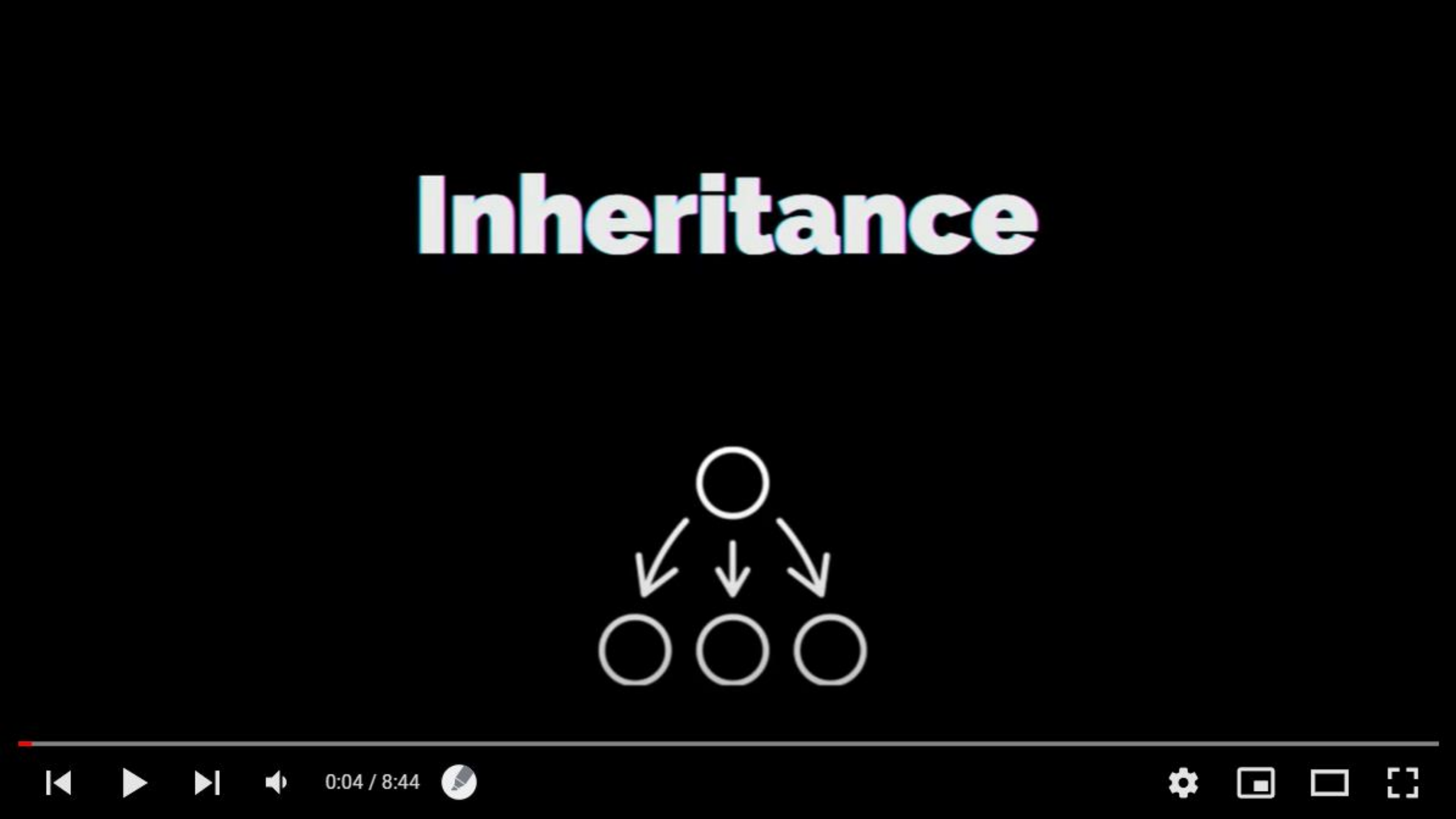
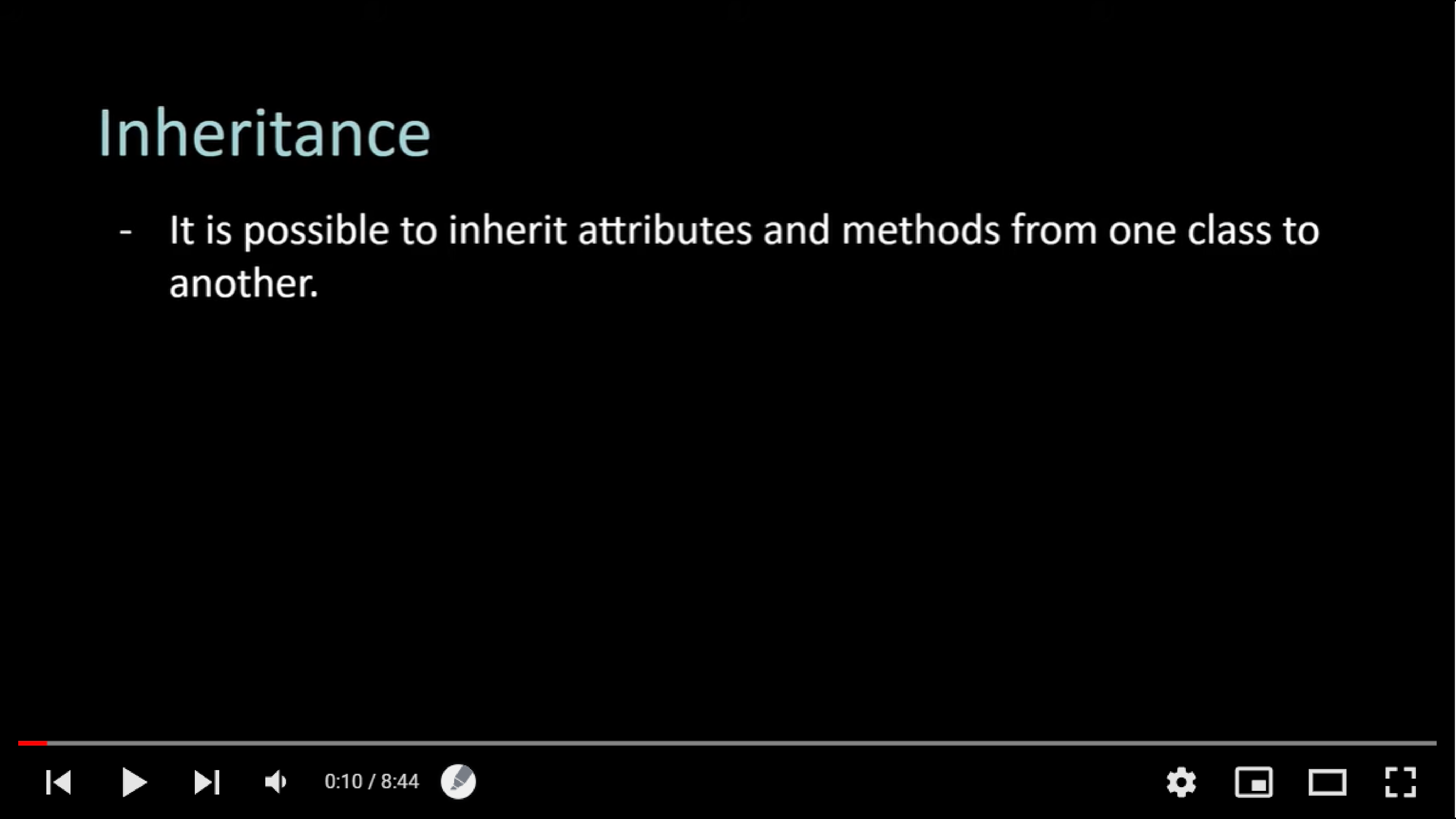
