Practical 10

Index Number – 22001212

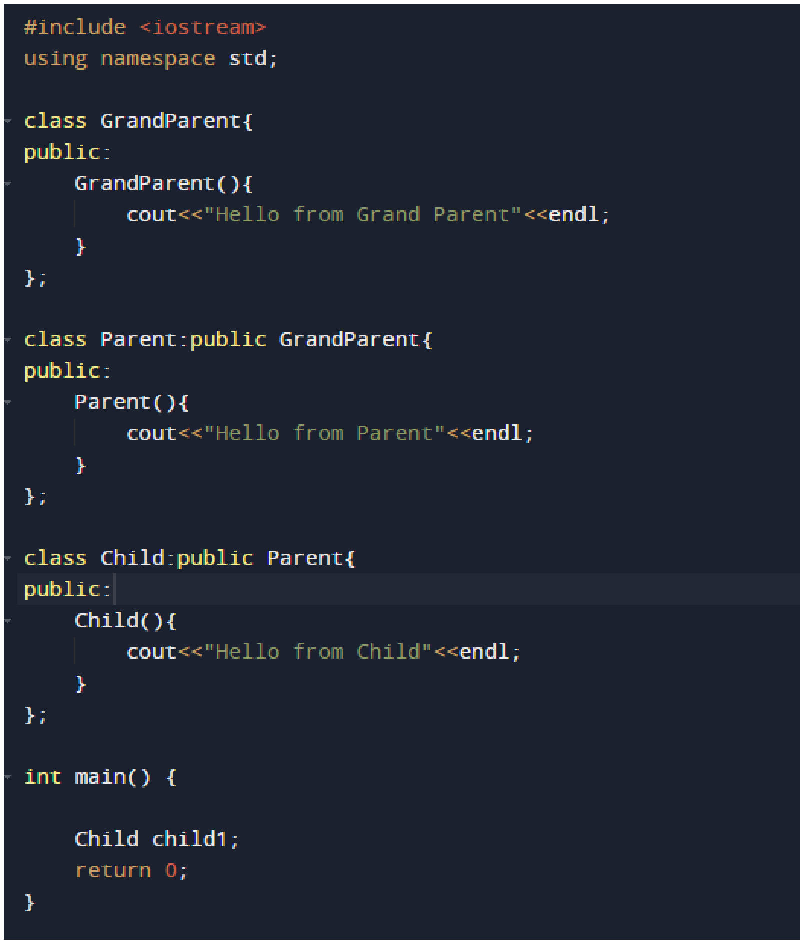
01) In the main function object ‘d’ which is from Derived class will be inherited by the Super classes ‘Base1’ and ‘Base2’. So it will give the following output.

Base1's constructor called

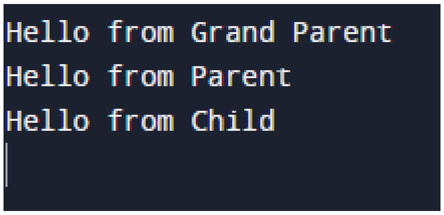
Base2's constructor called

Derived's constructor called

02)

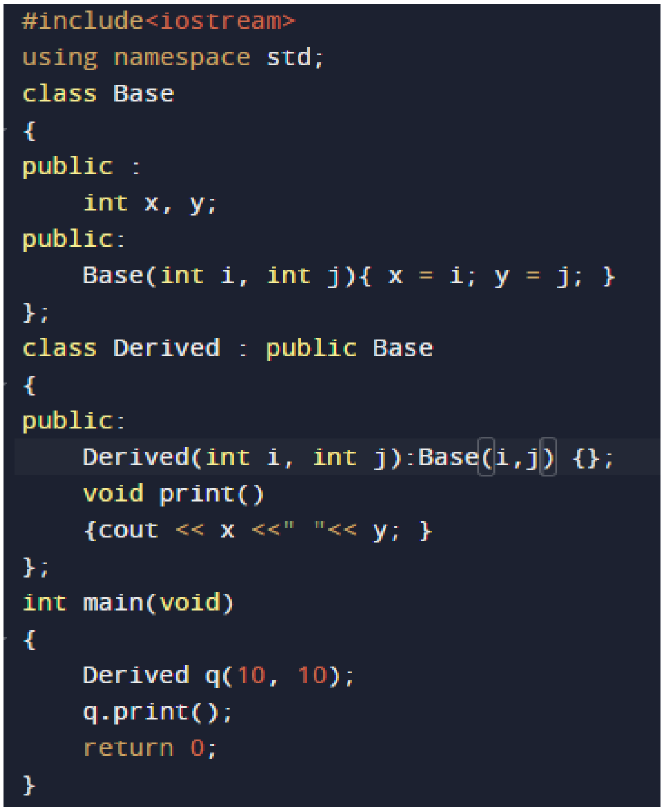


```cpp
#include <iostream>
using namespace std;

class GrandParent{
public:
    GrandParent(){
        cout<<"Hello from Grand Parent"<<endl;
    }
};

class Parent:public GrandParent{
public:
    Parent(){
        cout<<"Hello from Parent"<<endl;
    }
};

class Child:public Parent{
public:
    Child(){
        cout<<"Hello from Child"<<endl;
    }
};

int main() {

    Child child1;
    return 0;
}
```

