CS-101 Programming Fundamentals

Lab Manual

Habib University

PUBLISHED BY HABIB UNIVERSITY

BOOK-WEBSITE.COM

First printing, July 2018

Contents

1. Abstract classes

1.1 Objective

In this lab, we will learn about a type of class called abstract classes and its usage and implementation.

1.2 Description

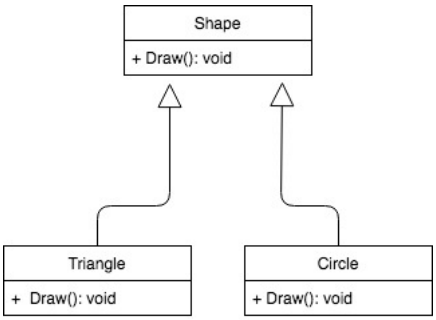

Figure 1.1: UML representation of abstract class

Sometimes implementation of all function cannot be provided in a base class because we don't know the implementation. Such a class is called **abstract class**. For example, let Shape be a base class. We cannot provide implementation of function draw() in Shape, but we know every derived class must have implementation of draw(). Similarly an Animal class doesn't have implementation of move() (assuming that all animals move), but all animals must know how to move. We cannot create objects of abstract classes. For this, we need to have at least one pure virtual function in the class.
A **pure virtual function** (or abstract function) in C++ is a virtual function for which we don't have

implementation, we only declare it. A pure virtual function is declared by assigning 0 in declaration. It's done by the syntax showed below.

```
// An abstract class
class Test
{
    // Data members of class
    public:
    // Pure Virtual Function
    virtual void show() = 0;

    /* Other members */
};
```

Let's see an example in whch a pure virtual function is declared in the abstract class and then defined in the derived class.

■ **Example 1.1**

```
class Base
{
    int x;
    public:
        virtual void func() = 0;
        int getX()
        {
            return x;
        }
};

// This class inherits from Base and implements func()
class Derived: public Base
{
    int y;
    public:
        void func()
        {
            cout << "func() called";
        }
};

int main()
{
    Derived d;
    d.func();
    return 0;
}
```

Now the abstract class has the attribute 'int x' and the function 'int GetX()'. This function will also be the part of derived class and is already defined in the base class. Now the function 'func()' is a pure virtual function which is defined in the base class and is implemented in the derived class. Now, if we create an instance of Derived class, we can access the functions 'GetX()' and 'func()'. The 'GetX()' will be called through the base class and 'func()' will be called through the derived class. The program above will output:

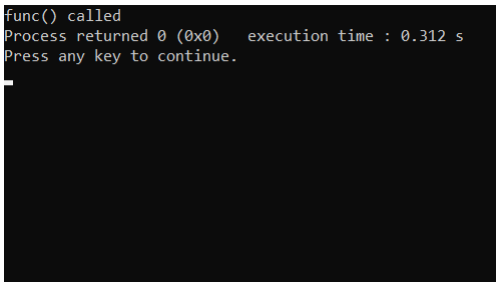

Figure 1.2: Output of implemented pure virtual function

■

The derived class can also become an abstract class if the pure virtual function declared in base class is not defined in the derived class.

■ **Example 1.2**

```
class Base
{
    public:
    virtual void show() = 0;
};

class Derived : public Base { };

int main(void)
{
    Derived d;
    return 0;
}
```

■

Corollary 1.2.1 — Interface vs Abstract classes. An interface does not have implementation of any of its methods, it can be considered as a collection of method declarations. In C++, an interface can be simulated by making all methods as pure virtual. In Java, there is a separate keyword for interface.

1.3 Problems

Problem 1.1 Make an abstract class of Shape with pure virtual function to calculate area. It should also have the attributes for the position of the Shape. The derived classes should be Square, Circle, Trapezium and Rectangle. All derived classes should implement the pure virtual function defined in the base class.

Instructions

- Make a struct for Position to have x and y coordinates.
- Define the attributes for dimensions according to the shape in derived classes

Problem 1.2 Make a base class with a pure virtual function. Derive a class from it and declare another pure virtual function in the derived class. Then derive an other class from the first derived class and declare a pure virtual function in it too. And then lastly derive another class from the second derived class, and define all the pure virtual functions in it

1.4 Feedback

Please write the things you've learned from this lab and suggestions to make it more better and easy to learn.