PUNE INSTITUTE OF COMPUTER TECHNOLOGY
DHANKAWADI, PUNE – 43.

LAB MANUAL

ACADEMIC YEAR: **2013-14**
DEPARTMENT**: INFORMATION TECHNOLOGY**
CLASS: **S.E.** SEMESTER: **II**
SUBJECT: **DATA STRUCTURES AND FILES LABORATORY**

INDEX OF LAB EXPERIMENTS

Expt. No.	Problem Definition/Statement	Revised on
1	**Implement stack as an abstract data type using linked list and use this ADT for conversion of infix expression to postfix, prefix and evaluation of postfix expression.**	**25/11/2013**
2	**Implement circular queue using array and perform followin operations on it.** **i) Add a record** **ii) Delete a record** **iii) Checking Empty** **iv) Checking Underflow** v) **Checking Overflow**	**25/11/2013**
3	**Implement priority queue as ADT using multiple linked lists ,one list for each priority for servicing patients in an hospital with priorities as i) Serious (top priority) ii) medium illness (medium priority) iii) General (Least priority).**	**25/11/2013**
4	**Construct an expression tree from postfix expression and perform recursive and non- recursive Inorder, Preorder and Postorder traversals.**	**25/11/2013**
5	**Implement binary search tree as an ADT and perform following primitive operations on it.** **i) Create a tree** **ii) Insert an element** **iii) Delete an element** iv) **Traversals – recursive and non-recursive.**	**25/11/2013**
6	**Construct an inorder threaded binary tree from postorder expression and traverse it in inorder and preorder.**	**25/11/2013**

7	**Represent any real world graph using adjacency list/adjacency matrix. Find minimum spanning tree using Prim's or Kruskal's algorithm.**	**25/11/2013**
8	**Represent a given graph using adjacency matrix/adjacency list and find the shortest path using Dijkstra's algorithm.**	**25/11/2013**
9	**Implementation of Hash table using array and handle collisions using Linear probing, chaining without replacement and Chaining with replacement.**	**25/11/2013**
10	**Implement Heap sort by constructing max or min heap.**	**25/11/2013**
11	**Implement an index sequential file for any Database and perform following operations on it** **i) Create Database** **ii) Display Database** **iii) Add a record** **iv) Delete a record** **v) Modify a record**	**25/11/2013**

Subject coordinator **HODIT**

(Prof. Nisha R. Sodha) **(Dr. Emmanuel M.)**

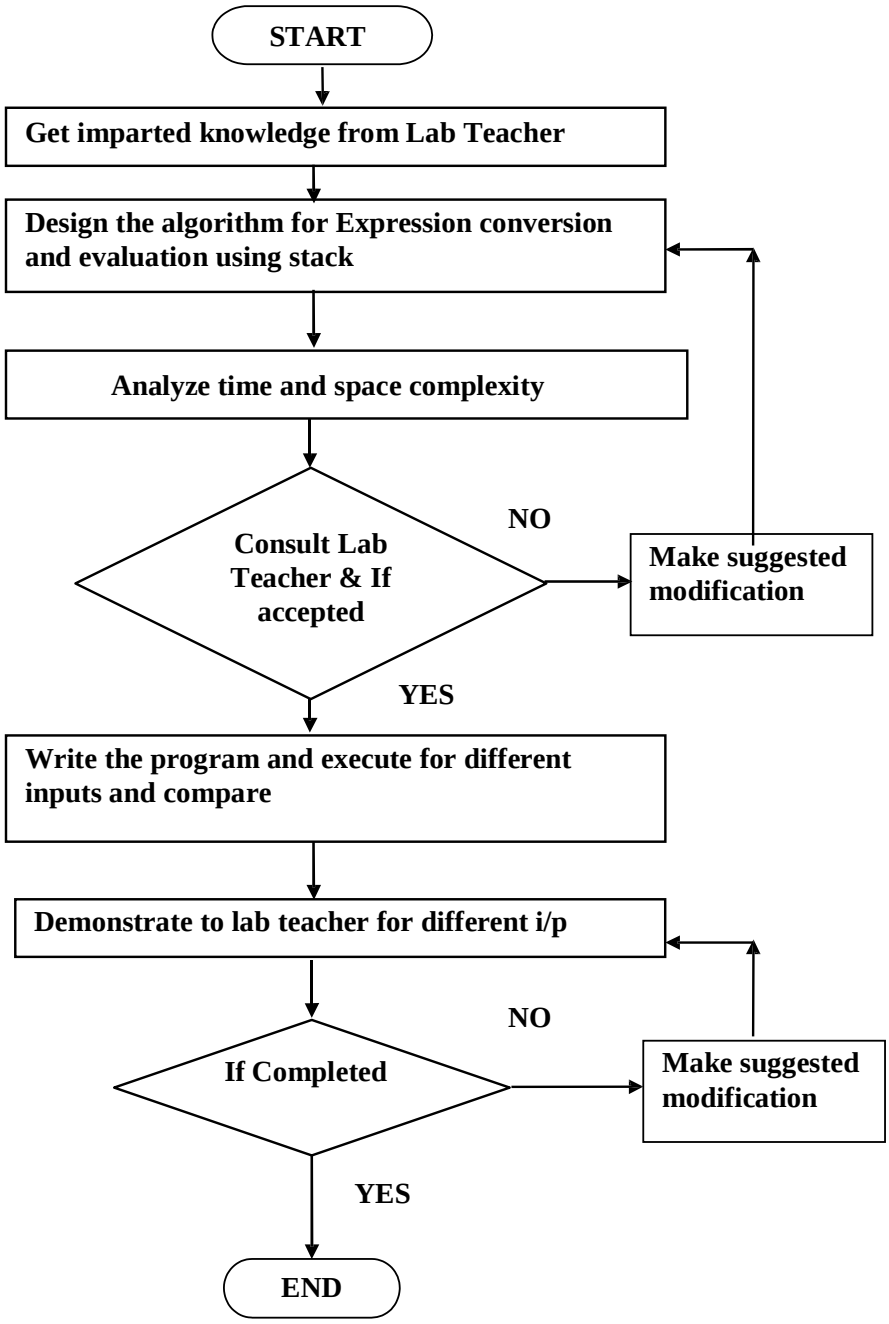
Revised On: 25/11/2013

TITLE	**EXPRESSION CONVERSION & EVALUATION**
PROBLEM STATEMENT	**Implement stack as an abstract data type using linked list and use this ADT for conversion of infix expression to postfix, prefix and evaluation of postfix expression.**
OBJECTIVE	To understand the concept of stack To understand the concept of Stack as an ADT using array and Linked List. To convert Infix to postfix and prefix and evaluation. Analyze the above algorithms
S/W PACKAGES AND HARDWARE APPARATUS USED	Linux OS: Fedora/ubantu, Eclipse/Codeblock like tools PC with the configuration as Pentium IV 1.7 GHz. 128M.B RAM, 40 G.B HDD, 15''Color Monitor, Keyboard, Mouse
REFERENCES	• R. Gilberg, B. Forouzan, "Data Structures: A pseudo code approach with C", Cenage Learning • "Fundamentals of Data Structures in C", E. Horowitz , S.Sahani
STEPS	Refer to student activity flow chart
INSTRUCTIONS FOR WRITING JOURNAL	• Title • Problem Definition • Concept of Stack • Algorithms for Infix to postfix conversion, Infix to prefix conversion, and evaluation of prefix and postfix expression. • Analysis of above for time and space complexity • Program code • Output for different i/p comparing time complexity • Conclusion

