University of Computer and Emerging Sciences

Laboratory Manual

for

Data Structures Lab

Course Instructor	Ms. Abeeda Akram
Lab Instructors	Sohaib Ahmad Ammara Nasir
Section	BCS-3A
Semester	Fall 2022

Department of Computer Science

FAST-NU, Lahore, Pakistan

Course Policies

1. After 5 minutes you will be marked late.
2. 80% attendance.
3. Discussion is not allowed during the lab.
4. Quizzes will be announced and unannounced.
5. No makeup for missed quiz.
6. Mobile phone usage within class will result in you being marked absent.
7. Labs have to be evaluated within the lab time, any submissions afterwards will not be accepted.

NOTE

Zero Tolerance for Plagiarism. Penalty will be given in accordance with the severity of plagiarism. This also includes forwarding the case to the DC Committee.

Objective of this lab:

After performing this lab, below mentioned concepts would be revised

1. Multidimensional Dynamic Arrays

Instructions:

• Make a separate project for each task.
• Indent your code properly.
• Use meaningful variable and function names. Follow the naming conventions.
• Use meaningful prompt lines and labels for all input/output.
• Make sure that there are NO dangling pointers or memory leaks in your program.

Question

C++ does not provide dynamic multidimensional arrays.
Develop a template class ndArray to provide such a facility. Note that the array should hold elements of any type in one linear array, in row major order, and should be able to provide any number of dimensions for users. Your ndArray- ADT should provide following functions:

1. **Insert**: Add an element at a given index.
2. **Delete**: Search and remove an element from an array at a given index.
3. **Extend**: Double the array in size.
4. **Shrink**: Reduce the array size to half.
5. **Subscript operator**: You will overload operator for (1 to 3) dimensions.
6. **Retrieve**: Search and return any element from the array.
7. **Print**: by overloading stream insertion operator (<<): Prints all elements of array according to dimensions. (Up till 3 rd dimensions print in table form, for higher dimensions just print linear data).
8. **Constructor**: Take dimensions of multi-D array.
9. **Destructor**
10. **Copy Constructor**

Hint: You can add an extra array to store dimensions.

```
class ndArray
{
        private:
                T* ptr;
                int totalDimensions;
                int* dimensions;
                int memoryLength;
                int* weights;
        public:
                ndArray(int* dimensions, int totalDimensions);
                ndArray<T> (const ndArray& obj);
                void Insert(int value, int *location);
                void Delete(int value);
                void Shrink();
                void Extend();
                T& operator()(int* location);
                T& operator[] (int i);
                T& operator()(int i, int j);
                T& operator()(int i, int j, int k);
                friend ostream& operator<<(ostream& out,const ndArray<T> arr);
                ~ndArray();
 };
```