EC-253	DATA STRUCTURES THROUGH C++ LAB	L	T	P	C
		-	-	3	2

COURSE OBJECTIVES:

1. To understand Object Oriented Programing features of C++.
2. To understand the concepts of encapsulation and compile time polymorphism.
3. To understand the concepts of inheritance, Runtime polymorphism and Templates.
4. To understand the concepts of Lists, Stacks and Queue ADTâ€™s.
5. To understand Binary trees and ADT's of BST and Various sorting techniques.

COURSE OUTCOMES:

After successful completion of the course, the students are able to

1. demonstrate about the basic Object-Oriented features of C++.
2. Experiment Implement about the concepts of encapsulation and compile time polymorphism.
3. Apply the concepts of Inheritance, Runtime polymorphism and Templates.
4. Implement Lists, Stacks and Queue ADTs and Handle BST ADT and different sorting algorithms.

List of Experiments:

1. Create a class HUGEINT by which we would be able to use much wider range of integers.Perform addition operation on two HUGEINTs.
2. Create a class TIME with appropriate data members to represent TIME. Construct a class implementation section to compare two TIMEs, to increment TIME by one second, to decrement TIME by one second and appropriate constructors to create TIME objects.
3. Write a class declaration for DATE and allow the operations to find nextday(), previousday(), leapyear(), compare() with appropriate constructors and destructors.
4. Create a user defined datatype STRING, allow possible operations by overloading (Relational operators,[], ( ), >, =).
5. Define RATIONAL class. Allow possible operations on RATIONALs by overloading operators(Arithmetic, Unary operators,>).
6. Program to implement (a) Single inheritance (b) Multiple inheritance (c) Hierarchical inheritance (d) Multipath inheritance.
7. Program to implement (a) runtime polymorphism (b) abstract base class concept.
8. Program to implement operations on single linked list.
9. Program to implement operations on doubly linked list.
10. Program to implement stack operations using arrays(with class templates) and linked lists.
11. Program to implement Queue operations using arrays and linked list.
12. Program to sort n elements using a) Merge Sort (with function templates). b)Quick Sort. c) Heap Sort.
13. Program to demonstrate BST ADT.

Note: A minimum of 10(Ten) experiments have to be performed and recorded by the candidate to attain eligibility for Semester End Practical Examination.