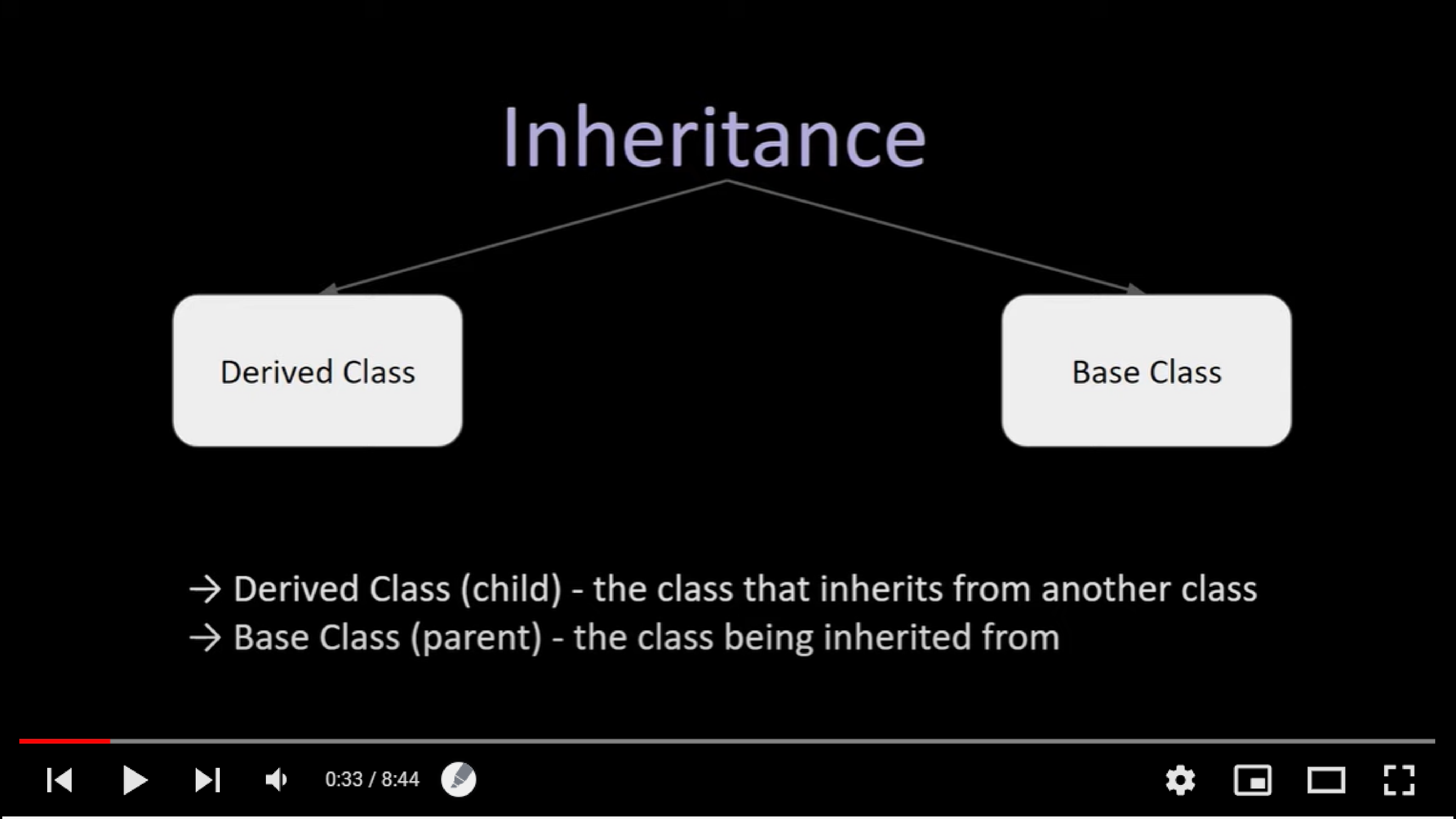


