

C++ PROGRAMS

Inheritance

C++ program to show the concept of Single Inheritance.

```
#include<iostream>
using namespace std;
class Animal {

    public:
    void eat() {
        cout << "I can eat!" << endl;
    }

    void sleep() {
        cout << "I can sleep!" << endl;
    }
};
// derived class
class Dog : public Animal {

    public:
    void bark() {
        cout << "I can bark!" << endl;
    }
};

int main() {
    Dog dog1; // Create object of the Dog class
    // Calling members of the base class
    dog1.eat();
    dog1.sleep();

    // Calling member of the derived class
    dog1.bark();

    return 0;
}
```

C++ program to show the concept of Multiple Inheritance.

```
#include<iostream>
using namespace std;
class Base
{
    public:
    void show1()
    {
        cout<<" This is show function of first base class" << endl;
    }
};
class Base2
{

```

```

public:
void show2()
{
cout<<" This is show function of second base class" << endl;
}
};
class derived: public Base1,public Base2
{
public:
void show3()
{
cout<<" This is show function of the derived class" << endl;
}
};
int main()
{
derived d;
d.show1();
d.show2();
d.show3();
return 0;
}

```

C++ Program for Hierarchical Inheritance.

```

#include<iostream>
using namespace std;
// base class
class Animal {
public:
void info()
{
cout << "I am an animal." << endl;
}
};

// derived class 1
class Dog : public Animal {
public:
void bark()
{
cout << "I am a Dog" << endl;
}
};

// derived class 2
class Cat : public Animal {
public:
void meow()
{
cout << "I am a Cat." << endl;
}
};

int main()
{
Dog dog1;

```

```

        dog1.info();
        dog1.bark();

        Cat cat1;

        cat1.info();
        cat1.meow();

        return 0;
}

```

C++ Program for Multilevel Inheritance.

```

#include<iostream>
using namespace std;
class Vehicle
{
    public:
        void Vehicle()
        {
            cout << "This is a Vehicle" << endl;
        }
};

class fourWheeler: public Vehicle
{
    public:
        void fourWheeler()
        {
            cout<<" Objects with 4 wheels are vehicles" << endl;
        }
};

class Car: public fourWheeler{
    public:
        void Car()
        {
            cout<<"Car has 4 Wheels" << endl;
        }
};

int main()
{
    Car obj;
    obj.Vehicle();
    obj.fourWheeler();
    obj.Car();
    return 0;
}

```

C++ Program for Hybrid Inheritance.

```

#include<iostream>
using namespace std;
class a
{

```

```
public:
void show()
{
cout<< "This is class A.";
}
};
class b: virtual public a
{

};
class c: virtual public a
{

};
class d: public b , public c
{

};

int main()
{
    d obj;
    obj.show();
    return 0;
}
```