Object Oriented Programming Lab

Spring2024

FAST National University of
Computer and Emerging Sciences

Learning Outcomes

In this lab you are expected to learn the following:

- Polymorphism

C++ Runtime Polymorphism Example

Let's see a simple example of run time polymorphism in C++.

// an example without the virtual keyword.

```
#include <iostream>
using namespace std;
class Animal {
   public:
void eat(){
cout<<"Eating...";
   }
};
class Dog: public Animal
{
 public:
 void eat()
   {        cout<<"Eating bread...";
   }
};
int main(void) {
  Dog d = Dog();
  d.eat();
  return 0;
}
```

Output:

```
Eating bread...
```

Question 1

Write a program to calculate the area of following shapes by using ***Public -- Single Inheritance***. The *base class* is "shape" and the *derived classes* are rectangle, triangle and circle. Attributes of all the classes are as under:

shape	rectangle	triangle	circle

protected:
string type;
public:
Virtual void area()
{
//definition
}
	public:
	void area ()
	{
	//definition
	}
	private:
	float height;
	float width;
		public:
		void area ()
		{
		//definition
		}
		private:
		float base;
		float height;
			public:
			void area ()
			{
			//definition
			}
			private:
			float radius;

Area = 0	Area = width* height	Area = 1/2 of the base X the height	$A = \pi r^2$

- Your classes must have default constructor and parameterized constructor (see submission file)
- Provide a display function in Shape
- Provide implementation of display function for all classes, in Shape Class, as the function Display the value of type as "Shape". In Rectangle the Display function should display

```
cout<<"Type : "<<type;
cout<<"Width :"<<width;
cout<<"Height :"<<height;
```

- Similarly provide the implementation of function display for all rest of classes according to their member functions.
- Instantiate Shape class and print its functions.

- Similarly instantiate all child classes.
- Now call the area function for each child class to compute area.
- Call the display function as well.

Question 2

Multilevel Inheritance Overriding

Multiple inheritances enable a derived class to inherit members from more than one parent. Here base classes are **Person** and **Employee**, Derived class is **Faculty**. Attributes are as under:

Person (Base Class)	**Employee (derived from Person)**	**Faculty (Derived from Employee)**
protected: char name[10]; char address[10];	**protected:** int Emp_no; float gross_pay; float house_rent; float medical_allow; float net_pay; virtual void calcSalary( )	**protected:** char designation[10]; char department[10]; virtual void calcSalary( )

Use the formula below to calculate net pay::

- *House rent is 45%.*
- *Medical Allowance is 5%.*

*Formula to calculate net_pay= gross_pay – ((45/100)*gross_pay – (5/100)*gross_pay)*

- Write default and parameterized constructors to initialize attributes of all classes.
- Write a function calcSalary for calculating netpay in Employee class
- Override calcSalary in Faculty class.
- Create an object of class “faculty” in main by using parameterized constructor.
- Calculate salary for the instance of the faculty class you created in the previous step.