University of Computer and Emerging Sciences

Lab Manual 01

for

Data Structures Lab

Course Instructor	Dr. Saira Karim
Lab Instructor (s)	Ms. Fariha Maqbool Ms. Humna Shabir
Section	BCS-3E
Semester	Fall 2022

Department of Computer Science

FAST-NU, Lahore, Pakistan

Objectives:

The objective of this lab is to revise following topics :

- ✓ Templates
- ✓ Pointers

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.

Exercise: Template Practice

Consider the definition of the following function template:

```
template <class type>
type surprise(type x, type y)
{
        return x + y;
}
```

Understand the code. What is the output of the following statements? Make a .cpp file to execute this code and add the output as comments in this file.

1. cout << surprise(5, 7) << endl;
2. string str1 = "Sunny";
   string str2 = " Day";
   cout << surprise(str1, str2) << endl;

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
3. Default constructor to initialize the array pointer and array size to 0
4. Parameterized constructor that sets the size of the array and allocates memory for it. It should initialize all array elements to zero.
5. Copy constructor for SimpleVector class (To make Deep copy)
6. Destructor to deallocate the memory of array
7. Getter and setter functions for data members
8. A **findItem** function that takes an item as argument and search the item from the array using linear search
9. A **sortArray** function that sorts the elements of array in ascending order. (Use sorting algorithm of your choice)
10. + operator that takes an integer variable as parameter. This method should add one integer to the array and increase the number of current elements. If the allocated size is not sufficient then increase the capacity of the array by double. Copy the old array in the new one (including the new element) and delete the old array.
11. – operator that takes no parameters. This method should delete the last added value from the array and change the current size. (note no memory allocation or deallocation will take place here)

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