Subject coordinator **HODIT**

(Prof. Nisha R. Sodha) **(Dr. Emmanuel M.)**

STUDENT ACTIVITY CHART



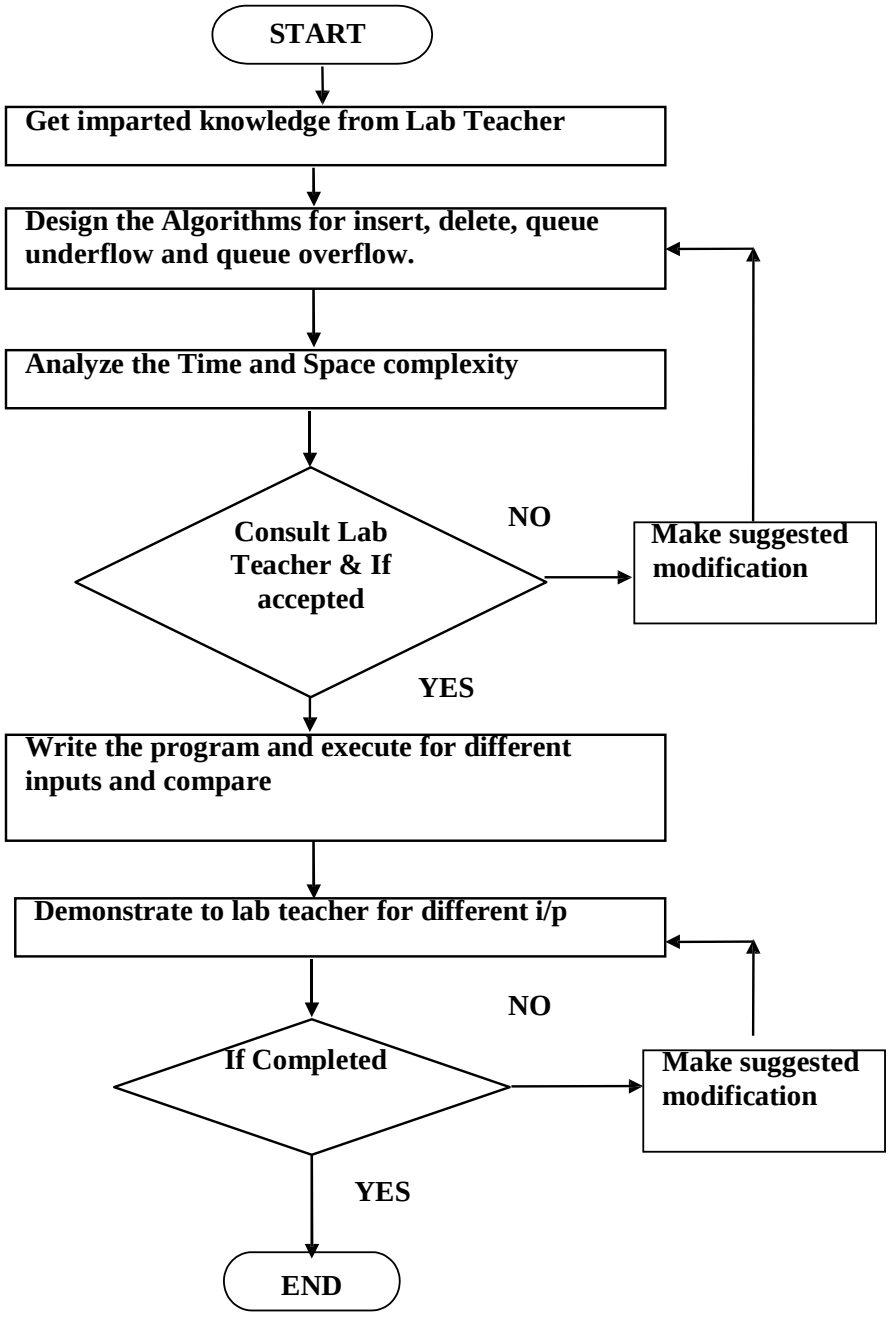
Revised On: 25/11/2013

TITLE	**CIRCULAR QUEUE**
PROBLEM STATEMENT /DEFINITION	**Implement circular queue using array and perform following operations on it.** **vi) Add a record** **vii) Delete a record** **viii) Checking Empty** **ix) Checking Underflow** **x) Checking Overflow**
OBJECTIVE	To understand the concept of Queue To understand the concept of Circular Queue
S/W PACKAGES AND HARDWARE APPARATUS USED	Linux OS: Fedora/ubantu, Eclipse/Codeblock like tools PC with the configuration as Pentium IV 1.7 GHz. 128M.B RAM, 40 G.B HDD, 15''Color Monitor, Keyboard, Mouse
REFERENCES	• R. Gilberg, B. Forouzan, "Data Structures: A pseudo code approach with C", Cenage Learning. • "Fundamentals of Data Structures in C", E. Horowitz , S.Sahani • "An Introduction to Data Structures with Applications", Jean Paul Tremblay, Paul G. Sorenson.
STEPS	Refer to student activity flow chart
INSTRUCTIONS FOR WRITING JOURNAL	• Title • Problem Definition • Concept of Queue, and Circular Queue • Algorithms for Insert, delete, queue empty. • Analysis of above for time and space complexity • Program code • Output for different I/P comparing time complexity • Conclusion