Inheritance



Inheritance

- It is possible to inherit attributes and methods from one class to another.



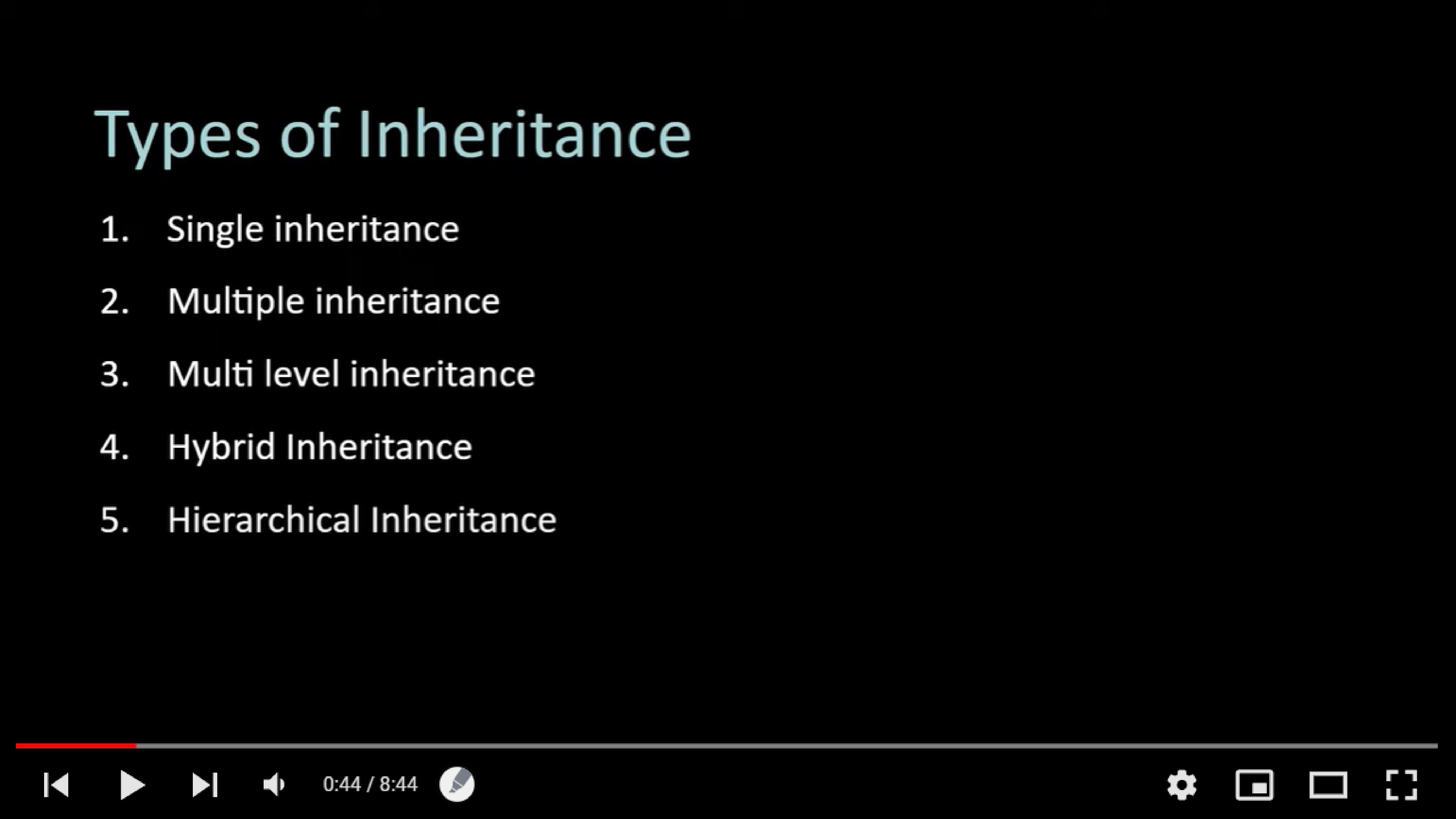
Inheritance

Derived Class

Base Class

→ Derived Class (child) - the class that inherits from another class

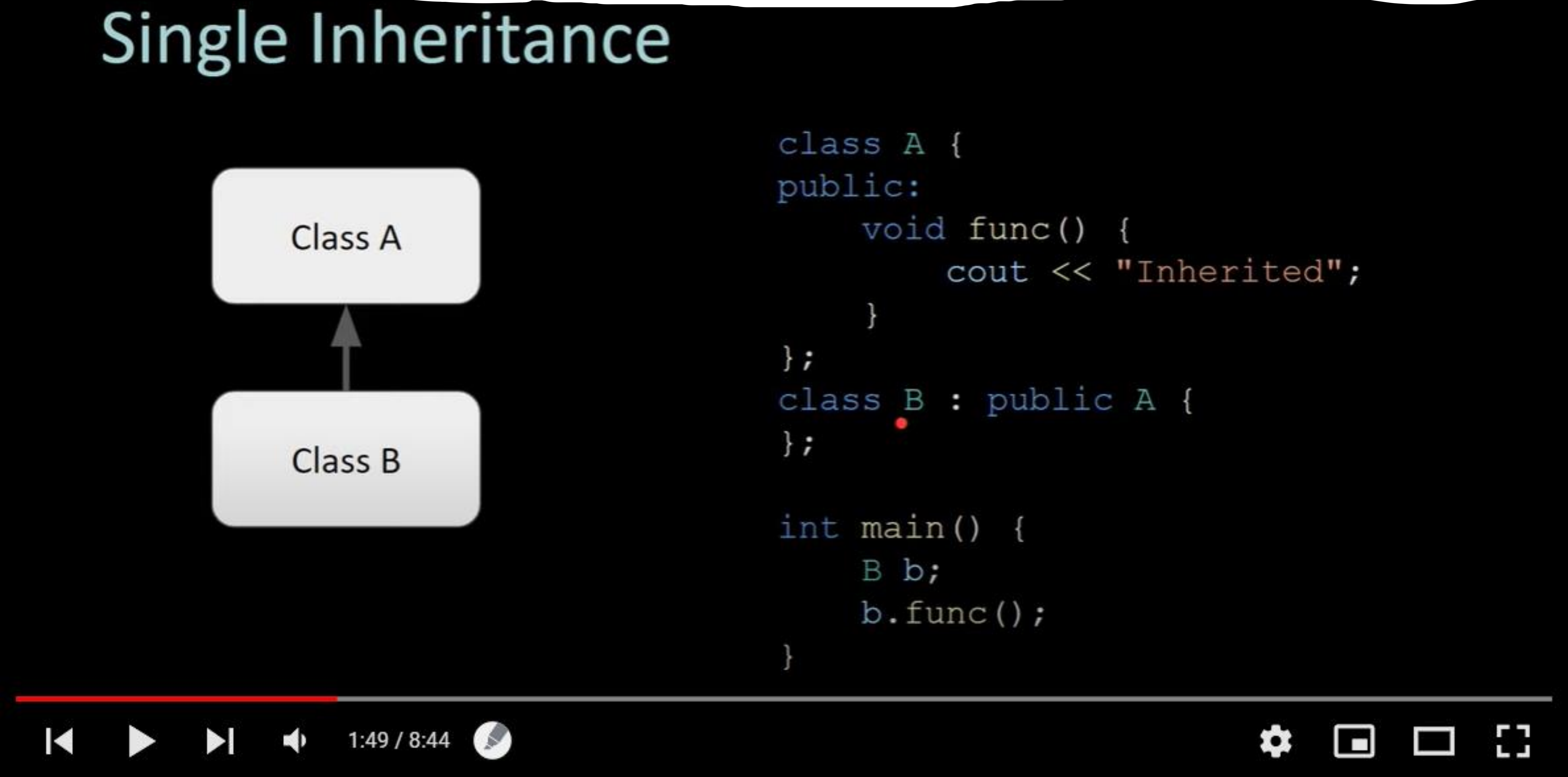
→ Base Class (parent) - the class being inherited from



Types of Inheritance

1. Single inheritance
2. Multiple inheritance
3. Multi level inheritance
4. Hybrid Inheritance
5. Hierarchical Inheritance

