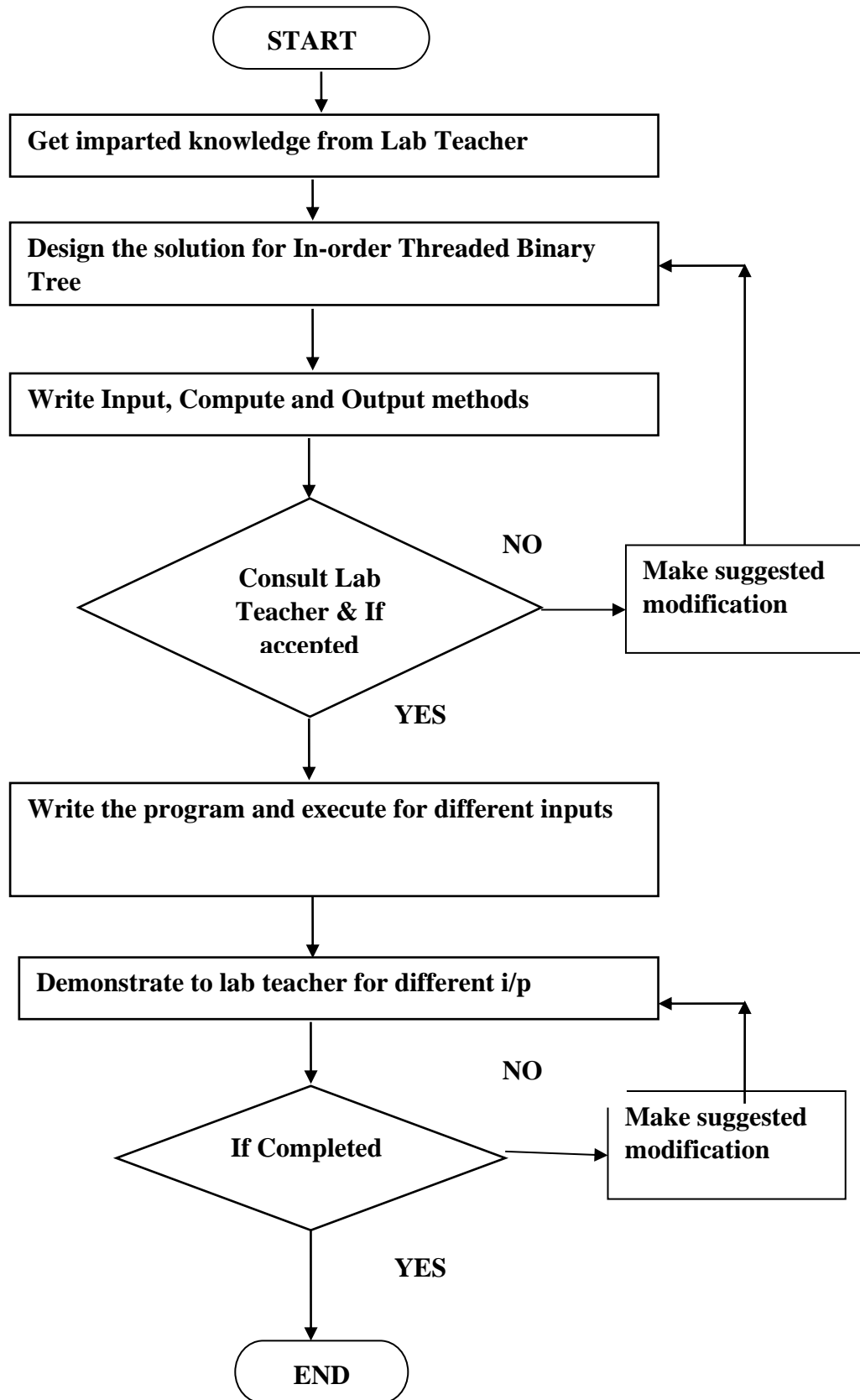
**Subject Coordinator**

Mrs. J. H. Jadhav

---

**HOD IT**

Dr. A. S. Ghotkar



<b>TITLE</b>	<b>7. Minimum spanning tree</b>
<b>PROBLEM STATEMENT /DEFINITION</b>	Represent a graph of your college campus using adjacency list /adjacency matrix. Nodes should represent the various departments/institutes and links should represent the distance between them. Find minimum spanning tree using a) Using Kruskal's algorithm. b) Using Prim's algorithm.
<b>OBJECTIVE</b>	To learn the concept of graph and minimum spanning tree.
<b>S/W PACKAGES AND HARDWARE APPARATUS USED</b>	Platform: Linux OS/Ubuntu/Fedora Tools: GEditor, Eclipse, CDT compiler. PC with the configuration as: Pentium IV 1.7 GHz. 128M.B RAM, 40 G.B HDD, 15" Color Monitor, Keyboard, Mouse
<b>REFERENCES</b>	1 Bruno R Preiss, "Data Structures and Algorithms with Object-Oriented Design Patterns in C++", Wiley India Edition 2 E. Balaguruswamy, "Object Oriented Programming Using C++" Tata McGrawHill
<b>STEPS</b>	Refer to student activity flow chart
<b>INSTRUCTIONS FOR WRITING JOURNAL</b>	<p><b>Title:</b></p> <p><b>Aim:</b></p> <p><b>Objectives:</b></p> <p><b>Theory:</b></p> <ol style="list-style-type: none"> <li>1. Definition</li> <li>2. Explanation (features, syntax)</li> <li>3. Diagram</li> <li>4. Example</li> <li>5. Working</li> <li>6. Advantages and disadvantages</li> <li>7. Applications</li> <li>8. Time and space complexity, if applicable</li> </ol> <p><b>Input&amp; Output:</b></p> <p><b>Validation:</b></p> <p><b>Test cases:</b></p>