Subject coordinator **HODIT**

(Prof. Nisha R. Sodha) **(Dr. Emmanuel M.)**

STUDENT ACTIVITY CHART



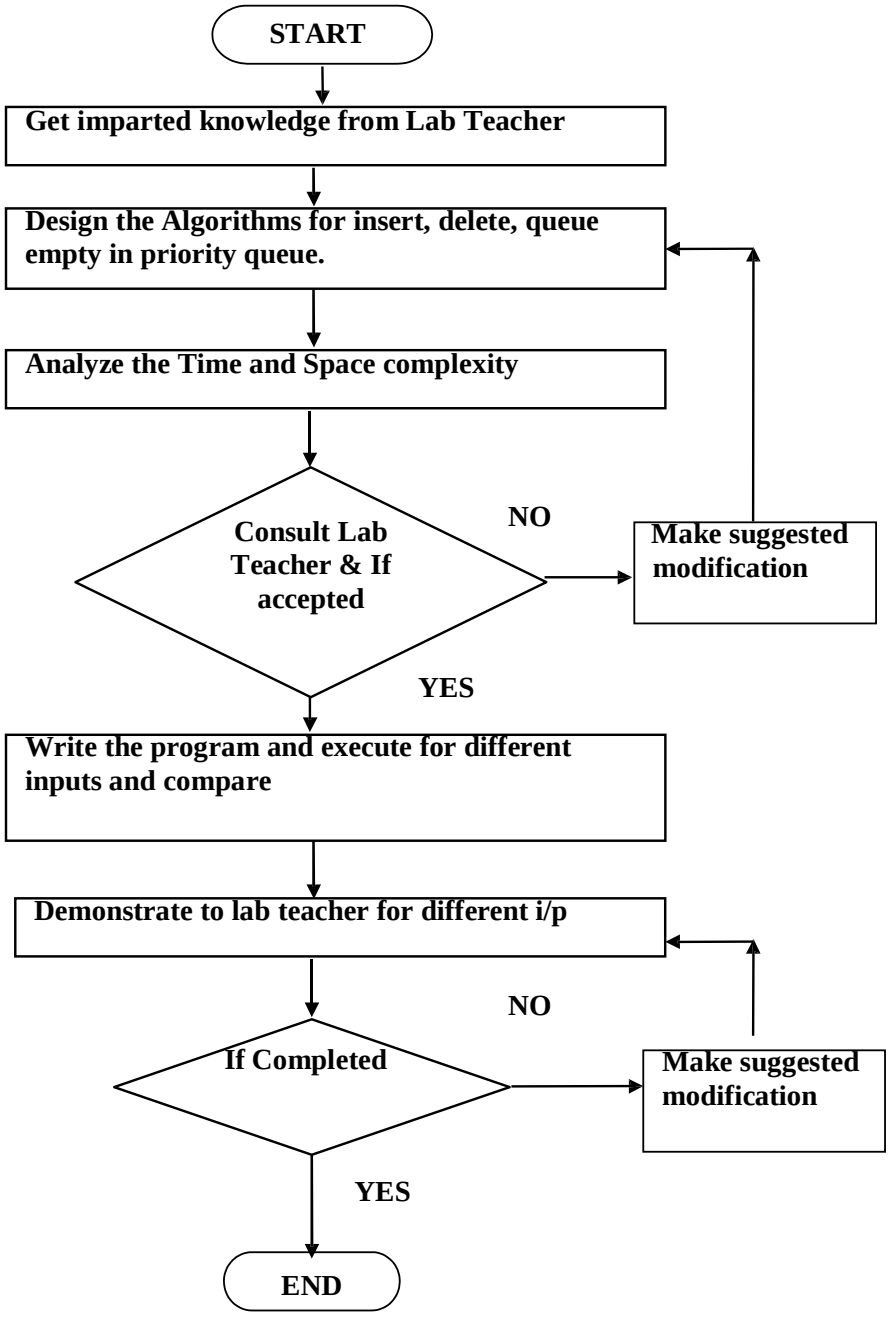
Revised On: 25/11/2013

TITLE	**PRIORITY QUEUE**
PROBLEM STATEMENT /DEFINITION	**Implement priority queue as ADT using multiple linked lists ,one list for each priority for servicing patients in an hospital with priorities as i) Serious (top priority) ii) medium illness (medium priority) iii) General (Least priority).**
OBJECTIVE	To understand the concept of Queue To understand the concept of Priority Queue
S/W PACKAGES AND HARDWARE APPARATUS USED	Linux OS: Fedora/ubantu, Eclipse/Codeblock like tools PC with the configuration as Pentium IV 1.7 GHz. 128M.B RAM, 40 G.B HDD, 15''Color Monitor, Keyboard, Mouse
REFERENCES	• R. Gilberg, B. Forouzan, "Data Structures: A pseudo code approach with C", Cenage Learning. • "Fundamentals of Data Structures in C", E. Horowitz , S.Sahani "An Introduction to Data Structures with Applications", Jean Paul Tremblay, Paul G. Sorenson.
STEPS	Refer to student activity flow chart
INSTRUCTIONS FOR WRITING JOURNAL	• Title • Problem Definition • Concept of Queue, and Priority Queue • Algorithms for Insert, delete, queue empty. • Analysis of above for time and space complexity • Program code • Output for different I/P comparing time complexity • Conclusion

