Lab 1 after MID

Polymorphism in C++

Name : Fatima Fiaz

SAP ID : 45470

BS CS 2nd sem 2A

Subject : OOP

[Pick the date]

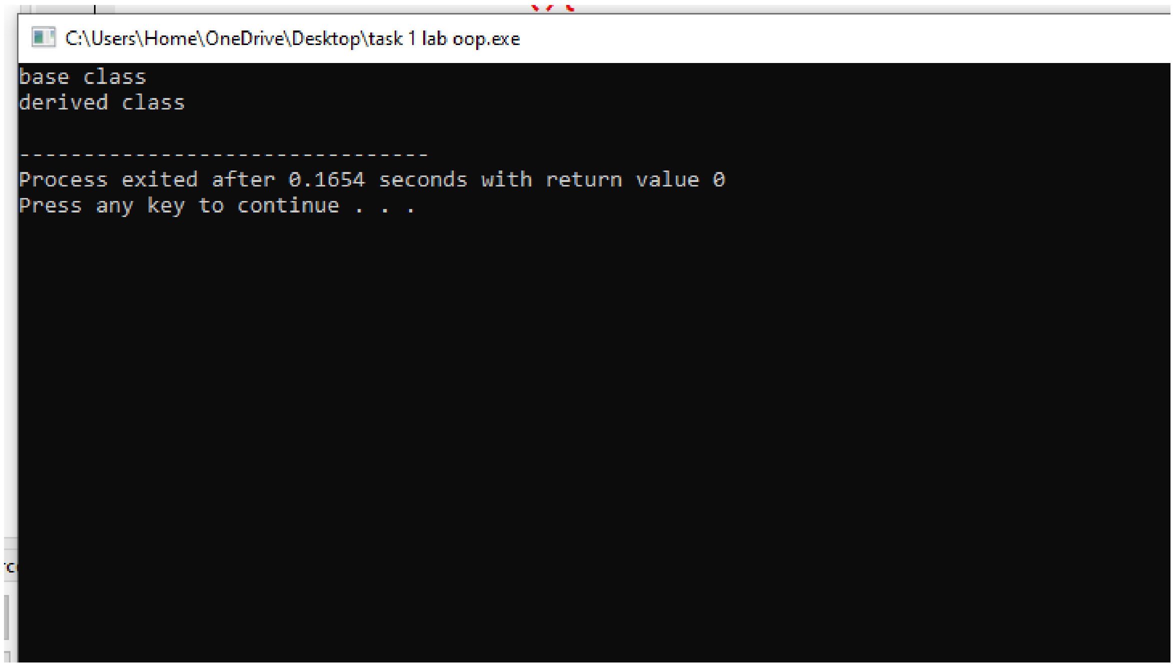
LAB TASK

On

Polymorphism in c++

- **Task 1**

  There is no error in this program , just we combine the various parts of the program and then complie it .after this , the code does not generate any error and complie successfully.

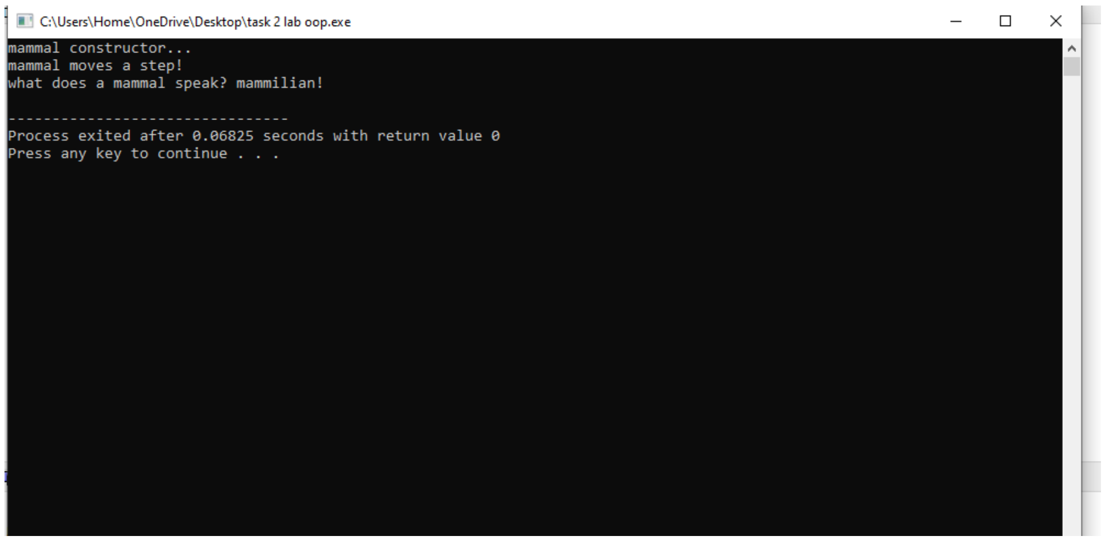


```
C:\Users\Home\OneDrive\Desktop\task 1 lab oop.exe
base class
derived class

--------------------------------
Process exited after 0.1654 seconds with return value 0
Press any key to continue . . .
```