---

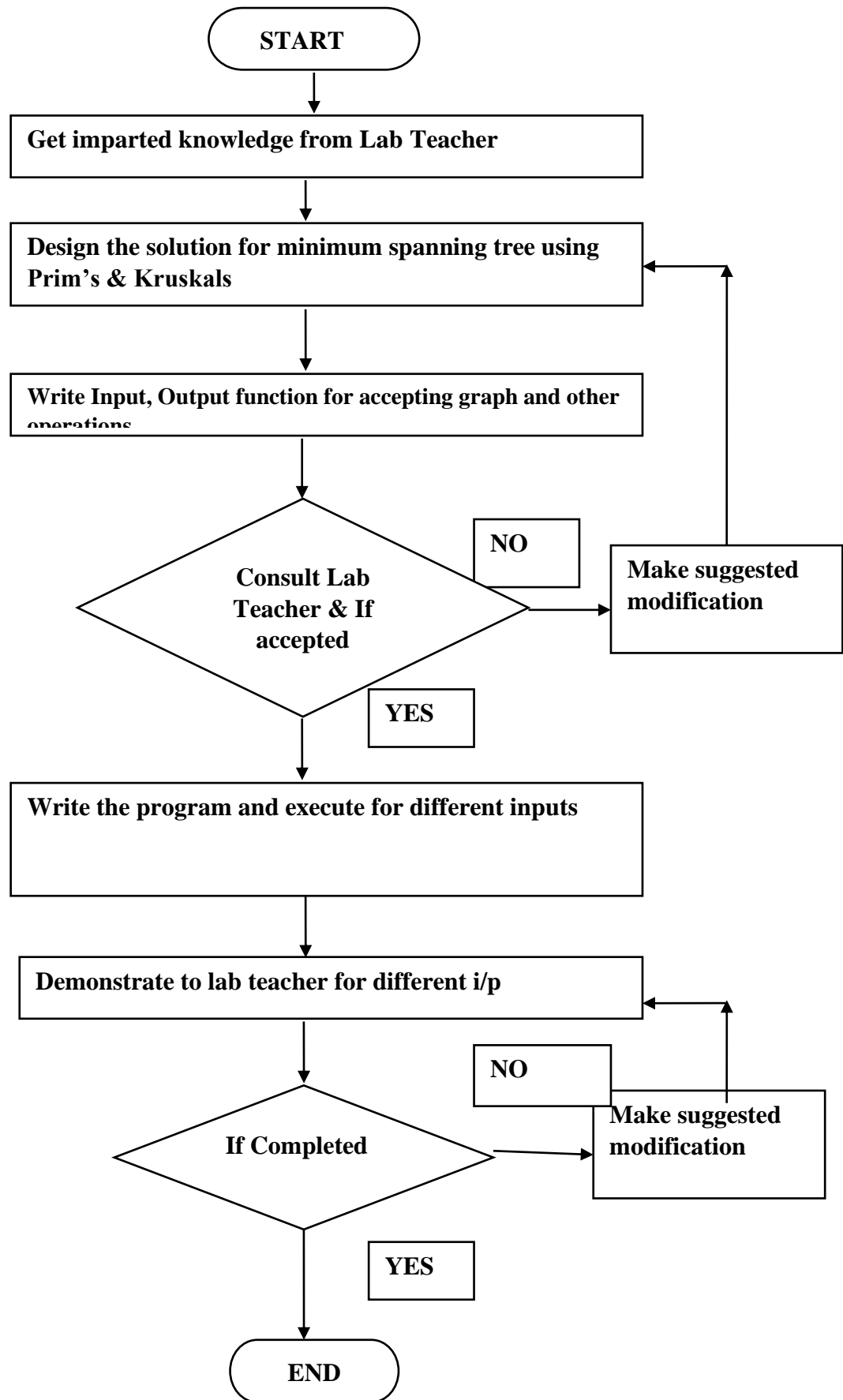
**Subject Coordinator**

Mrs. J. H. Jadhav

---

**HOD IT**

Dr. A. S. Ghotkar



<b>TITLE</b>	<b>8. Shortest path using Dijkstra's algorithm</b>
<b>PROBLEM STATEMENT /DEFINITION</b>	Represent a graph of city using adjacency matrix /adjacency list. Nodes should represent the various landmarks and links should represent the distance between them. Find the shortest path using Dijkstra's algorithm from single source to all destination. Analyze the implemented algorithm for space and time complexity
<b>OBJECTIVE</b>	To learn the concept of shortest path using Dijkstra's algorithm.
<b>S/W PACKAGES AND HARDWARE APPARATUS USED</b>	Platform: Linux OS/Ubuntu/Fedora Tools: GEditor, Eclipse, CDT compiler. PC with the configuration as: Pentium IV 1.7 GHz. 128M.B RAM, 40 G.B HDD, 15’’Color Monitor, Keyboard, Mouse
<b>REFERENCES</b>	1 Bruno R Preiss, “Data Structures and Algorithms with Object-Oriented Design Patterns in C++”, Wiley India Edition 2 E. Balaguruswamy, “Object Oriented Programming Using C++” Tata McGrawHill
<b>STEPS</b>	Refer to student activity flow chart
<b>INSTRUCTIONS FOR WRITING JOURNAL</b>	<b>Title:</b> <b>Aim:</b> <b>Objectives:</b> <b>Theory:</b> <ol style="list-style-type: none"> <li>1. Definition</li> <li>2. Explanation (features, syntax)</li> <li>3. Diagram</li> <li>4. Example</li> <li>5. Working</li> <li>6. Advantages and disadvantages</li> <li>7. Applications</li> <li>8. Time and space complexity, if applicable</li> </ol> <b>Input&amp; Output:</b> <b>Validation:</b> <b>Test cases:</b>