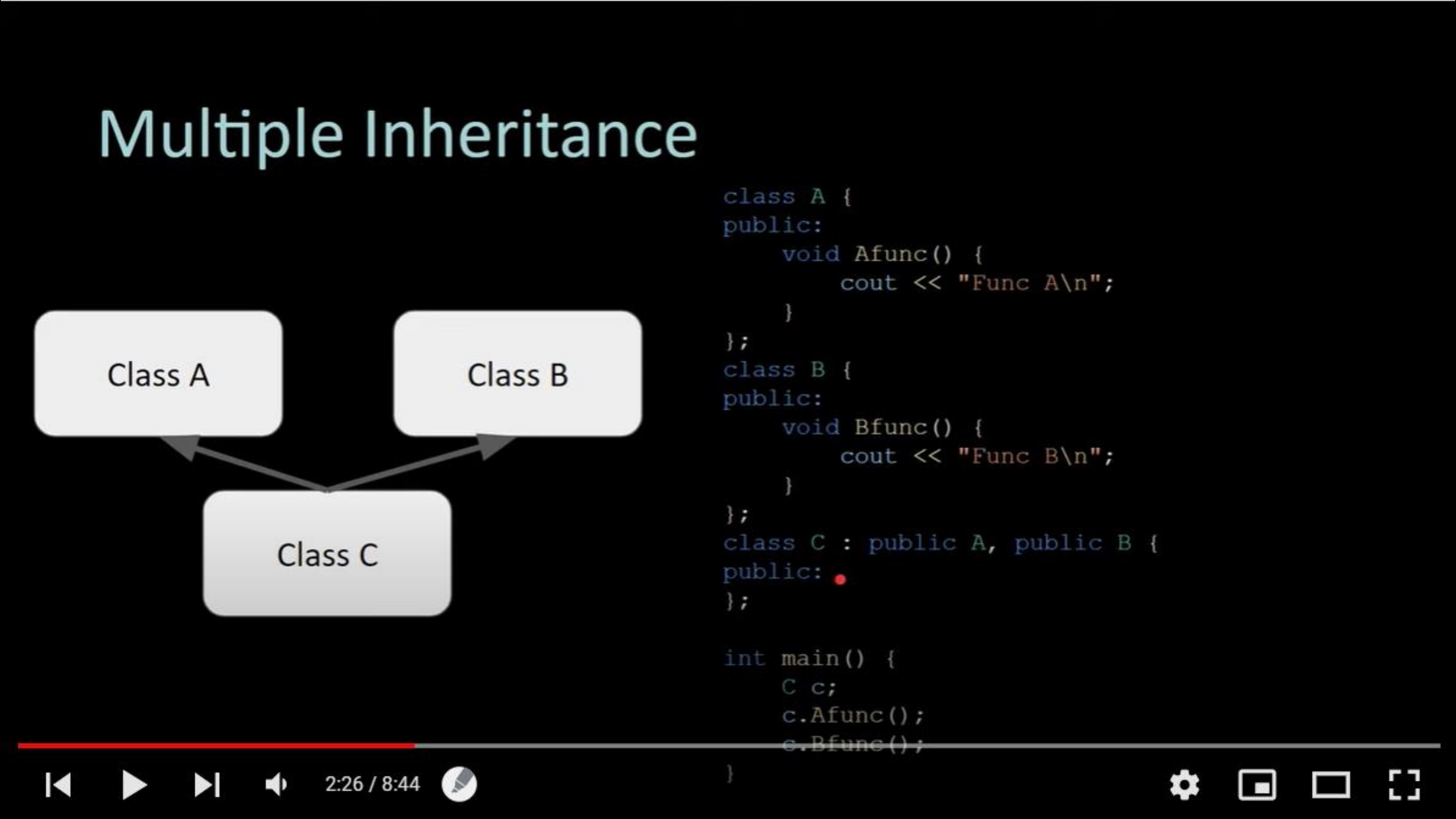
REMEMBER : PROTECTED REMAINS PROTECTED IN DERIVED CLASS UNLESS PRIVATE IS USED WHILE DERIVING



Single Inheritance

```
class A {
public:
    void func() {
        cout << "Inherited";
    }
};
class B : public A {
};

int main() {
    B b;
    b.func();
}
```