Subject coordinator **HODIT**

(Prof. Nisha R. Sodha) **(Dr. Emmanuel M.)**

STUDENT ACTIVITY CHART



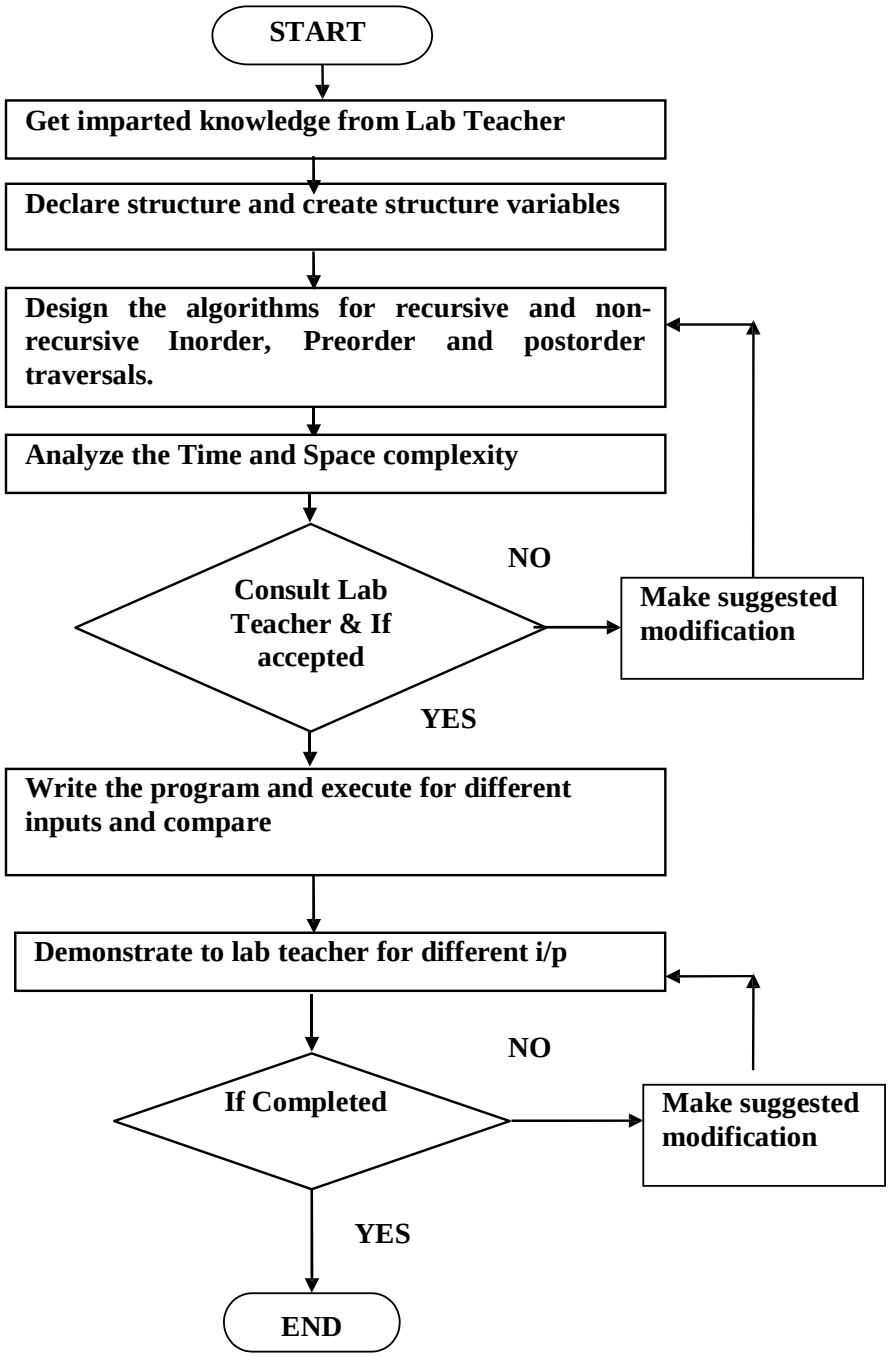
Revised On: 25/11/2013

TITLE	**EXPRESSION TREE**
PROBLEM STATEMENT /DEFINITION	**Construct an expression tree from postfix expression and perform recursive and non- recursive Inorder, Preorder and Postorder traversals.**
OBJECTIVE	To learn tree representations To understand tree creation and traversals
S/W PACKAGES AND HARDWARE APPARATUS USED	Linux OS: Fedora/ubantu, Eclipse/Codeblock like tools PC with the configuration as Pentium IV 1.7 GHz. 128M.B RAM, 40 G.B HDD, 15''Color Monitor, Keyboard, Mouse
REFERENCES	• R. Gilberg, B. Forouzan, "Data Structures: A pseudo code approach with C", Cenage Learning. • "Fundamentals of Data Structures in C", E. Horowitz , S.Sahani "An Introduction to Data Structures with Applications", Jean Paul Tremblay, Paul G. Sorenson.
STEPS	Refer to student activity flow chart
INSTRUCTIONS FOR WRITING JOURNAL	• Title • Problem Definition • Concept of tree and tree representations. • Algorithms for recursive and non-recursive Inorder, Preorder and postorder traversals. • Analysis of above for time and space complexity • Program code • Output for different I/P comparing time complexity • Conclusion

Subject coordinator **HODIT**

(Prof. Nisha R. Sodha) **(Dr. Emmanuel M.)**

STUDENT ACTIVITY CHART



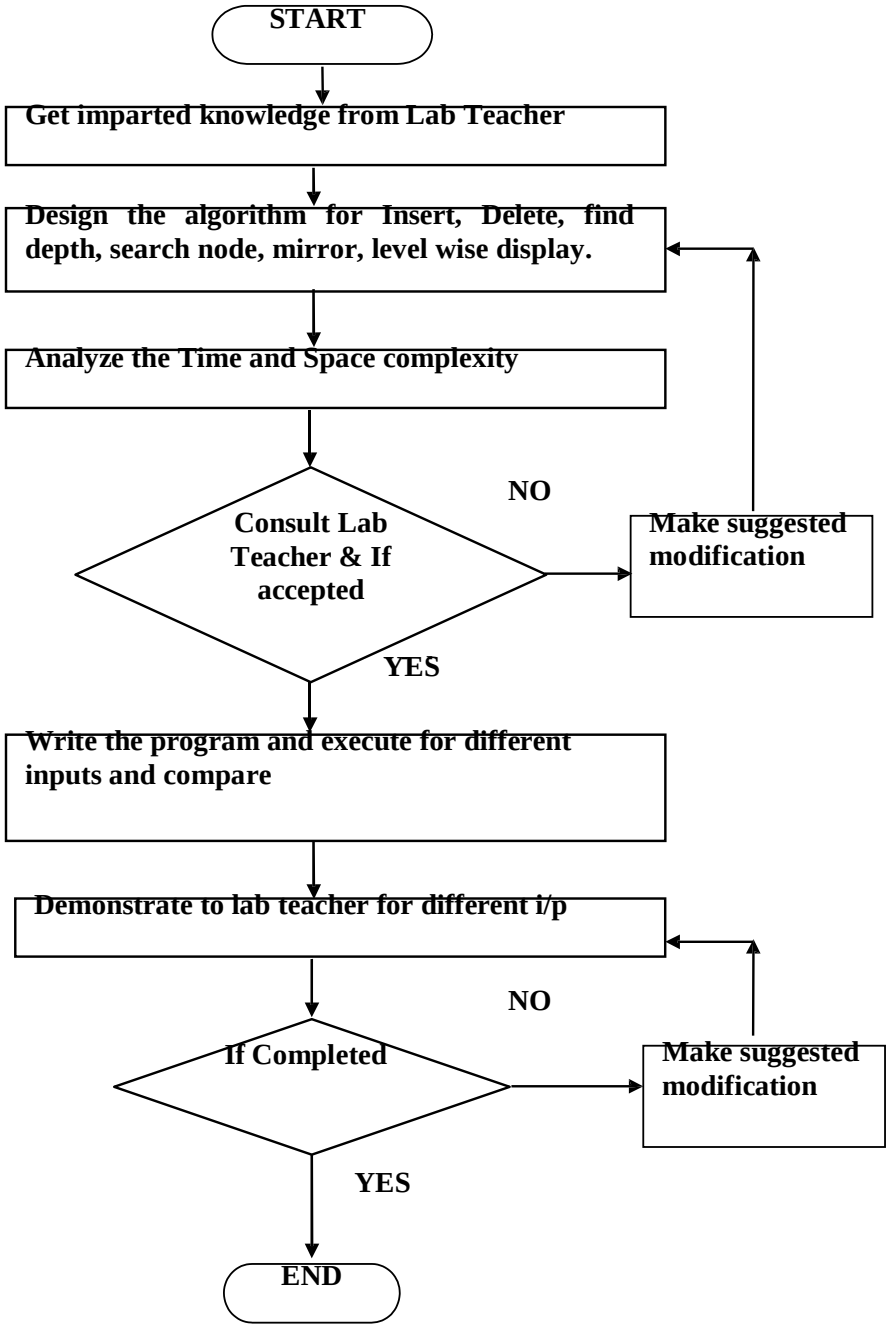
Revised On: 25/11/2013

TITLE	**BINARY SEARCH TREE**
PROBLEM STATEMENT /DEFINITION	**Implement binary search tree as an ADT and perform following primitive operations on it.** **i) Create a tree** **ii) Insert an element** **iii) Delete an element** iv) **Traversals – recursive and non-recursive.**
OBJECTIVE	To understand creation of BST To understand BST as an ADT To understand applications of BST
S/W PACKAGES AND HARDWARE APPARATUS USED	Linux OS: Fedora/ubantu, Eclipse/Codeblock like tools PC with the configuration as Pentium IV 1.7 GHz. 128M.B RAM, 40 G.B HDD, 15''Color Monitor, Keyboard, Mouse
REFERENCES	• "Fundamentals of Data Structures in C", E. Horowitz , S.Sahani "An Introduction to Data Structures with Applications", Jean Paul Tremblay, Paul G. Sorenson. • "Data Structures and Algorithm Analysis in C++", M. Weiss
STEPS	Refer to student activity flow chart
INSTRUCTIONS FOR WRITING JOURNAL	• Title • Problem Definition • Concept of BST • Algorithms to Create, Insert, Delete & traverse BST.. • Analysis of above for time and space complexity • Program code • Output for different I/P comparing time complexity • Conclusion

