**National University of Computer and Emerging Sciences **

**Lab Manual**

**Object Oriented Programming**

Course Instructor Ms. Saira karim

Lab Instructor (s) Ms. Mamoona Akbar

Ms. Sonia Anum

Section BSCS 2B

Semester Spring 2022

Department of Computer Science

FAST-NU, Lahore, Pakistan

Objectives

After performing this lab, students shall be able to:

∙ virtual Destructor

∙ Abstract Classes

∙ Templates

**TASK 1: (Hint:** Virtual Destructor**)**

Write a program to practice memory management alongside polymorphism. You are not allowed to change function prototypes

Implement following class structure. In addition to this you are to implement destructors in all classes below to ensure dynamically allocated memory is properly deleted.

| Person (Base Class) | Employee (Derived) | Student (Derived) |
| --- | --- | --- |
| //member variables  {  **Private:**  String \*fullName;  Int \*height;  **Public:**  **Person(string name,Int height)** //constructor  **Virtual void printInfo();(1)** //this function is to print all private variables  //destructor to be  implemented alongside type of class eg cout<<”person destructor” (5)  } | //member variables  {  **Private:**  String \*departement;  Int \*ID;  **Public:**  **Employee(string name,Int height,string**  **departement,Int id) :**  **Person( name, height)** //constructor  **void printInfo();**  //this function is to print all private variables alongside type of class  //destructor to be  implemented alongside type of class eg  cout<<”employee  destructor”  } | //member variables  {  **Private:**  String \*schoolName;  **Public:**  Student **(string name,Int height, string**  **SchoolName) : Person( name, height)**  //constructor  **void printInfo();**  //this function is to print all private variables alongside type of class  //destructor to be  implemented alongside type of class eg cout<<”student destructor”  } |

Main Program:

1. Create an array of base class pointers of size 2. (2d Array)

2. Initialize each of the base class pointer with employee and student object respectively. 3. Run a loop to call printInfo on the array created.

4. Call delete operator on the array of base class to test the memory management.

**TASK 2:**

Create a Template class **Calculator** that have following data member

∙ Num1

∙ Num2

∙ Result

Following are the functionalities that you perform.

∙ Constructor

∙ Parameterize constructor

∙ Sum() : that takes two number as argument and store result in Result variable. ∙ Sum() : that takes three number as argument and store result in Result variable. ∙ Subtract() : that takes two number as argument and store result in Result variable. ∙ Subtract() : that takes three number as argument and store result in Result variable. ∙ Multiplication():that takes two number as argument and store result in Result variable. ∙ Multiplication():that takes three number as argument and store result in Result variable. ∙ Division():that takes two number as argument and store result in Result variable. ∙ Division():that takes three number as argument and store result in Result variable. ∙ Modulus():that takes two number as argument and store result in Result variable. ∙ Max():that takes two number as argument and store result in Result variable ∙ Min():that takes two number as argument and store result in Result variable

**TASK 3:**

We have to calculate the area of a rectangle, a square and a circle. Create an abstract class 'Shape' with three abstract methods namely 'RectangleArea' taking two parameters, 'SquareArea' and 'CircleArea' taking one parameter each. The parameters of 'RectangleArea' are its length and breadth, that of 'SquareArea' is its side and that of 'CircleArea' is its radius. Now create another class 'Area' containing all the three methods 'RectangleArea', 'SquareArea' and 'CircleArea' for printing the area of rectangle, square and circle respectively. Create an object of class 'Area' and call all the three methods.