Multiple Inheritance

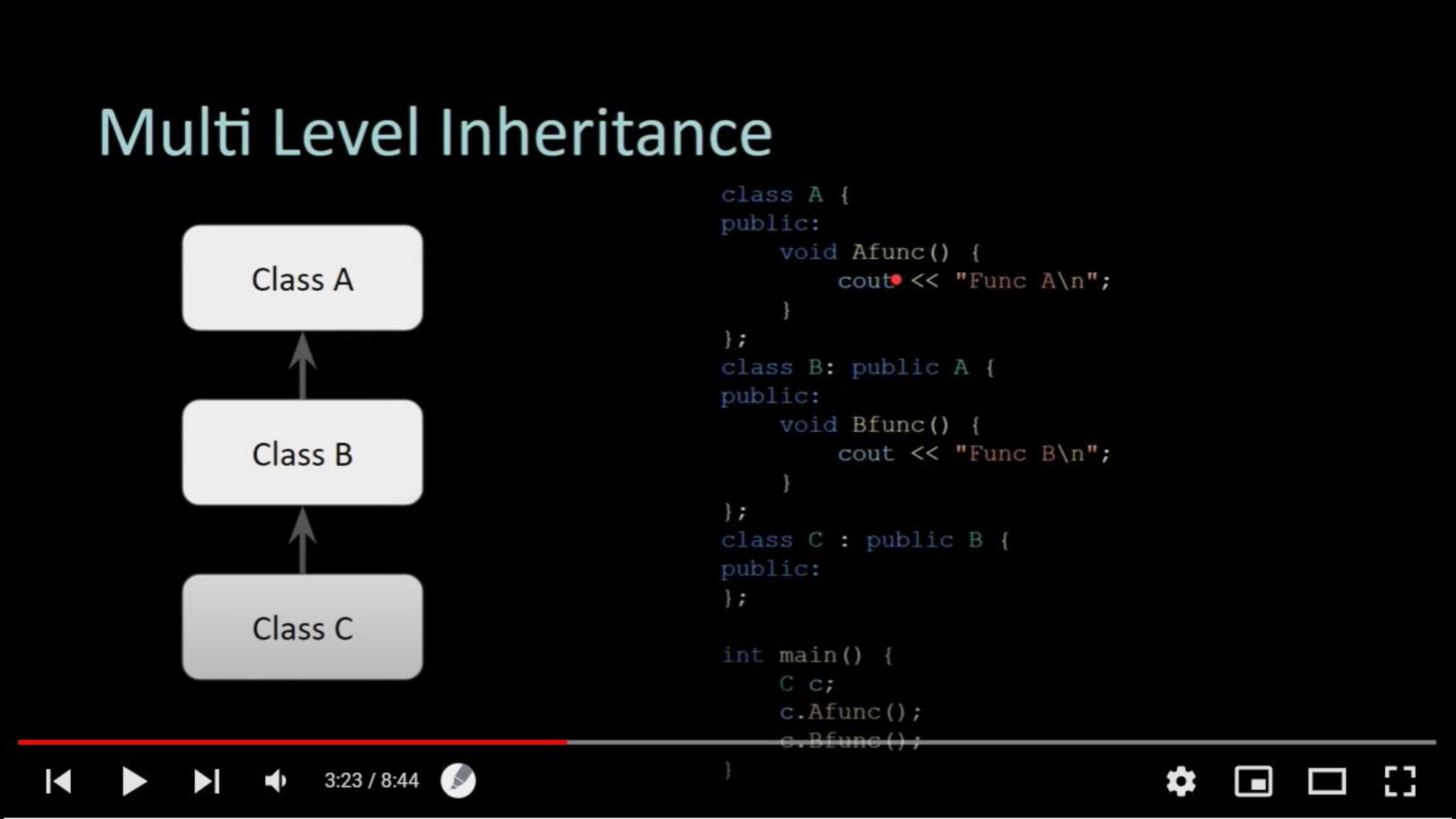
Class A

Class B

Class C

```
class A {
public:
    void Afunc() {
        cout << "Func A\n";
    }
};
class B {
public:
    void Bfunc() {
        cout << "Func B\n";
    }
};
class C : public A, public B {
public:
};

int main() {
    C c;
    c.Afunc();
    c.Bfunc();
}
```