---

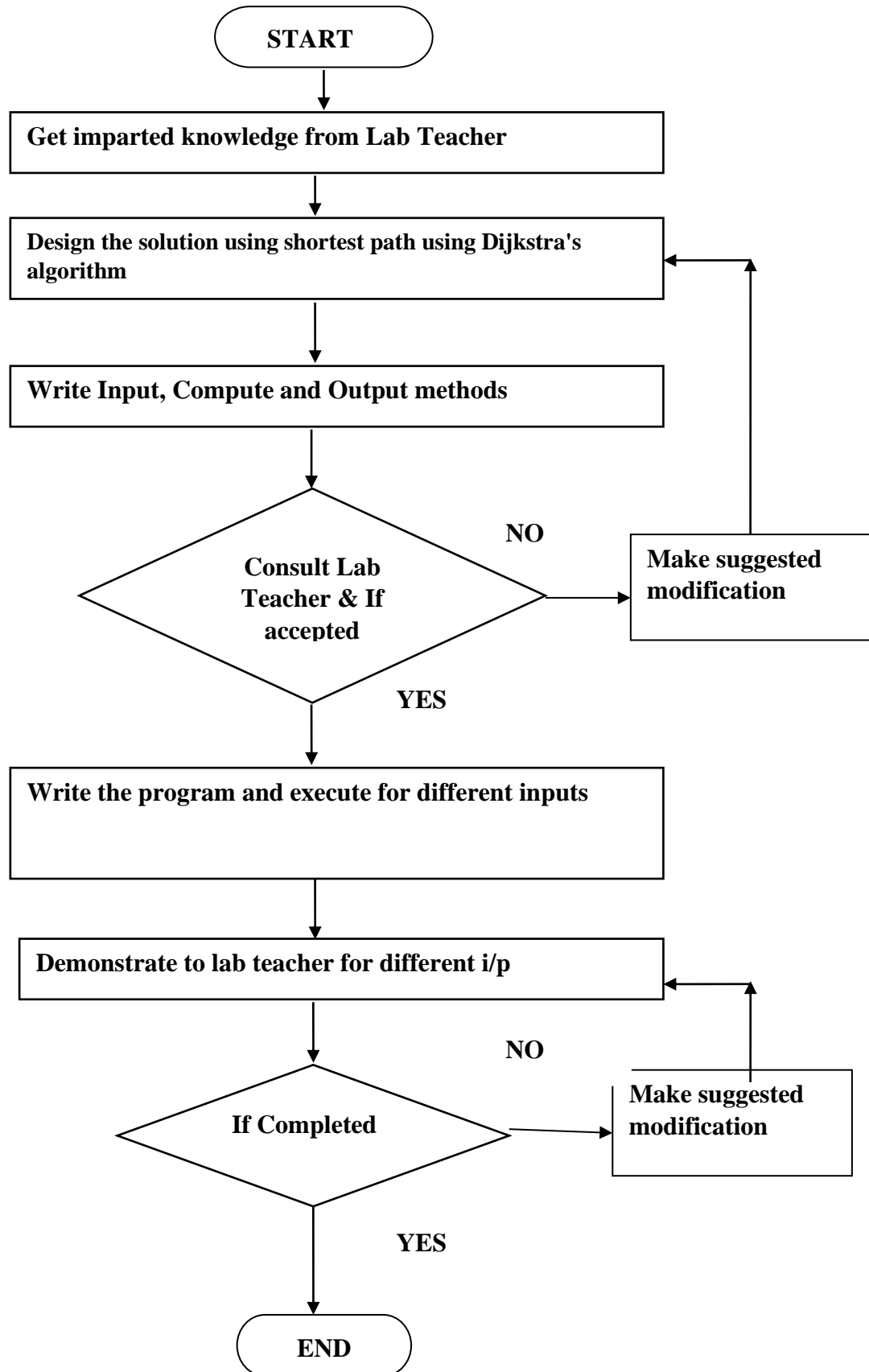
**Subject Coordinator**

Mrs. J. H. Jadhav

---

**HOD IT**

Dr. A. S. Ghotkar





<b>TITLE</b>	<b>9. Heap sort</b>
<b>PROBLEM STATEMENT /DEFINITION</b>	Implement <b>Heap sort</b> to sort given set of values using max or min heap.
<b>OBJECTIVE</b>	To learn the concept of Heap sort and its implementation.
<b>S/W PACKAGES AND HARDWARE APPARATUS USED</b>	Platform: Linux OS/Ubuntu/Fedora Tools: GEditor, Eclipse, CDT compiler. PC with the configuration as: Pentium IV 1.7 GHz. 128M.B RAM, 40 G.B HDD, 15’’Color Monitor, Keyboard, Mouse
<b>REFERENCES</b>	1 Bruno R Preiss, “Data Structures and Algorithms with Object-Oriented Design Patterns in C++”, Wiley India Edition 2 E. Balaguruswamy, “Object Oriented Programming Using C++” Tata McGrawHill
<b>STEPS</b>	Refer to student activity flow chart
<b>INSTRUCTIONS FOR WRITING JOURNAL</b>	<p><b>Title:</b></p> <p><b>Aim:</b></p> <p><b>Objectives:</b></p> <p><b>Theory:</b></p> <ol style="list-style-type: none"> <li>1. Definition</li> <li>2. Explanation (features, syntax)</li> <li>3. Diagram</li> <li>4. Example</li> <li>5. Working</li> <li>6. Advantages and disadvantages</li> <li>7. Applications</li> <li>8. Time and space complexity, if applicable</li> </ol> <p><b>Input&amp; Output:</b></p> <p><b>Validation:</b></p> <p><b>Test cases:</b></p>