Subject coordinator **HODIT**

(Prof. Nisha R. Sodha) **(Dr. Emmanuel M.)**

STUDENT ACTIVITY CHART



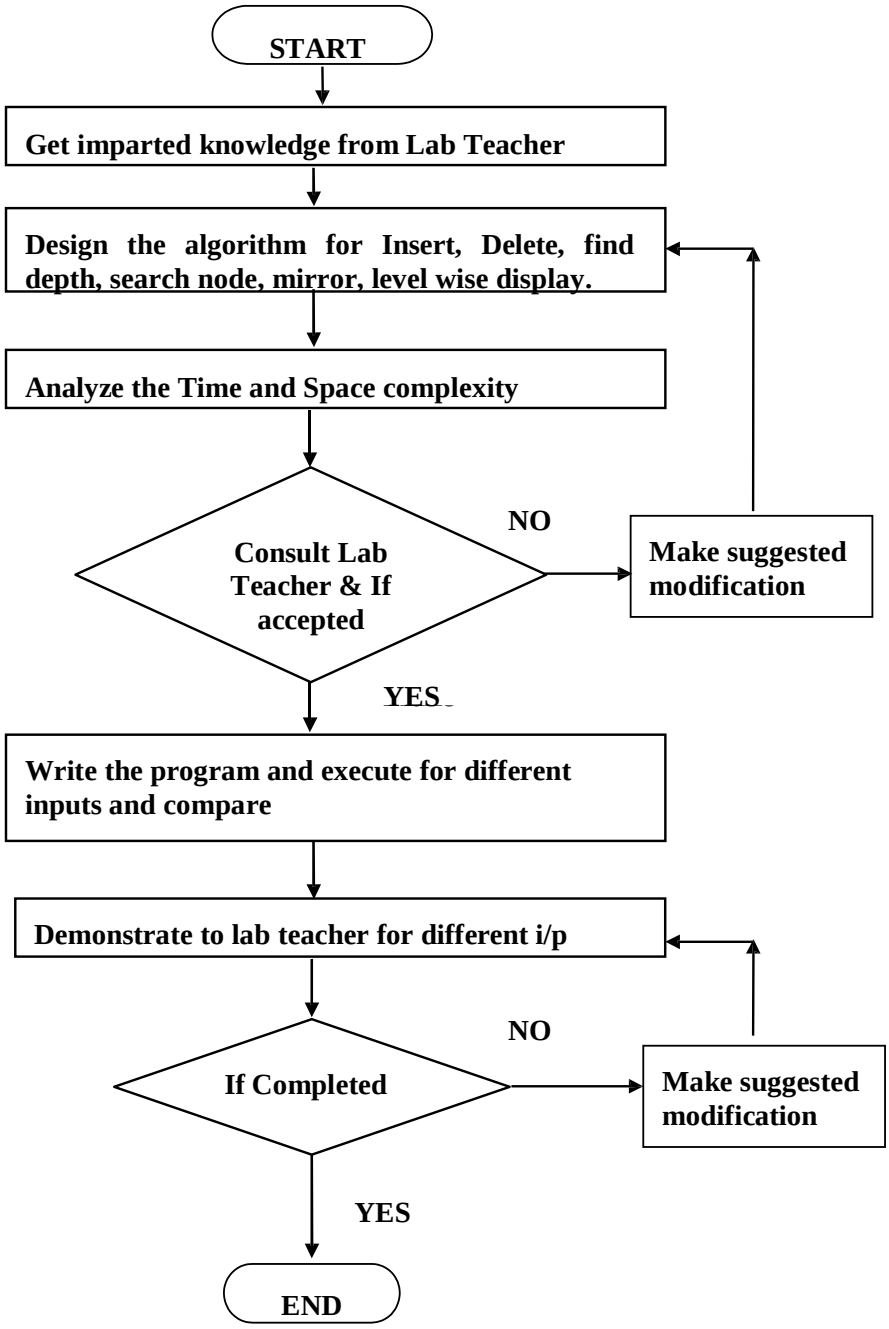
Revised On: 25/11/2013

TITLE	**INORDER THREADED BINARY TREE**
PROBLEM STATEMENT /DEFINITION	**Construct an inorder threaded binary tree from postorder expression and traverse it in inorder and preorder.**
OBJECTIVE	To understand concept of inorder TBT To understand applications of TBT
S/W PACKAGES AND HARDWARE APPARATUS USED	Linux OS: Fedora/ubantu, Eclipse/Codeblock like tools PC with the configuration as Pentium IV 1.7 GHz. 128M.B RAM, 40 G.B HDD, 15''Color Monitor, Keyboard, Mouse
REFERENCES	• "Fundamentals of Data Structures in C", E. Horowitz , S.Sahani • "An Introduction to Data Structures with Applications", Jean Paul Tremblay, Paul G. Sorenson. • "Data Structures and Algorithm Analysis in C++", M. Weiss
STEPS	Refer to student activity flow chart
INSTRUCTIONS FOR WRITING JOURNAL	• Title • Problem Definition • Concept of TBT • Algorithm to construct & traverse TBT. • Analysis of above for time and space complexity • Program code • Output for different I/P comparing time complexity • Conclusion

