8.)**8. Write a program to illustrate working of object, constructor and class.**

```
// Write a program to illustrate working of object, constructor and class
#include <iostream>
using namespace std;
class MyClass
{
        int a;
public:
         MyClass(int x):a(x){}
         void show(){
                cout << a << endl;
         }
        // ~ MyClass();

};
int main(){
        MyClass obj(4);

        obj.show();
        return 0;
}
```

9. Write a cpp program to demonstrate the use of an inline function in a class . A class will collect the details of a student and display.

```
 // Write a cpp program to demonstrate the use of an inline function in a class . A class
will collect the details of a student and display.
#include <iostream>
#include <string>
using namespace std;
class Student{
private:
   int roll;
   string name;
   string department;
public:
   Student():roll(0), name(""), department(""){}

   inline void setRoll(int roll){
      this->roll = roll;
   }
   inline void setName(string name){
      this->name = name;
   }
```

```
    inline void setDepartment(string department){
        this->department = department;
    }

    void showDetails(){
        cout << "Roll no: " << this->roll << endl
             << "Name: " << this->name << endl
             << "Department: " << this->department << endl;
    }

};

int main(){
    Student s;

    s.setRoll(24);
    s.setName("Vishal");
    s.setDepartment("CSE/MCA");

    s.showDetails();
}
```

11. Write a program to show access to private,public and protected using Inheritance

```
#include <iostream>

using namespace std;




class Base {

  public:

    int x;  // public member variable

  private:

    int y;  // private member variable

  protected:
```

```
    int z;  // protected member variable
};



class Derived : public Base {
  public:
    void setValues() {
        x = 10;  // can access public member variable of Base class
        // y = 20;  // cannot access private member variable of Base class
        z = 30;  // can access protected member variable of Base class
    }
    void displayValues() {
        cout << "x: " << x << endl;  // can access public member variable of Base class
        // cout << "y: " << y << endl;  // cannot access private member variable of Base class
        cout << "z: " << z << endl;  // can access protected member variable of Base class
    }
};

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

12. C++ Program to Overriding the member functions using Inheritance

```
// C++ Program to Overriding the member functions using Inheritance

// function overriding means both the function will have same name and same no of arguments as well and also same type of arguments.

// when we inherit any calss and base and derived class both have same above mentioned function then this is know as overriden function

#include <iostream>

using namespace std;

// class other{

// public:

//      virtual void display(){

//              cout << "taliking from other\n";

//      }

// };

class Base{

public:

        void display(){

                cout << "I'm taliking from base class\n";

        }

};

class Derived: public Base{

public:

        nvoid display(){

                cout << "I'm taliking from Derived class\n";
```

```
        }
};
int main(){
        // while accessing using object
        // Derived obj;
        // obj.display();

        // Base *p = new Derived();
        Base *p;
        Derived obj;
        p = &obj;
        p->display();
        return 0;
}
```

13. C++ Program to find area and volume using multiple inheritance

```
#include <iostream>
using namespace std;
class Area{
protected:
        double l,w;
public:
        Area(int a, int b):l(a),w(b){}
```

```
};
class Volume{
protected:
	double h;
public:
	Volume(int a):h(a){}

};
class Box: public Area, public Volume{
public:
	Box(int a, int b, int c):Area(a,b), Volume(c){
		// cout << "Box Area: " << getBoxArea() << endl << "Box Volume: " << getBoxVolume() << endl;
	}

	double getBoxArea(){
		return l * w;
	}
	double getBoxVolume(){
		return getBoxArea() * h;
	}
};
int main(){

	Box obj(4,2,3);
```

```
	cout << "Box Area: " << obj.getBoxArea() << endl << "Box Volume: " << obj.getBoxVolume() << endl;

	return 0;

}
```

14. C++ Program to illustrates the use of Constructors in multiple inheritance

```
// C++ Program to illustrates the use of Constructors in multiple inheritance

#include<iostream>

using namespace std;


class A{

protected:

	int a;

public:

	A(int x):a(x){}


};


class B{


protected:

int b;

public:

	B(int x):b(x){}
```

```
};


class C: public A, public B{
private:
        int  c;
public:
        C(int x, int y, int z):A(x), B(y),c(z){
                cout<<"a="<<a<<endl<<"b="<<b<<endl<<"c="<<c<<endl;
        }


};
int main(){
        C obj(2,3,4);


        return 0;
}
```