---

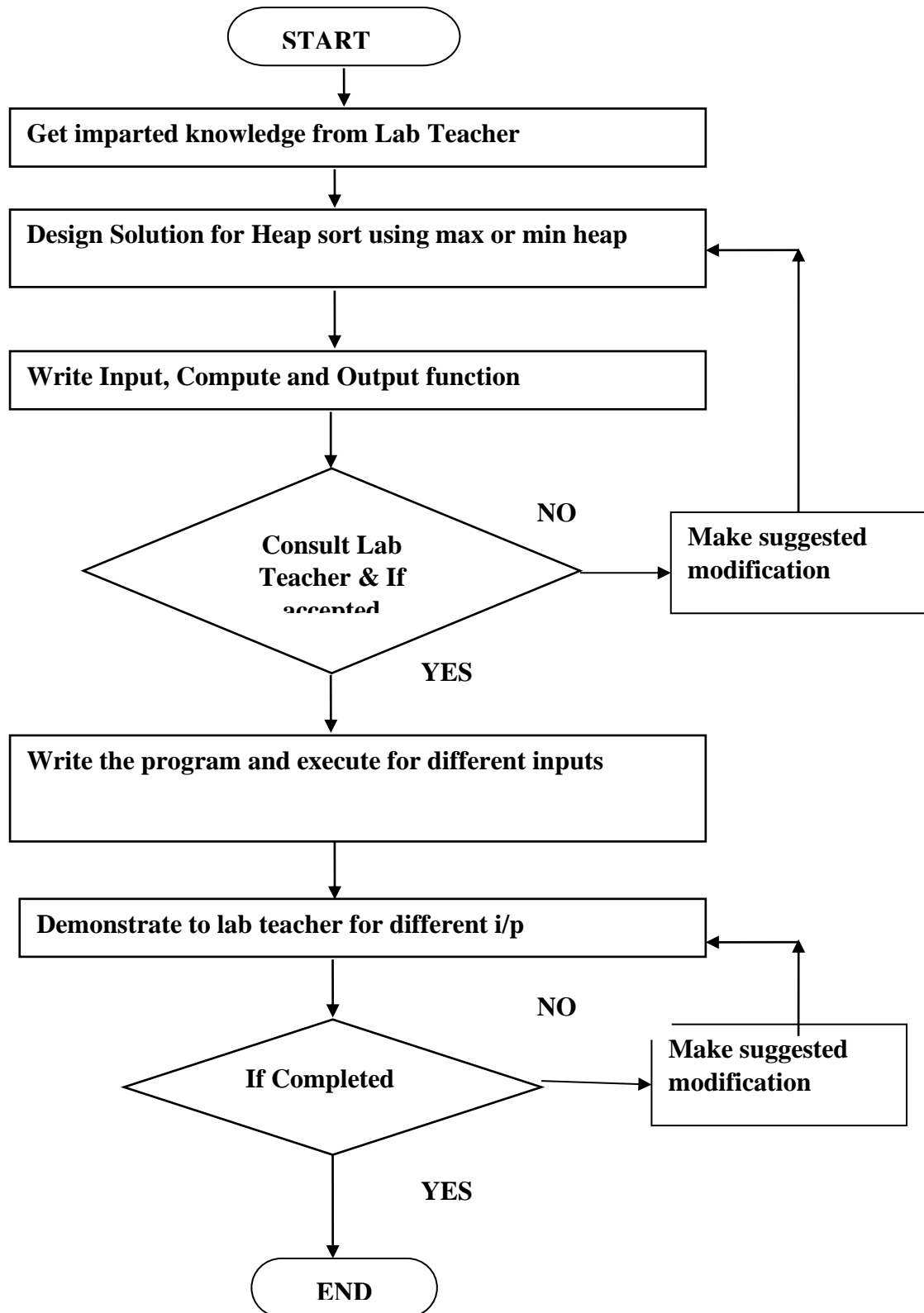
**Subject Coordinator**

Mrs. J. H. Jadhav

---

**HOD IT**

Dr. A. S. Ghotkar



<b>TITLE</b>	<b>10. Sequential file</b>
<b>PROBLEM STATEMENT /DEFINITION</b>	<p>Department maintains student's database. The file contains roll number, name, division and address. Write a program to create a sequential file to store and maintain student data. It should allow the user to</p> <ol style="list-style-type: none"> <li>Create a student database</li> <li>Add a information of student</li> <li>Delete information of student</li> <li>Search and Display information of particular student. <ol style="list-style-type: none"> <li>If record of student does not exist an appropriate message is displayed.</li> <li>If student record is found it should display the student details.</li> </ol> </li> <li>Display Record of All students in tabular form</li> </ol>
<b>OBJECTIVE</b>	To learn the concept of file handling and different operations on it.
