Object Oriented Programming Lab

SPRING - 2024

LAB 13

FAST National University of
Computer and Emerging Sciences

Learning Outcomes

In this lab you are expected to learn the following:

- Abstract Classes

Abstract Classes:

An **abstract class** is a class that is designed to be specifically used as a base class. An abstract class contains **at least one pure virtual function**. You declare a pure virtual function by using a pure specifier (= 0) in the declaration of a virtual member function in the class declaration.

Example:

```
class AbstractClass {
public:
   virtual void AbstractMemberFunction() = 0; // Pure virtual function
                                                                  // makes this class Abstract class.

   virtual void NonAbstractMemberFunction1(); // Virtual function.

   void NonAbstractMemberFunction2();
};
```

The main difference between '**virtual function'** and **'pure virtual function'** is that 'virtual function' **has its definition in the base class** and also the inheriting derived classes redefine it. The pure virtual function **has no definition in the base class**, and all the inheriting derived classes have to redefine it.

Be Careful !

Abstract class **cannot** be used as a parameter type, a function return type, and **not** to declare an object of an abstract class. It **can be** used to declare pointers and references to an abstract class.

Question 1

Suppose there are some employees working in a firm. The firm hires only two types of employee: either **driver** or **developer**

Now, you have to develop a software to store information about them. So, here is an idea there is no need to make objects of employee class. We will make objects to only driver or developer. Also, both must have some salary. So, there must be a common function to know about salary.

This need will be best accomplished with **abstract class**.

So, we can make 'Employee' an abstract class and 'Developer' and 'Driver' its subclasses.

```
class Employee                    //  abstract base class

{

virtual int getSalary() = 0;     // pure virtual function

};
```

Here **getSalary()** function in the class **Employee** is a pure virtual function. Since the Employee class contains this pure virtual function, therefore it is an **abstract base class**.
Since the abstract function is defined in the subclasses, therefore the function 'getSalary()' is defined in both the subclasses (**driver & developer**) of the class Employee.

Attribute for Employe class: (decide which attribute should be protected/private/public)

- int Emp_no

Attribute for driver class:

- Int salary

Attribute for developer class:

- Int salary

1. Write default and parameterized constructors to initialize attributes of all classes.
2. Make Employee class Abstract by declaring at least one pure virtual function getSalary() . You do not need to provide body for it as it is a pure virtual function and can only be implemented by child class of Employee. A pure virtual function is declared as below virtual float getSalary()=0;
3. Write a class driver, make it a child of Employee, declare its member function getsalary()
4. Write a class developer, make it a child of Employee, declare its member function getsalary()..
5. Since Employee class is abstract and cannot be instantiated, but we can create a pointer of it and make it point to the objects of child classes' one by one, i.e.
   Base* ptr=new child (1, 4, 6)

Similarly instantiate all child classes in **main ()**.

Submission Instructions:

1. Now create a new folder with name *ROLLNO_SEC_LAB01* **e.g. i22XXXX_A_LAB09**
2. You need to display your roll no and name before the output of each question.
3. Move all of your .**cpp and .h files** to this newly created directory and compress it into a **.zip file**.
4. Now you have to submit this zipped file on Google Classroom.
5. If you don't follow the above-mentioned submission instruction, you will be marked **zero.**
6. Plagiarism in the Lab Task will result in **zero** marks in the whole category.