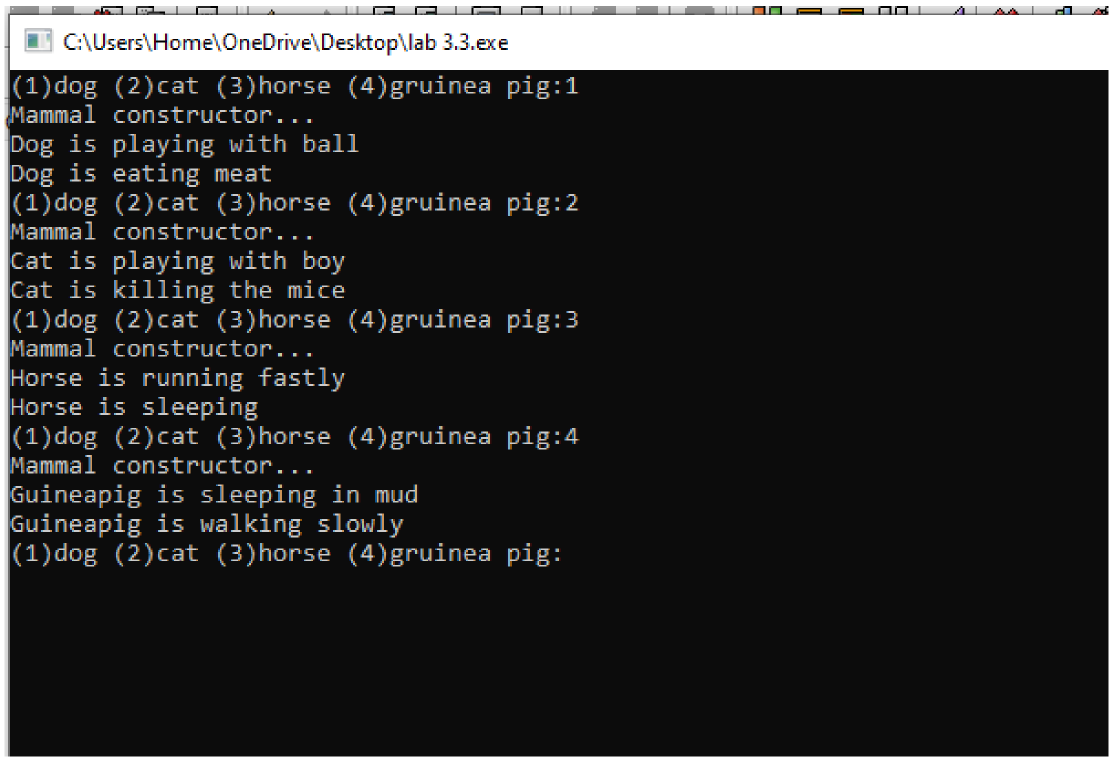
- **Task 2**

  After merging the program, in one file it was still generating the error. So did some changes in it by inheriting the class dog with public mammal and then it generate successfully .


```
mammal constructor...
mammal moves a step!
what does a mammal speak? mammilian!

--------------------------------
Process exited after 0.06825 seconds with return value 0
Press any key to continue . . .
```