Multi Level Inheritance

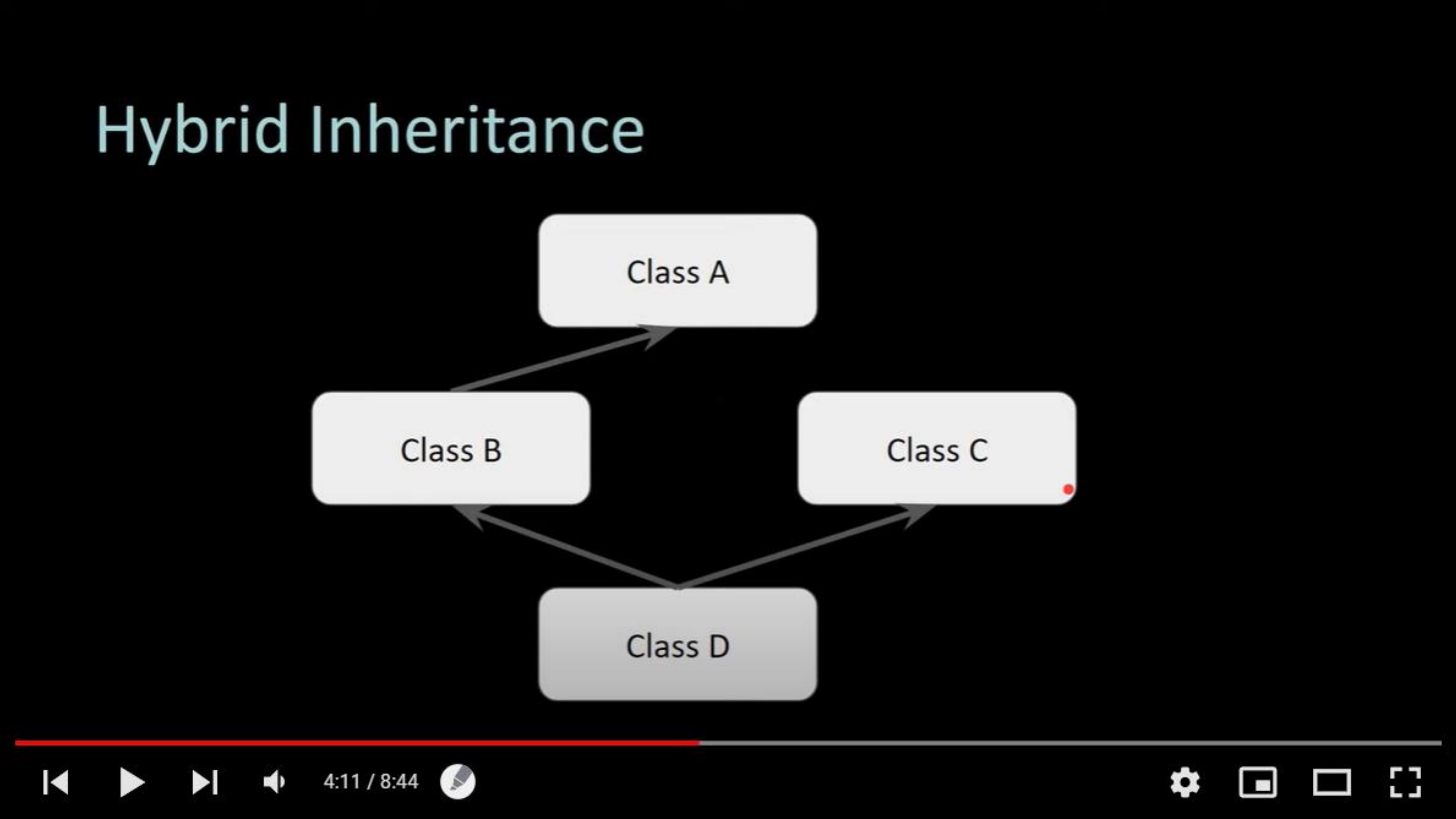
Class A

Class B

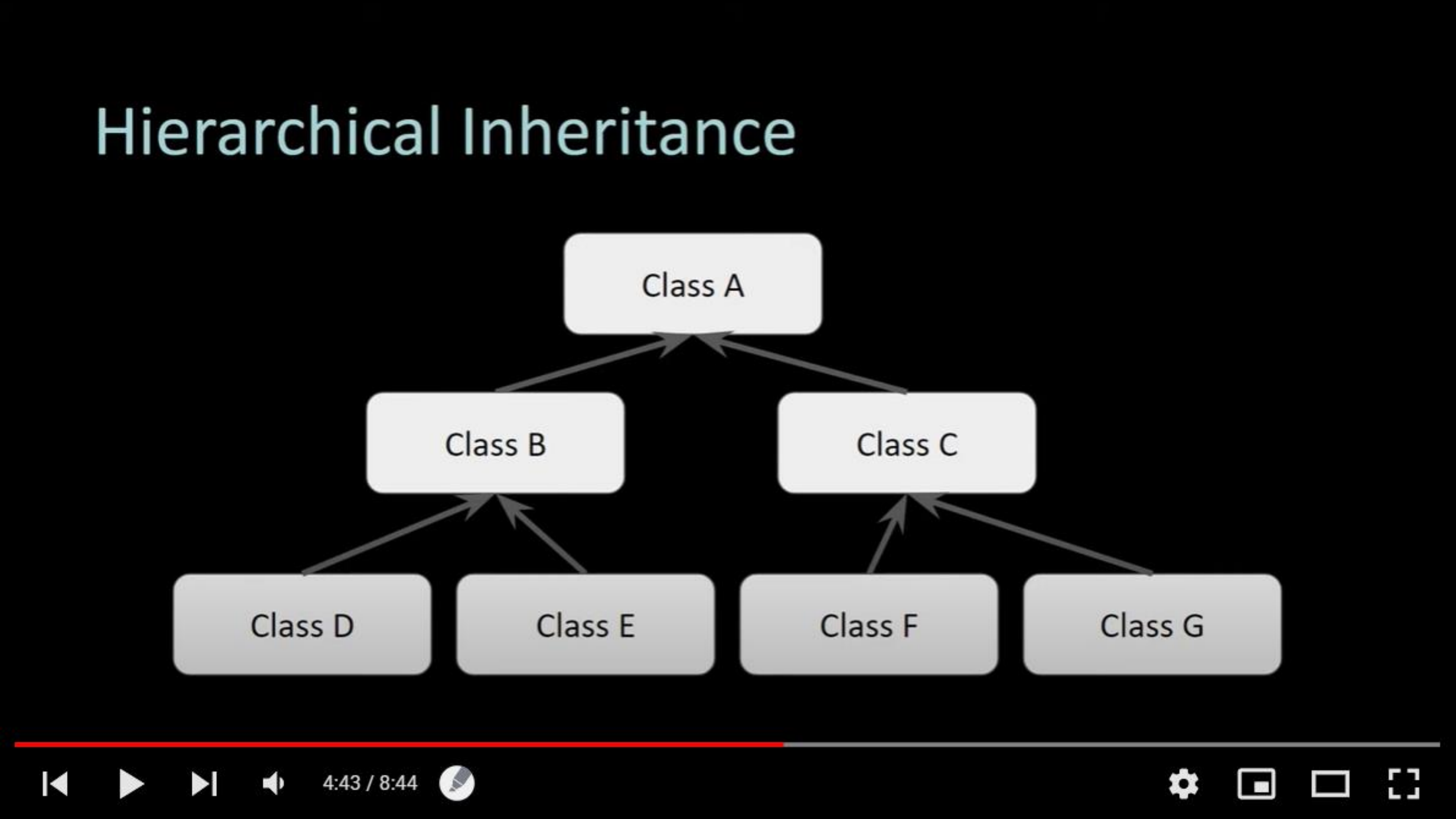
Class C

```
class A {
public:
    void Afunc() {
        cout << "Func A\n";
    }
};
class B: public A {
public:
    void Bfunc() {
        cout << "Func B\n";
    }
};
class C : public B {
public:
};

int main() {
    C c;
    c.Afunc();
    c.Bfunc();
}
```