Subject coordinator **HODIT**

(Prof. Nisha R. Sodha) **(Dr. Emmanuel M.)**

STUDENT ACTIVITY CHART



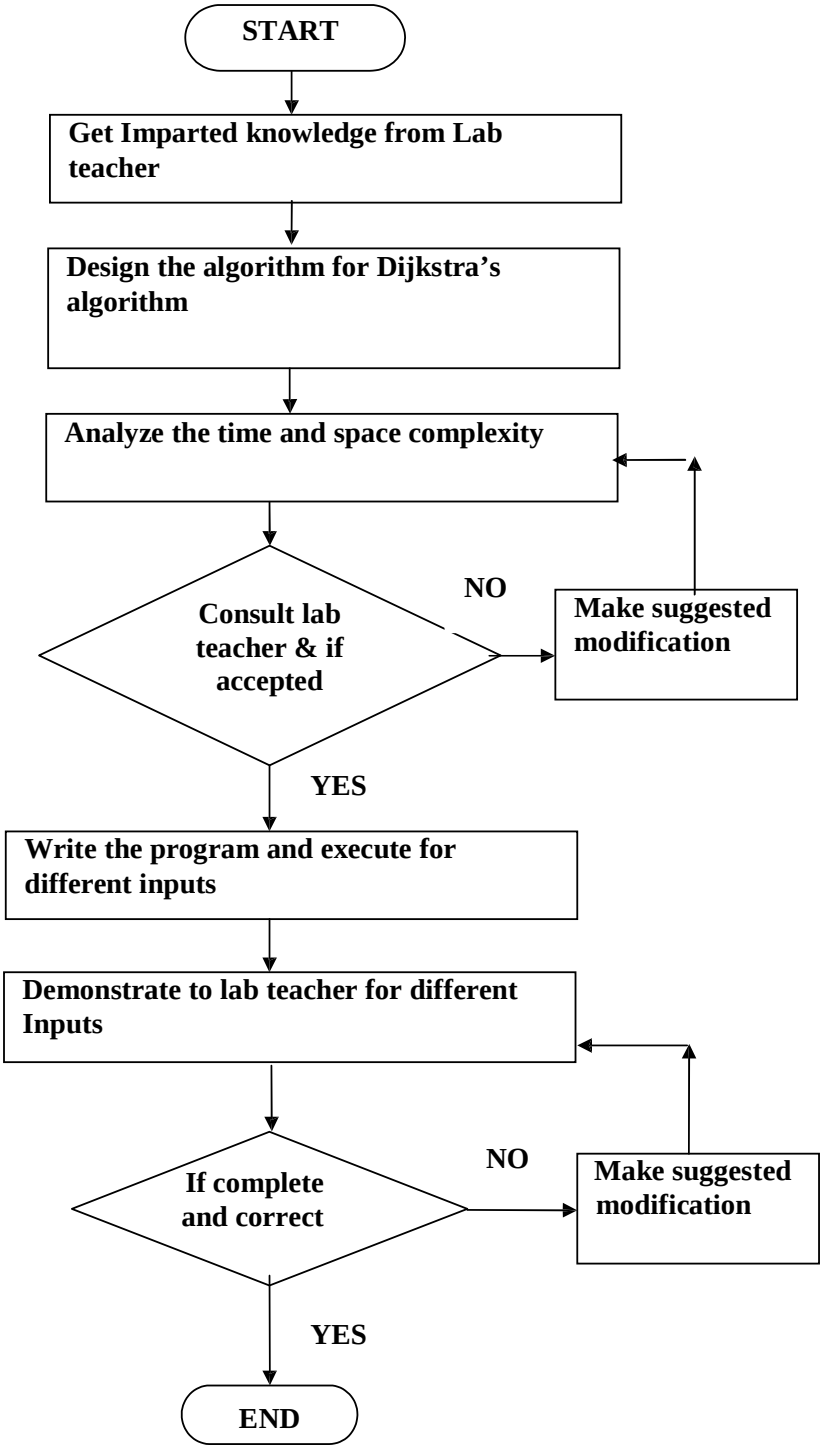
Revised On: 25/11/2013

TITLE	**MINIMUM SPANNING TREE ALGORITHMS**
PROBLEM STATEMENT /DEFINITION	**Represent any real world graph using adjacency list/adjacency matrix.. Fi minimum spanning tree using Prim's or Kruskal's algorithm.**
OBJECTIVE	To understand the concept of Graph To understand the concept of MST using Prim's & Krushkal's algorithm.
S/W PACKAGES AND HARDWARE APPARATUS USED	Linux OS: Fedora/ubantu, Eclipse/Codeblock like tools PC with the configuration as Pentium IV 1.7 GHz. 128M.B RAM, 40 G.B HDD, 15''Color Monitor, Keyboard, Mouse
REFERENCES	• "Fundamentals of Data Structures in C", E. Horowitz , S.Sahani • "An Introduction to Data Structures with Applications", Jean Paul Tremblay, Paul G. Sorenson. • "Data Structures and Algorithm Analysis in C++", M. Weiss
STEPS	Refer to student activity flow chart
INSTRUCTIONS FOR WRITING JOURNAL	• Title • Theory for Shortest Path • Prim's & Krushkal's algorithm for MST • Analysis of above for time and space complexity • Program code • Output for different inputs. • Conclusion

Subject coordinator **HODIT**

(Prof. Nisha R. Sodha) **(Dr. Emmanuel M.)**

STUDENT ACTIVITY CHART



Revised On: 25/11/2013

TITLE	**DIJKSTRA'S ALGORITHM**
PROBLEM STATEMENT /DEFINITION	**Represent a given graph using adjacency matrix/adjacency list and find the shortest path using Dijkstra's algorithm.**
OBJECTIVE	To understand the concept of Shortest Path To understand the concept of Dijkstra's algorithm.
S/W PACKAGES AND HARDWARE APPARATUS USED	Linux OS: Fedora/ubantu, Eclipse/Codeblock PC with the configuration as Pentium IV 1.7 GHz. 128M.B RAM, 40 G.B HDD, 15''Color Monitor, Keyboard, Mouse
REFERENCES	• "Fundamentals of Data Structures in C", E. Horowitz , S.Sahani "An Introduction to Data Structures with Applications", Jean Paul Tremblay, Paul G. Sorenson. • "Data Structures and Algorithm Analysis in C++", M. Weiss
STEPS	Refer to student activity flow chart
INSTRUCTIONS FOR WRITING JOURNAL	• Title • Theory for Shortest Path • Dijkstra's algorithm • Analysis of above for time and space complexity • Program code • Output for different inputs. • Conclusion

