National University of Computer and Emerging Sciences

Laboratory Manual

for

Data Structures Lab

Course Instructor	Ms. Zareen Alamgir
Lab Instructor(s)	Ms. Fariha Maqbool Ms.Humna Shabir
Section	BCS-3F
Semester	Fall 2022

Department of Computer Science

FAST-NU, Lahore, Pakistan

Objectives:

In this lab, students will practice:

1. Revision of Templates
2. Revision of Pointers
3. Searching and Sorting

Function templates are special functions that can operate with generic types. This allows us to create a function template whose functionality can be adapted to more than one variable type or class without repeating the code for each type. This is achieved through template parameters. A template parameter is a special kind of parameter that can be used to pass a type as parameter. These function templates can use these parameters as if they were regular types. The format for declaring function templates with type parameters is:

template <class identifier> function_declaration;

While defining a function template the body of the function definition is preceded by a statement **template <class identifier>.** The identifier can then be used as the data type of the parameters, the return type of the function, the data type of local variables and/or the data types of parameters.

Task:

Implement a Template Class called **SimpleVector** which has following data members and functions:

1. A pointer to point to an integer array
2. An integer variable to store the array size
1. Default constructor to initialize the array pointer and array size to 0
3. Parameterized constructor that sets the size of the array and allocates memory for it. It should initialize all array elements to zero.
4. Copy constructor for **SimpleVector** class
5. Destructor to deallocate the memory of array
2. A findItem function that takes an item as argument and search the item from the array using linear search
3. A sortArray function that sorts the elements of array in ascending order. (Use sorting algorithm of your choice)
4. = operator which should make a deep copy.
5. [] operator which should set the element at index to the value specified in parameter. If the index is out of bound then return false otherwise return true.

Provide a sufficient main program that tests all of the above functions. (including destructors too).