Hybrid Inheritance



Hierarchical Inheritance

Class A

Class B

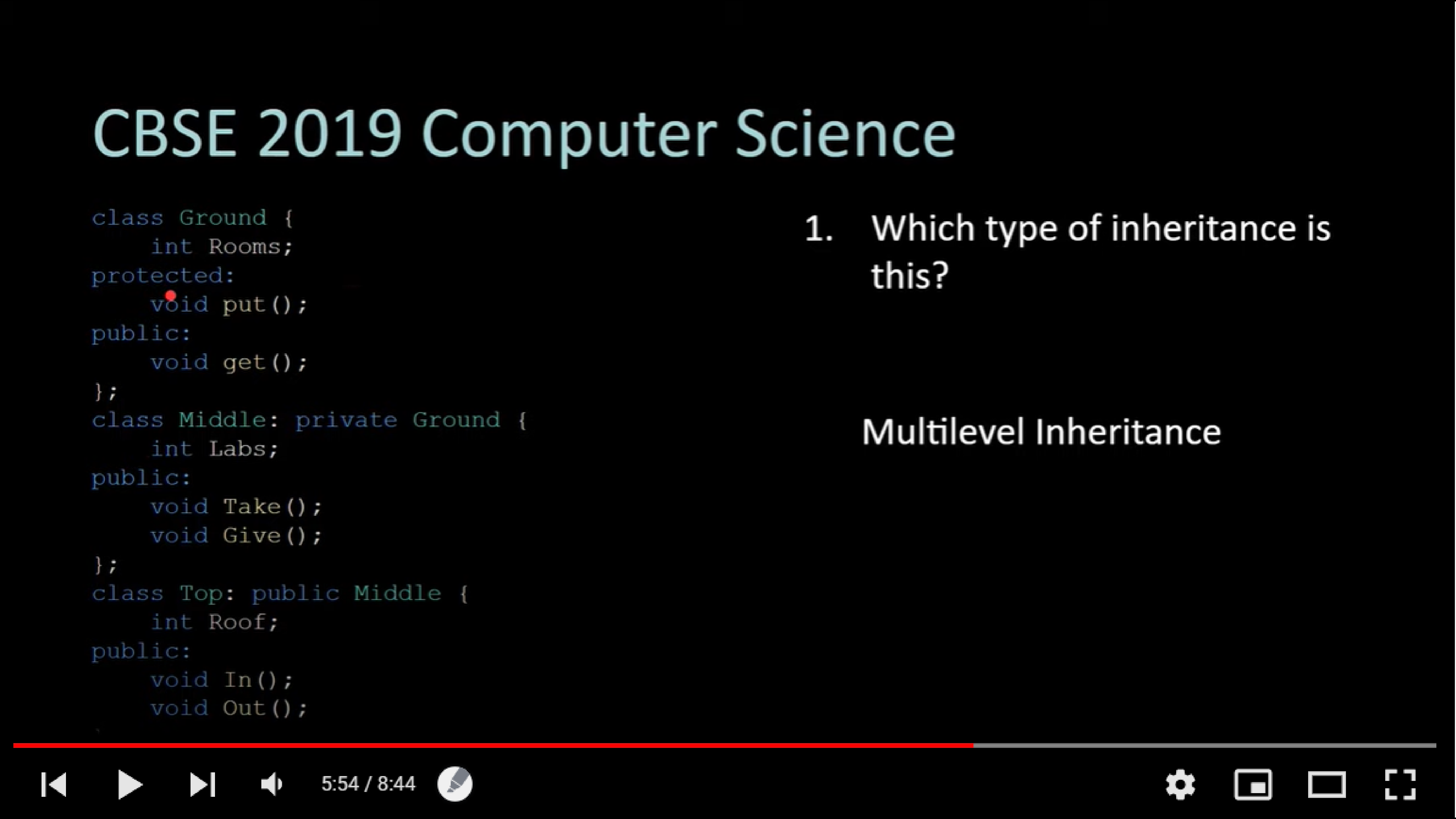
Class C

Class D

Class E

Class F

Class G

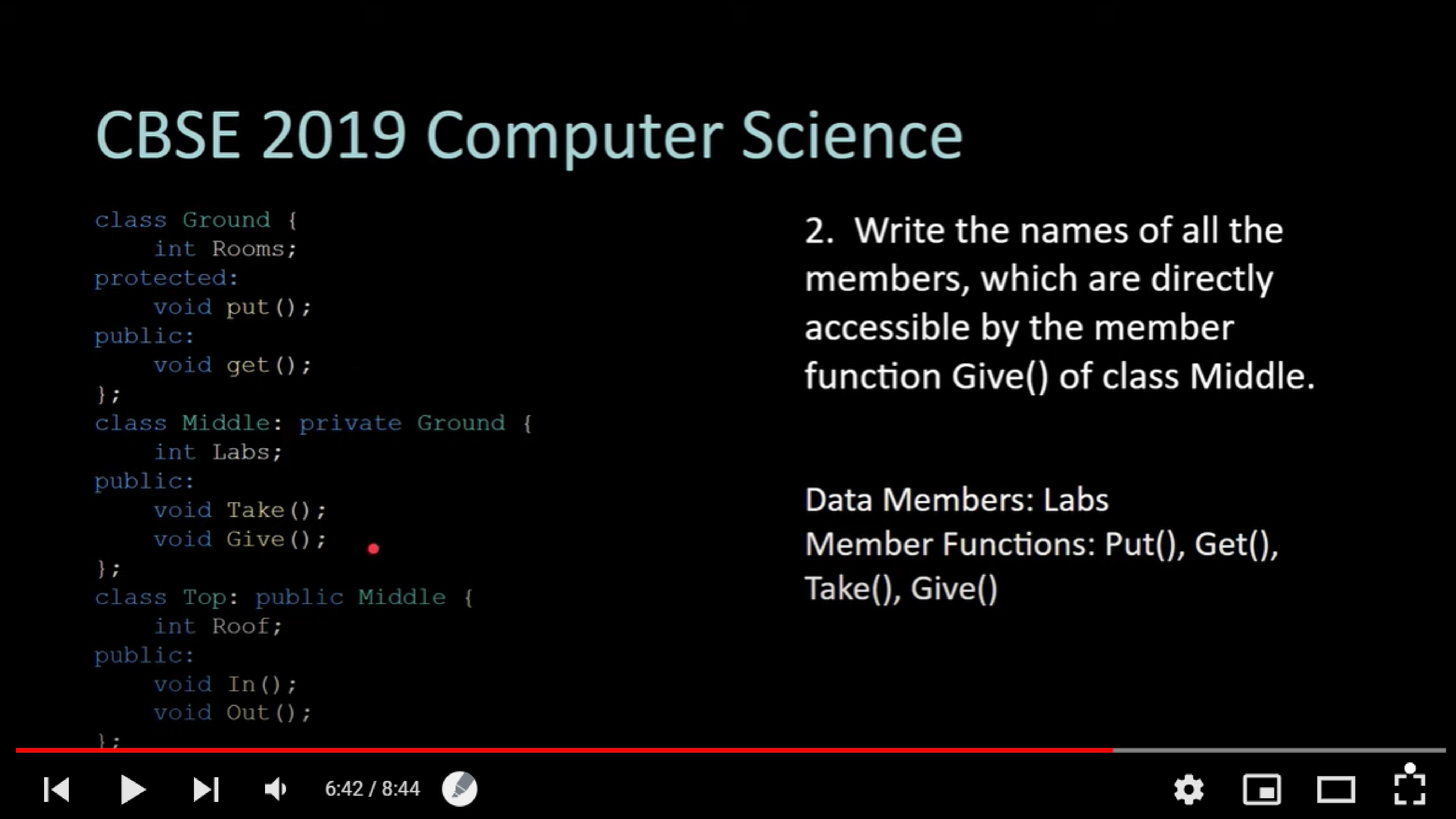


CBSE 2019 Computer Science

```
class Ground {
    int Rooms;
protected:
    void put();
public:
    void get();
};
class Middle: private Ground {
    int Labs;
public:
    void Take();
    void Give();
};
class Top: public Middle {
    int Roof;
public:
    void In();
    void Out();
```

