National University of Computer and Emerging Sciences

Lab Manual 12
Object Oriented Programming

Course Instructor	Mr. Usama Hassan
Lab Instructor (s)	Ms. Fariha Maqbool Mr. Sohaib Ahmad
Section	BSE-2C
Semester	Spring 2023

Department of Computer Science
FAST-NU, Lahore, Pakistan

Objectives

After performing this lab, students shall be able to:

- ✔ Abstract classes
- ✔ Pure Virtual Functions
- ✔ Pure Virtual Functions

Exercise 1: Defining an Abstract Class

Create a new project in your favorite C++ development environment.
Define an abstract class named "Shape" with the following code:

```
class Shape
{
public:
    virtual double area() = 0;
    virtual double perimeter() = 0;
};
```

The Shape class contains two pure virtual functions: "area" and "perimeter". These functions will be overridden in derived classes to calculate the area and perimeter of different shapes.

Compile and run the code. Observe that you cannot create an instance of the Shape class because it contains pure virtual functions.

Exercise 2: Deriving from an Abstract Class

Create a new class named "Rectangle" that inherits from the Shape class.

Override the "area" and "perimeter" functions in the Rectangle class to calculate and return the area and perimeter of a rectangle. The functions should take two arguments: the width and the height of the rectangle.

```
class Rectangle : public Shape
{
public:
    Rectangle(double width, double height) :
        width(width), height(height)
    {}

    double area() override
    {
        return width * height;
    }
```

```
    double perimeter() override
    {
        return 2 * (width + height);
    }

private:
    double width, height;
};
```

Compile and run the code. Observe that you can now create instances of the Rectangle class and call the "area" and "perimeter" functions on them.

Exercise 3: Creating an Array of Shapes

Create an array of Shape pointers with a size of 5.

```
Shape* shapes[5];
```

Allocate memory for five Rectangle objects and store the pointers to these objects in the array of Shape pointers.

```
for (int i = 0; i < 5; i++)
{
    shapes[i] = new Rectangle(i + 1, i + 2);
}
```

Loop through the array of Shape pointers and call the "area" and "perimeter" functions on each object. Observe that the correct implementation of the functions is called for each object, even though they are all stored as Shape pointers.

```
for (int i = 0; i < 5; i++)
{
    cout << "Shape " << i << ": ";
    cout << "Area = " << shapes[i]->area();
    cout << ", Perimeter = " << shapes[i]->perimeter() << endl;
}
```

Exercise 4: Adding Another Derived Class

Create a new class named "Circle" that inherits from the Shape class.

Override the "area" and "perimeter" functions in the Circle class to calculate and return the area and circumference of a circle. The functions should take one argument: the radius of the circle.

```
class Circle : public Shape
{
public:
    Circle(double radius) :
        radius(radius)
    {}

    double area() override
    {
        return 3.14159265 * radius * radius;
    }

    double perimeter() override
    {
        return 2 * 3.14159265 * radius;
    }

private:
    double radius;
};
```

Add two Circle objects to the array of Shape pointers.

```
shapes[5] = new Circle(5);
shapes[6] = new Circle(10);
```

Loop through the array of Shape pointers and call the "area" and "perimeter" functions on each object. Observe that the correct implementation of the functions is called for each object, regardless of whether it is a Rectangle or a Circle.

```
for (int i = 5; i < 7; i++)
{
    cout << "Shape " << i << ": ";
    cout << "Area = " << shapes[i]->area();
    cout << ", Perimeter = " << shapes[i]->perimeter() << endl;
}
```

Bonus:

Question 1:
Define a class named "Book" that has two private variables named "title" and "author". The class should have a constructor that sets the title and author of the book, and two methods named "getTitle" and "getAuthor" that return the title and author of the book, respectively.

Question 2:
Define a class named "Movie" that has two private variables named "title" and "director". The class should have a constructor that sets the title and director of the movie, and two methods named "getTitle" and "getDirector" that return the title and director of the movie, respectively.

Question 3:
Define an abstract class named "Media" that has two pure virtual functions named "getType" and "getInfo". The class should also have a final method named "getTitle" that returns the title of the media.

Question 4:
Derive two classes from the "Media" class: "Book" and "Movie". The "Book" class should override the "getType" and "getInfo" functions to return "Book" and the title and author of the book, respectively. The "Movie" class should override the "getType" and "getInfo" functions to return "Movie" and the title and director of the movie, respectively.