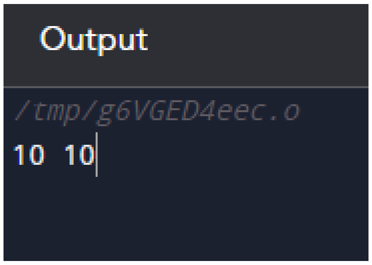


```
Hello from Grand Parent
Hello from Parent
Hello from Child
```

03)

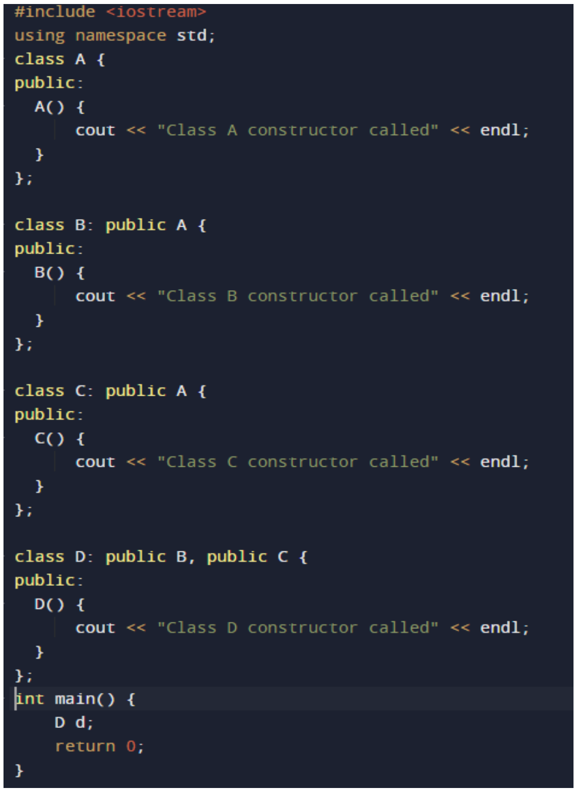


```cpp
#include<iostream>
using namespace std;
class Base
{
public :
    int x, y;
public:
    Base(int i, int j){ x = i; y = j; }
};
class Derived : public Base
{
public:
    Derived(int i, int j):Base(i,j) {};
    void print()
    {cout << x <<" "<< y; }
};
int main(void)
{
    Derived q(10, 10);
    q.print();
    return 0;
}
```

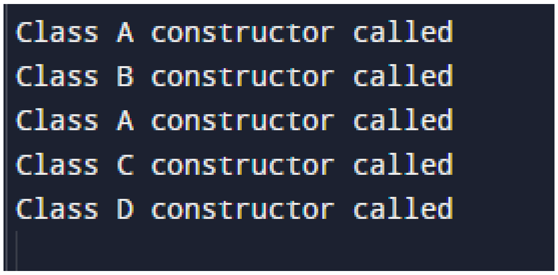


```
Output
/tmp/g6VGED4eec.o
10 10
```

04)