- **<u>Task 3</u>**
  There was a just int main in this function which was not a complete program , so just wrote the remaining part of program and then compile it by adding some activities of animals.


```
(1)dog (2)cat (3)horse (4)gruinea pig:1
Mammal constructor...
Dog is playing with ball
Dog is eating meat
(1)dog (2)cat (3)horse (4)gruinea pig:2
Mammal constructor...
Cat is playing with boy
Cat is killing the mice
(1)dog (2)cat (3)horse (4)gruinea pig:3
Mammal constructor...
Horse is running fastly
Horse is sleeping
(1)dog (2)cat (3)horse (4)gruinea pig:4
Mammal constructor...
Guineapig is sleeping in mud
Guineapig is walking slowly
(1)dog (2)cat (3)horse (4)gruinea pig:
```

Questions & Answers

- **Are inherited members and functions passed along to subsequent generations? If Dog derives from Mammal, and Mammal derives from Animal, does Dog inherit Animal's functions and data?**

  Yes, in object-oriented programming language, when a class is derived from another class, the derived class inherits all the members and functions of the base class.
  So if Dog is derived from Mammal, and Mammal is derived from Animal, then Dog would inherit all the members and functions of Mammal.

- **If, in the example above, Mammal overrides a function in Animal, which does Dog get, the original or the overridden function?**
  If a mammal overrides a function in animal, dog is derived from mammal then the dog will not get an original function as derive class replace the implementation of base class and it gets overridden function.

- **Can a derived class make a public base function private?**
  No, it cannot make a public class function private. Private inheritance makes the public and protected members of the base class private in the derived class.

- **Why not make all class functions virtual?**
  Due to some some code complexities, design clarities , performances overhead . we do not make all class functions virtual.

- **If a function (SomeFunc()) is virtual in a base class and is also overloaded, so as to take either an integer or two integers, and the derived class overrides the form taking one integer, what is called when a pointer to a derived object calls the two-integer form?**

The overriding of the one-integer form hides the entire base class function, and thus you will get a compile error complaining that that function requires only one integer.it depends on how derived classes overrides the virtual function.

<u>Here are some more questions:</u>

1. **What is a v-table?**
   A v-table, or virtual function table, is a common way for compilers to manage virtual functions in C++.

2. **What is a virtual destructor?**
   Virtual Destructor in C++ is a member function that is used to free up the memory allocated used by the object of a child class or derived class when it is removed from the memory using the parent class pointer object. Virtual destructor in C++ is mainly responsible for resolving the memory leak problem.

3. **How do you show the declaration of a virtual constructor?**
   You can declare the function in the base class using the virtual keyword. Once you declare the function in the base class, you can use a pointer or reference to call the virtual class and execute its virtual version in the derived class.
4. **How can you create a virtual copy constructor?**
   The creation of a virtual constructor is not possible because There is no virtual memory table present while calling the constructor. So, the construction of a virtual constructor is not possible.

5. **How do you invoke a base member function from a derived class in which you've overridden that function?**
   We use the scope resolution operator :: . We can also access the overridden function by using a pointer of the base class to point to an object of the derived class and then calling the function from that pointer.

6. **How do you invoke a base member function from a derived class in which you have not overridden that function?**

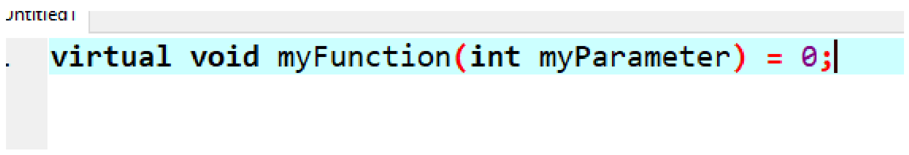
You have simply write a function as usual and call it .

7. **If a base class declares a function to be virtual, and a derived class does not use the term virtual when overriding that class, is it still virtual when inherited by a third-generation class?**
   Yes because the virtually is still inherited in it and it cannot be turned off.

8. **What is the protected keyword used for?**
   The protected keyword is an access modifier used for attributes, methods and constructors, making them accessible in the same package and subclasses.