<b>S/W PACKAGES AND HARDWARE APPARATUS USED</b>	<p>Platform: Linux OS/Ubuntu/Fedora</p> <p>Tools: GEditor, Eclipse: CDT compiler.</p> <p>PC with the configuration as: Pentium IV 1.7 GHz. 128M.B RAM, 40 G.B HDD, 15''Color Monitor, Keyboard, Mouse</p>
<b>REFERENCES</b>	<ol style="list-style-type: none"> <li>Bruno R Preiss, "Data Structures and Algorithms with Object-Oriented Design Patterns in C++", Wiley India Edition</li> <li>E. Balaguruswamy, "Object Oriented Programming Using C++"Tata McGrawHill</li> </ol>
<b>STEPS</b>	Refer to student activity flow chart
<b>INSTRUCTIONS FOR WRITING JOURNAL</b>	<p><b>Title:</b></p> <p><b>Aim:</b></p> <p><b>Objectives:</b></p> <p><b>Theory:</b></p> <ol style="list-style-type: none"> <li>Definition</li> <li>Explanation (features, syntax)</li> <li>Diagram</li> <li>Example</li> <li>Working</li> <li>Advantages and disadvantages</li> <li>Applications</li> <li>Time and space complexity, if applicable</li> </ol> <p><b>Input&amp; Output:</b></p> <p><b>Validation:</b></p> <p><b>Test cases:</b></p>

---

**Subject Coordinator**

Mrs. J. H. Jadhav

---

**HOD IT**

Dr. A. S. Ghotkar