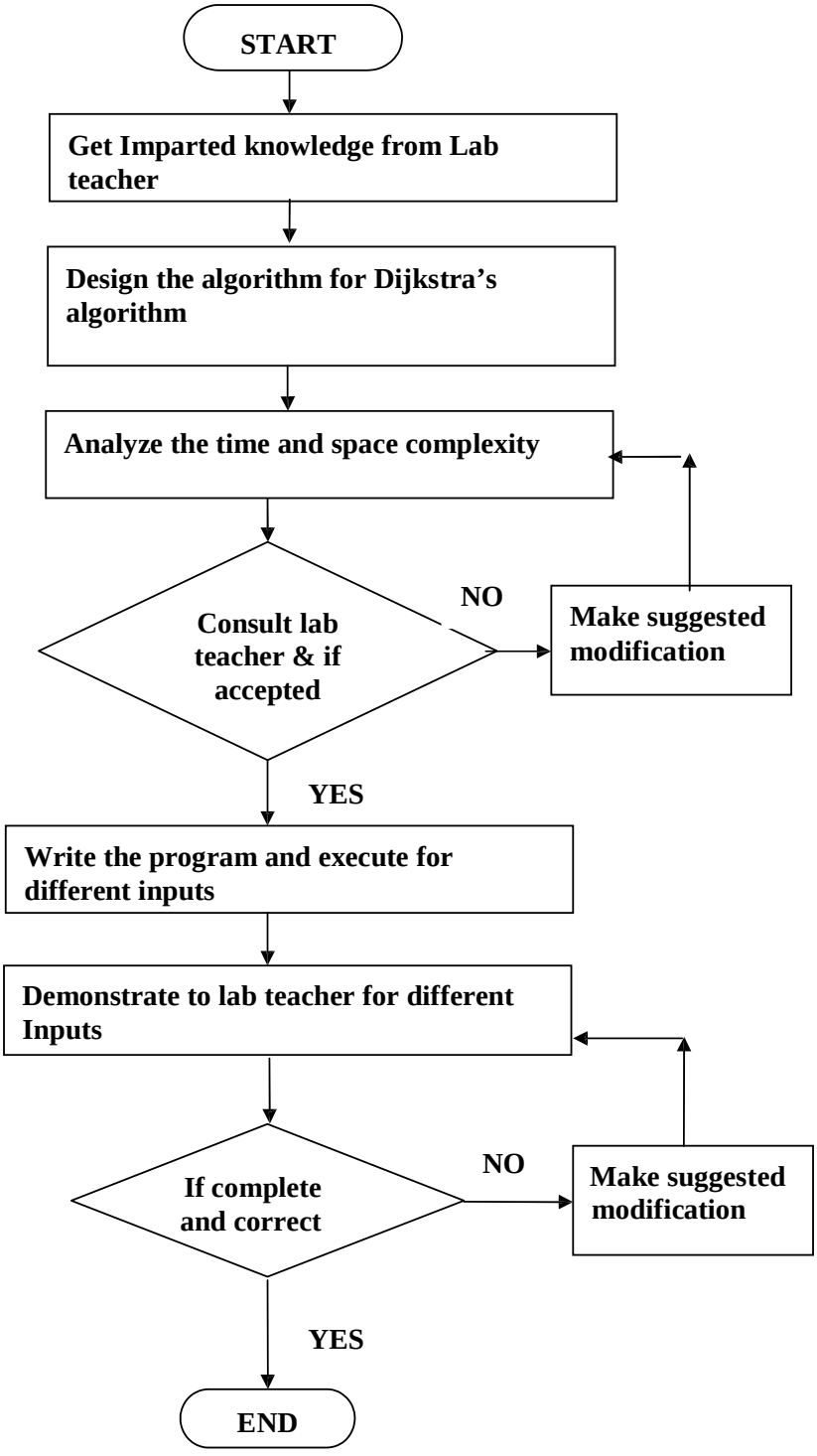
Subject coordinator

(Prof. Nisha R. Sodha)

HODIT

(Dr. Emmanuel M.)

STUDENT ACTIVITY CHART



Revised On: 25/11/2013

TITLE	**HASH TABLE**
PROBLEM STATEMENT	**Implementation of Hash table using array and handle collisions using Linear probing, chaining without replacement and Chaining with replacement.**
OBJECTIVE	To understand the concept of Hashing Different hashing techniques. To understand linear probing with replacement. To understand chaining without replacement
S/W PACKAGES AND HARDWARE APPARATUS USED	Linux OS: Fedora/ubantu, Eclipse/Codeblock PC with the configuration as Pentium IV 1.7 GHz. 128M.B RAM, 40 G.B HDD, 15''Color Monitor, Keyboard, Mouse
REFERENCES	• "Fundamentals of Data Structures in C", E. Horowitz , S.Sahani • "An Introduction to Data Structures with Applications", Jean Paul Tremblay, Paul G. Sorenson. • "Data Structures and Algorithm Analysis in C++", M. Weiss
STEPS	Refer to student activity flow chart
INSTRUCTIONS FOR WRITING JOURNAL	• Title • Problem Definition • Concept of Hashing • Algorithms for linear probing and chaining. • Analysis of above for time and space complexity • Program code • Output for different i/p comparing time complexity • Conclusion

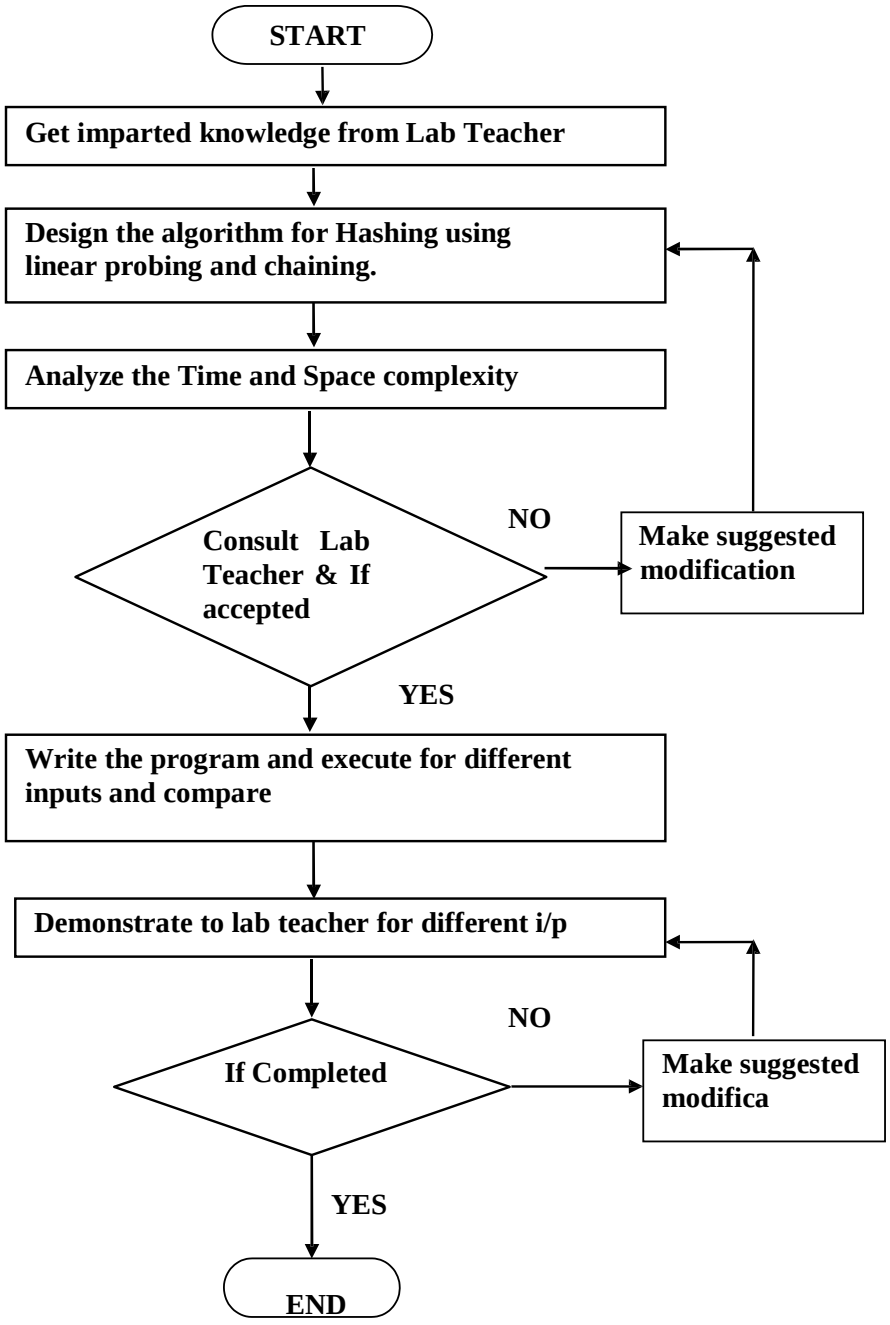
Subject coordinator

(Prof. Nisha R. Sodha)

HODIT

(Dr. Emmanuel M.)

