OOP Experiment 12

Inheritance in C++

Class: SE **Div.:** C

Name: Jaiee Bhange **Roll No.:** SEETC307

Aim:

To understand the concept of inheritance and different characteristics of inheritance. To understand inheritance in an object-oriented system.

Objective:

- To understand the role of inheritance in daily life.
- To understand why inheritance is essential.
- To understand the implementation of inheritance and benefits.
- To understand the role of inheritance in code optimization.

PRE-TEST

1. C++ provides a special …… called the constructor, which enables an object to initialize itself when it is created.

+ A. friend function
+ B. member function
+ C. public function
+ D. private function

ANS: B. Member function

2. Constructors are normally used to …….. and to allocate memory.

+ A. define variables
+ B. allocate variables
+ C. initialize variables
+ D. initialize object

ANS: C. initialize variables

3. A constructor that accepts no parameters is called the …….

+ A. default constructor
+ B. parameterized constructor
+ C. implicit constructor
+ D. null constructor

ANS: A. default constructor

4.Constructors cannot be inherited, through a derived class can call the ……. constructor.

+ A. base class
+ B. derived class
+ C. void class
+ D. default class

ANS: A. base class

5.A ……. is used to declare and initialize an object from another object.

+ A. default constructor
+ B. default argument constructor
+ C. implicit constructor
+ D. copy constructor

ANS: D. copy constructor

Simulation:

```
#include<iostream.h>
#include<conio.h>
class base
{
public:
void display()
{
cout<<”base class”;
}
};
```

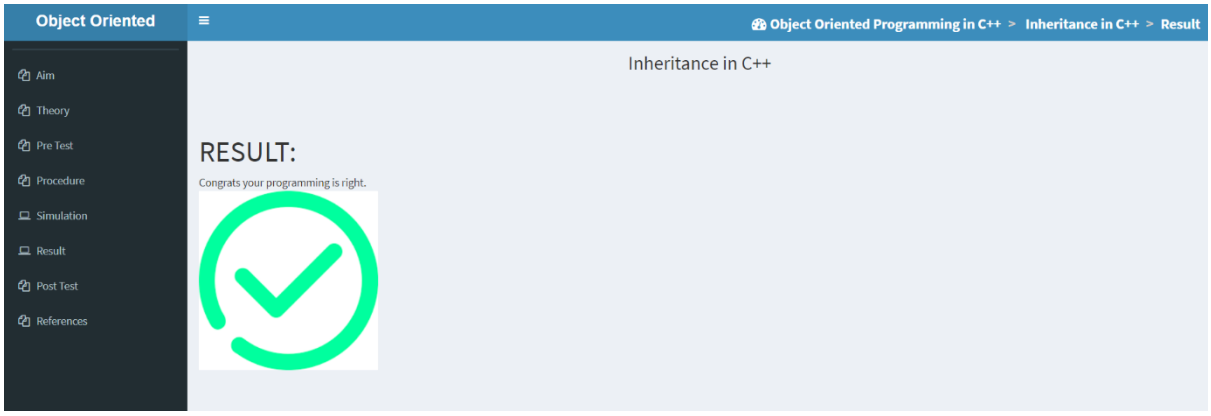
```
d : public base
```

```
{
public:
void display1()
{
cout<<”derived class”;
}
};
void main()
{
derived d1;
```

```
  d1.display();
```

```
d1.display1();
getch();
}
```

Result-



POST TEST

1. What does inheritance allows you to do?

Ans: Inheritance allows programmers to create classes that are built upon existing classes, to specify a new implementation while maintaining the same behaviors (realizing an interface), to reuse code and to independently extend original software via public classes and interfaces.

2. What is the syntax of inheritance of class?

Ans: The ": public base_class_name" is the essential syntax of inheritance; the function of this syntax is that the class will contain all public and protected variables of the base class.

3. Explain the difference between Overriding vs. overloading?

Ans: Difference between Overloading and Overriding is:-

1) Function Overloading happens in the same class when we declare same functions with different arguments in the same class. Function Overriding is happens in the child class when child class overrides parent class function.
2) In function overloading function signature should be different for all the overloaded functions. In function overriding the signature of both the functions (overriding function and overridden function) should be same.
3) Overloading happens at the compile time thats why it is also known as compile time polymorphism while overriding happens at run time which is why it is known as run time polymorphism.
4) In function overloading we can have any number of overloaded functions. In function overriding we can have only one overriding function in the child class.

4. Explain the term Polymorphism.

Ans: Polymorphism means "many forms", and it occurs when we have many classes that are related to each other by inheritance. Like we specified in the previous chapter; Inheritance lets us inherit attributes and methods from another class. Polymorphism uses those methods to perform different tasks.

1. Which among the following best describes the Inheritance?

A. Copying the code already written
B. Using the code already written once
C. Using already defined functions in programming language
D. Using the data and functions into derived segment

ANS: D. Using the data and functions into derived segment

2. Which among the following is correct for multiple inheritance?

A. class student{public: int marks;}s; class stream{int total;}; class topper:public student, public stream{ };
B. class student{int marks;}; class stream{ }; class topper: public student{ };
C. class student{int marks;}; class stream:public student{ };
D. class student{ }; class stream{ }; class topper{ };

ANS: A. class student{public: int marks;}s; class stream{int total;}; class topper:public student, public stream{ };

3. In Multipath inheritance, in order to remove duplicate set of records in child class, we _____ .

A. Write Virtual function in parent classes
B. Write virtual functions is base class
C. Make base class as virtual base class
D. All of these

ANS: C. Make base class as virtual base class

4. Output of following program?

```
#include <iostream>
using namespace std;
class Base1
   public:
     Base1()
     { cout << " Base1's constructor called" << endl; }
};
class Base2 {
public:
   Base2()
```

```
   { cout << "Base2's constructor called" << endl; }
};
class Derived: public Base1, public Base2 {
public:
  Derived()
  { cout << "Derived constructor called" << endl; }
};
int main()
{
  Derived d;
  return 0;
}
```

A. Compiler Dependent
B. Base1’s constructor called
Base2’s constructor called
Derived constructor called
C. Base2’s constructor called
Base1’s constructor called
Derived constructor called
D. Compiler Error

ANS: B. Base1’s constructor called
Base2’s constructor called
Derived constructor called

5. Can we pass parameters to base class constructor through derived class or derived class constructor?

A. Yes
B. No

ANS: A. Yes