<u>Some more exercises:</u>

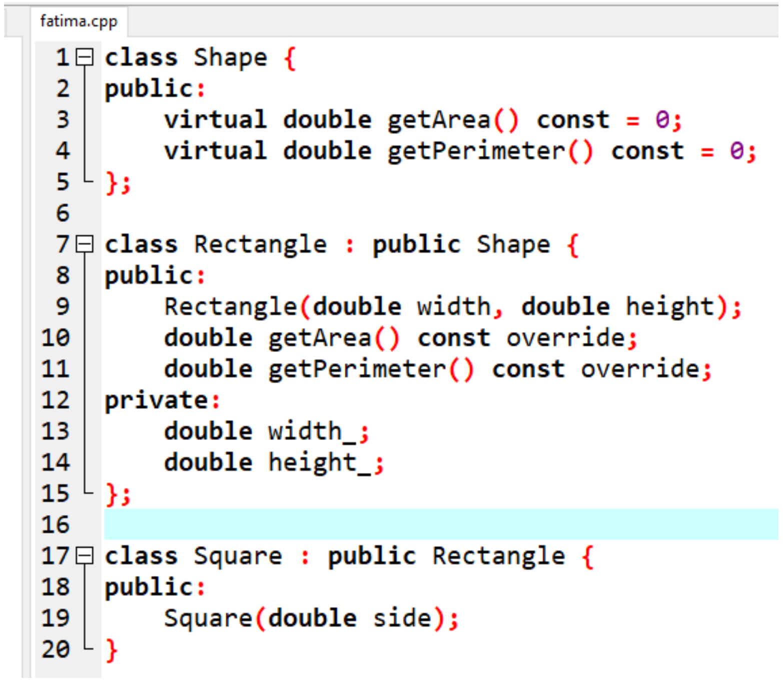
1. **Show the declaration of a virtual function that takes an integer parameter and returns void.**
   We should declare a virtual function by using a keyword of virtual that takes an integer parameter and returns void is as :



```
virtual void myFunction(int myParameter) = 0;
```

2. **Show the declaration of a class Square, which derives from Rectangle, which in turn derives from Shape.**

   The declaration of class square which derives from rectangle is :

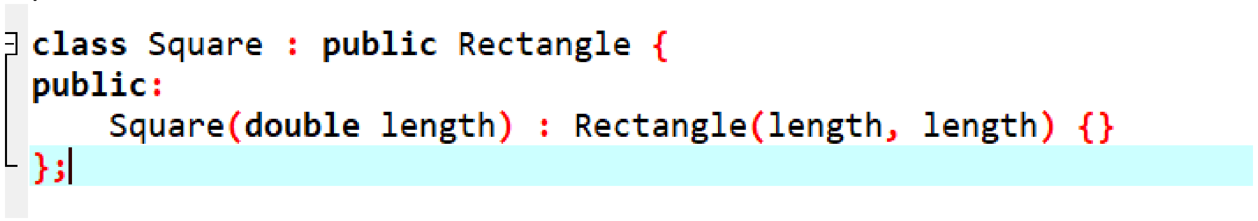

fatima.cpp

```cpp
class Shape {
public:
    virtual double getArea() const = 0;
    virtual double getPerimeter() const = 0;
};

class Rectangle : public Shape {
public:
    Rectangle(double width, double height);
    double getArea() const override;
    double getPerimeter() const override;
private:
    double width_;
    double height_;
};

class Square : public Rectangle {
public:
    Square(double side);
}
```

3. **If, in Exercise 2, Shape takes no parameters, Rectangle takes two (length and width), but Square takes only one (length), show the constructor initialization for Square.**
   When the shapes takes no parameter and Rectangle takes two lenghth nd width but square takes only one then the constructor initialize for square will.



```cpp
class Square : public Rectangle {
public:
    Square(double length) : Rectangle(length, length) {}
};
```

The upper part will remain same , just square part will change like this .

4. **Write a virtual copy constructor for the class Square (in Exercise 3).**
   It will show error because copy constructors cannot be virtual.

5. **BUG BUSTERS: What is wrong with this code snippet? void SomeFunction (Shape); Shape * pRect = new Rectangle; SomeFunction(*pRect);**

It will be written as :

```
void SomeFunction (Shape& Shape);
Shape * pRect = new Rectangle;
 SomeFunction(*pRect);
```

- Now someFunction is in parameter by reference.

6. **BUG BUSTERS: What is wrong with this code snippet? class Shape() { public: Shape(); virtual ~Shape(); virtual Shape(const Shape &); };**

   Copy constructor is not declared correctly . it has the same name as the class name and it has no return type.