STUDENT ACTIVITY CHART



Revised On: 25/11/2013

TITLE	**HEAP SORT ALGORITHM**
PROBLEM STATEMENT	**Implement Heap sort by constructing max or min heap.**
OBJECTIVE	To understand the concept of Heap Tree. To understand application of Heap data structure tree.
S/W PACKAGES AND HARDWARE APPARATUS USED	Linux OS: Fedora/ubantu, Eclipse/Codeblock like tools PC with the configuration as Pentium IV 1.7 GHz. 128M.B RAM, 40 G.B HDD, 15''Color Monitor, Keyboard, Mouse
REFERENCES	• "Fundamentals of Data Structures in C", E. Horowitz , S.Sahani • "An Introduction to Data Structures with Applications", Jean Paul Tremblay, Paul G. Sorenson. • "Data Structures and Algorithm Analysis in C++", M. Weiss
STEPS	Refer to student activity flow chart
INSTRUCTIONS FOR WRITING JOURNAL	• Title • Problem Definition • Concept of max and min heap tree. • Algorithms for Heap Tree creation and coding. • Analysis of above for time and space complexity • Program code • Output for different I/P comparing time complexity • Conclusion

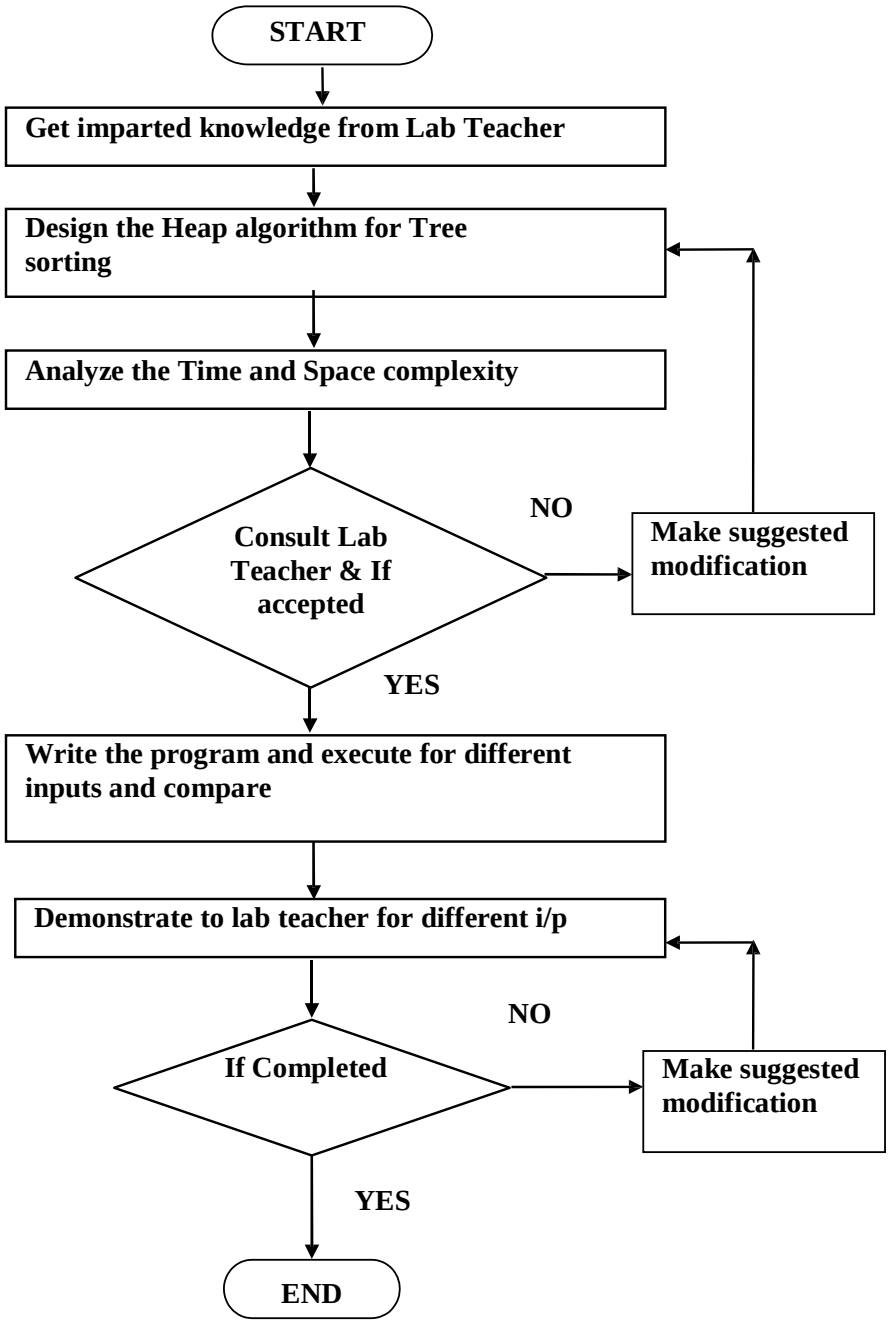
Subject coordinator

(Prof. Nisha R. Sodha)

HODIT

(Dr. Emmanuel M.)

STUDENT ACTIVITY CHART



Revised On: 25/11/2013

TITLE	**FILE PRIMITIVE OPERATIONS**
PROBLEM STATEMENT	**Implement an index sequential file for any Database and perform following operations on it** **i) Create Database** **ii) Display Database** **iii) Add a record** **iv) Delete a record** **v) Modify a record**
OBJECTIVE	To understand the concept of Index sequential files and various primitive operations possible on them.
S/W PACKAGES AND HARDWARE APPARATUS USED	Windows 2000, Turbo C++, PC with the configuration as Pentium IV 1.7 GHz. 128M.B RAM, 40 G.B HDD, 15''Color Monitor, Keyboard, Mouse
REFERENCES	• "Fundamentals of Data Structures in C", E. Horowitz , S.Sahani • "An Introduction to Data Structures with Applications", Jean Paul Tremblay, Paul G. Sorenson. • "Data Structures and Algorithm Analysis in C++", M. Weiss
STEPS	Refer to student activity flow chart
INSTRUCTIONS FOR WRITING JOURNAL	• Title • Problem Definition • Concept of Index sequential files. • Algorithms for add records, delete records, search records, modify records in Index sequential files. • Analysis of above for time and space complexity • Program code • Output for different i/p comparing time complexity • Conclusion

Subject coordinator **HODIT**

(Prof. Nisha R. Sodha) **(Dr. Emmanuel M.)**

STUDENT ACTIVITY CHART