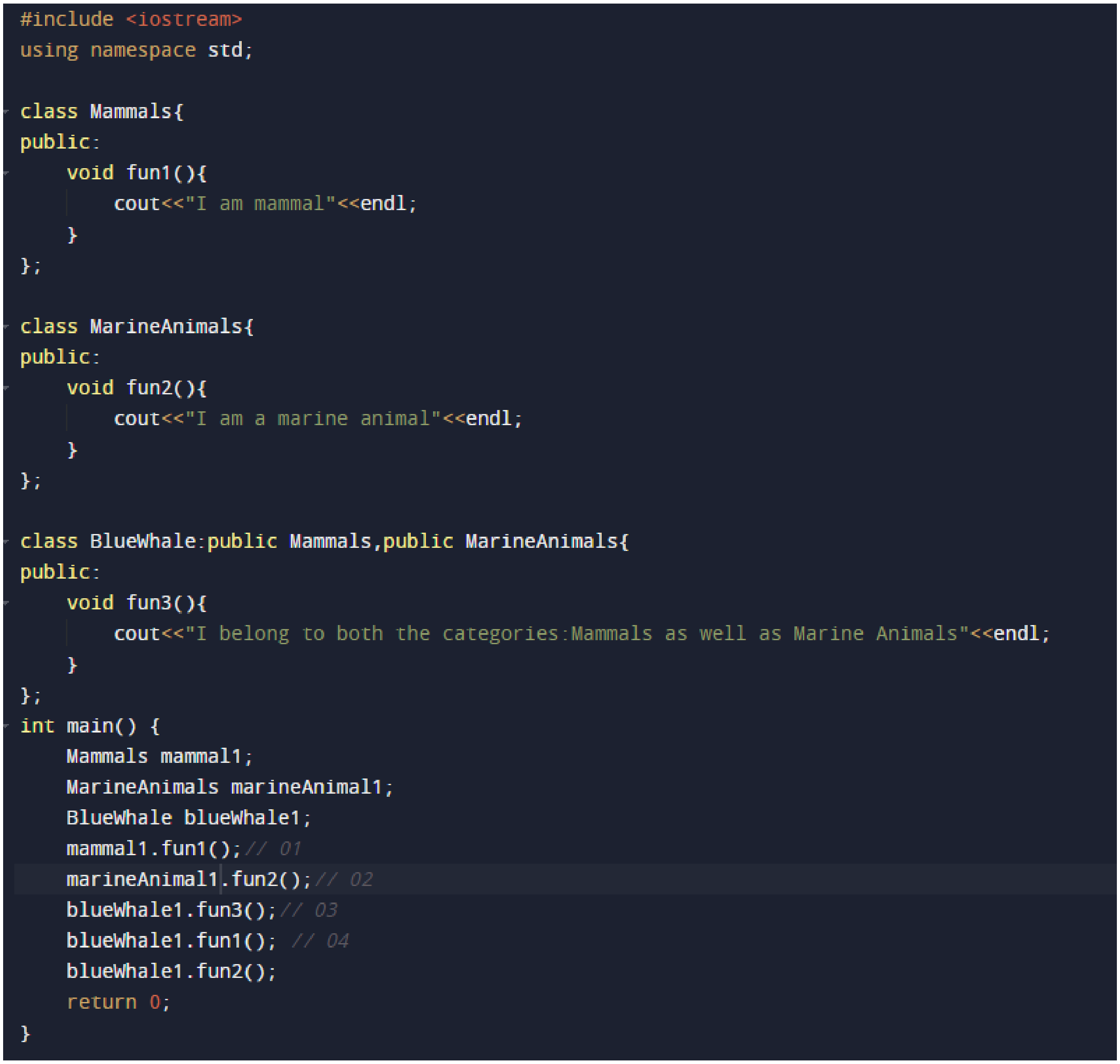
```cpp
#include <iostream>
using namespace std;
class A {
public:
  A() {
      cout << "Class A constructor called" << endl;
  }
};

class B: public A {
public:
  B() {
      cout << "Class B constructor called" << endl;
  }
};

class C: public A {
public:
  C() {
      cout << "Class C constructor called" << endl;
  }
};

class D: public B, public C {
public:
  D() {
      cout << "Class D constructor called" << endl;
  }
};
int main() {
    D d;
    return 0;
}
```



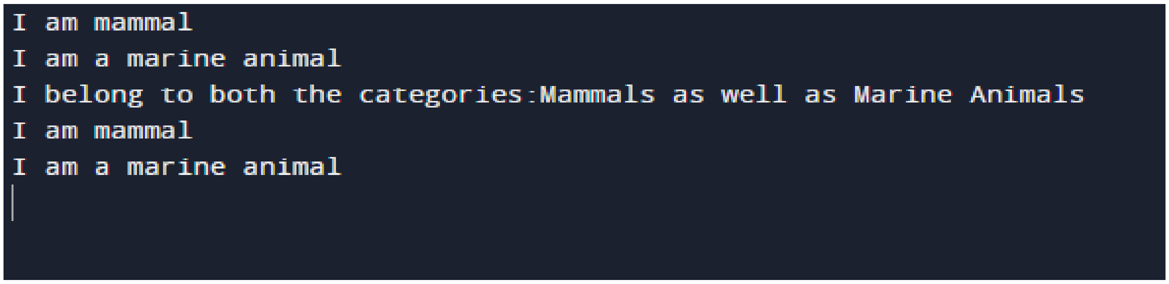
```
Class A constructor called
Class B constructor called
Class A constructor called
Class C constructor called
Class D constructor called
```

- When we are creating object ‘d’ from class D, class A was called twice and 2 objects were created.

05)



```
#include <iostream>
using namespace std;

class Mammals{
public:
    void fun1(){
        cout<<"I am mammal"<<endl;
    }
};

class MarineAnimals{
public:
    void fun2(){
        cout<<"I am a marine animal"<<endl;
    }
};

class BlueWhale:public Mammals,public MarineAnimals{
public:
    void fun3(){
        cout<<"I belong to both the categories:Mammals as well as Marine Animals"<<endl;
    }
};
int main() {
    Mammals mammal1;
    MarineAnimals marineAnimal1;
    BlueWhale blueWhale1;
    mammal1.fun1();// 01
    marineAnimal1.fun2();// 02
    blueWhale1.fun3();// 03
    blueWhale1.fun1(); // 04
    blueWhale1.fun2();
    return 0;
}
```



```
I am mammal
I am a marine animal
I belong to both the categories:Mammals as well as Marine Animals
I am mammal
I am a marine animal
```