1. Which type of inheritance is this?

Multilevel Inheritance

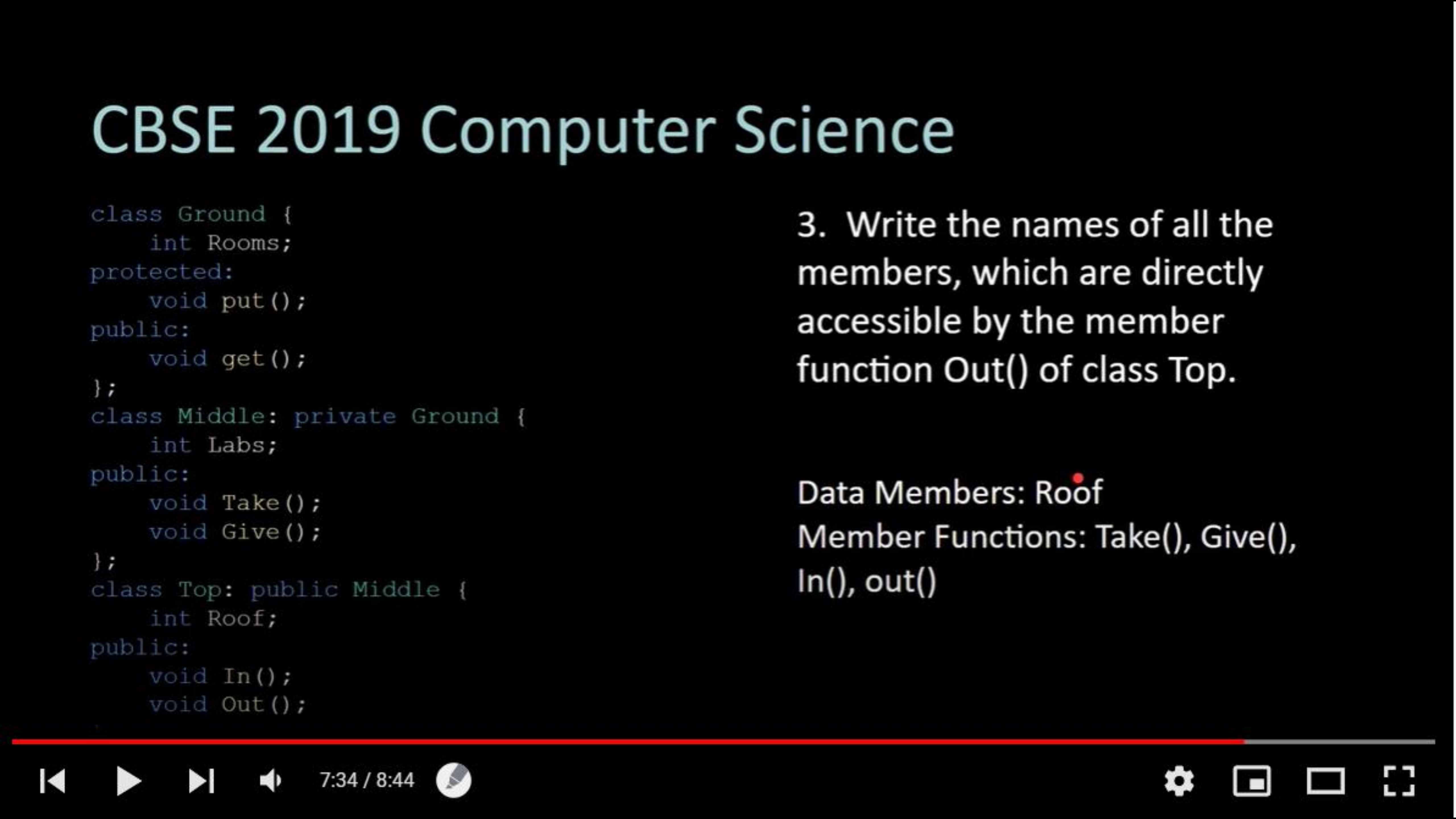


CBSE 2019 Computer Science

```
class Ground {
    int Rooms;
protected:
    void put();
public:
    void get();
};
class Middle: private Ground {
    int Labs;
public:
    void Take();
    void Give();
};
class Top: public Middle {
    int Roof;
public:
    void In();
    void Out();
};
```

2. Write the names of all the members, which are directly accessible by the member function Give() of class Middle.

Data Members: Labs
Member Functions: Put(), Get(), Take(), Give()



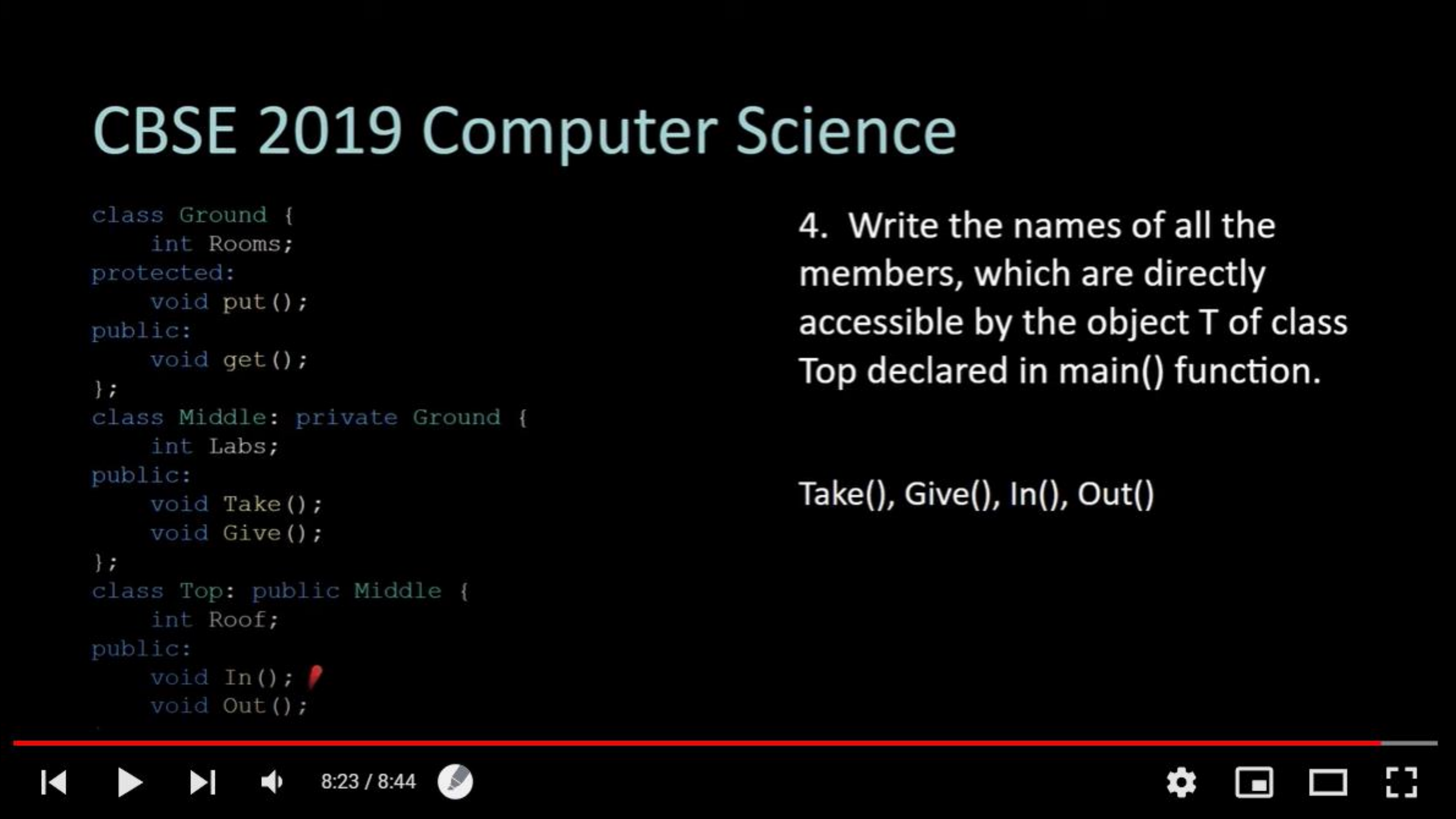
CBSE 2019 Computer Science

```
class Ground {
    int Rooms;
protected:
    void put();
public:
    void get();
};
class Middle: private Ground {
    int Labs;
public:
    void Take();
    void Give();
};
class Top: public Middle {
    int Roof;
public:
    void In();
    void Out();
```

3. Write the names of all the members, which are directly accessible by the member function Out() of class Top.

Data Members: Roof

Member Functions: Take(), Give(), In(), out()



CBSE 2019 Computer Science

```
class Ground {
    int Rooms;
protected:
    void put();
public:
    void get();
};
class Middle: private Ground {
    int Labs;
public:
    void Take();
    void Give();
};
class Top: public Middle {
    int Roof;
public:
    void In();
    void Out();
```

4. Write the names of all the members, which are directly accessible by the object T of class Top declared in main() function.

Take(), Give(), In(), Out()