Inheritance in Oops

What is Inheritance?

The process by which one class acquires the properties(data members) and functionalities(methods) of another class is called inheritance. This is basically done by creating new classes, reusing the properties of existing ones.

The old class is known as a base class or superclass or parent class. The new class is called the derived class or subclass or child class. Inheritance represents the IS-A relationship, also known as a parent-child relationship.

The biggest **advantage of Inheritance** is that the code that is already present in the base class need not be rewritten in the child class.

Child Class

The class that extends the features of another class is known as **a child class, subclass, or derived class.**

Parent Class

The class whose properties and functionalities are used(inherited) by another class is known as **a parent class, superclass, or Base class.**

Syntax of java Inheritance

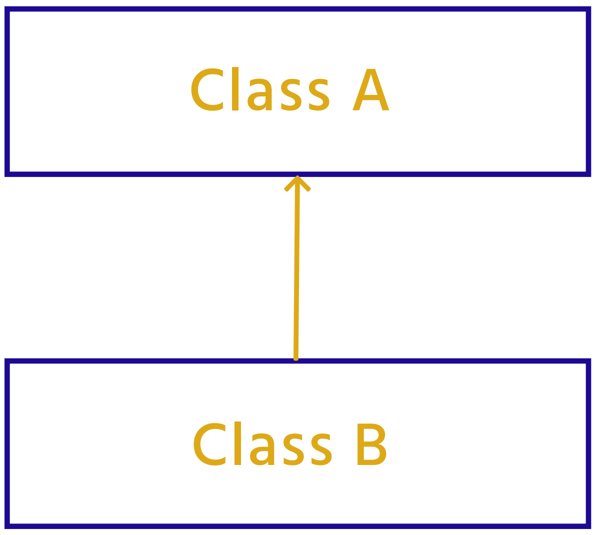
```
class Superclass_Name{
        //methods and fields
}

class Subclass_Name extends Superclass_Name
{
        //methods and fields
}
```

The extends keyword indicates that you are making a new class that derives from an existing class.

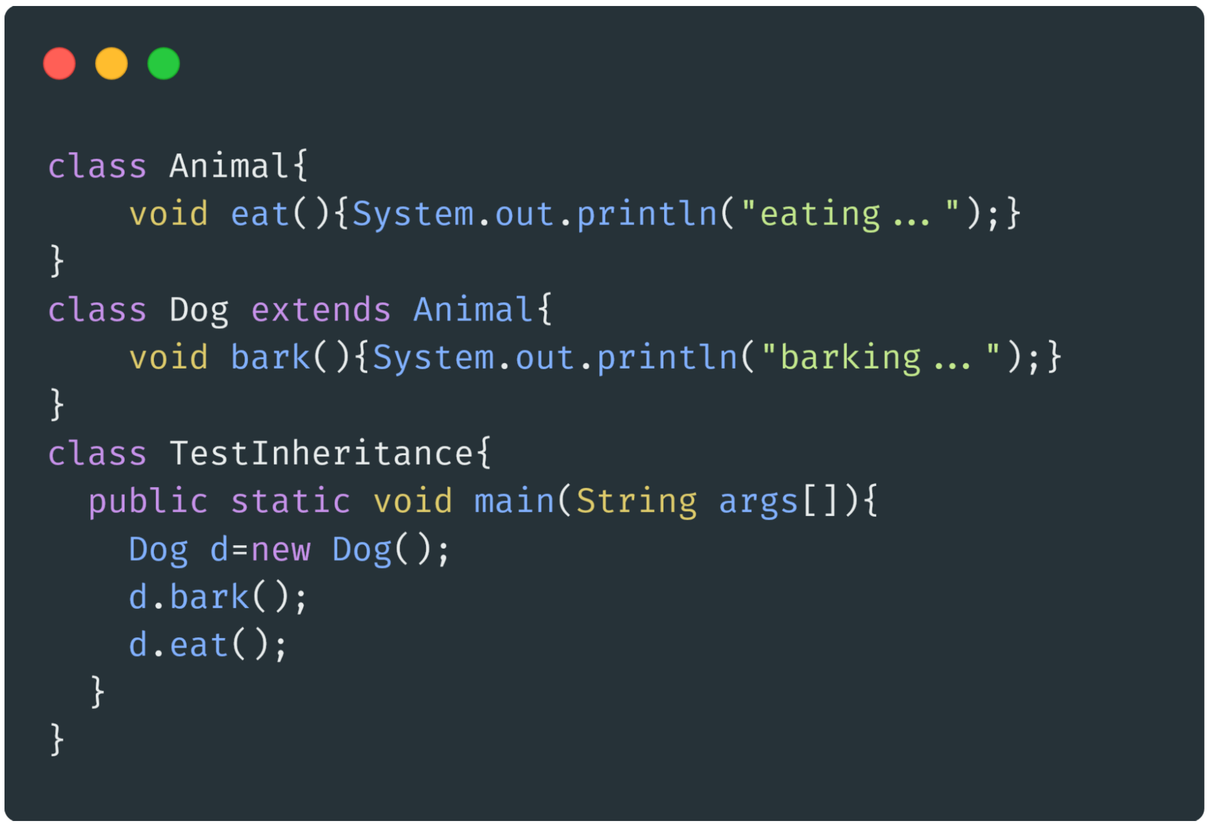
Types of inheritance in java

1] Single Inheritance:

When a class extends another one class only then we call it a single inheritance.



E.g.



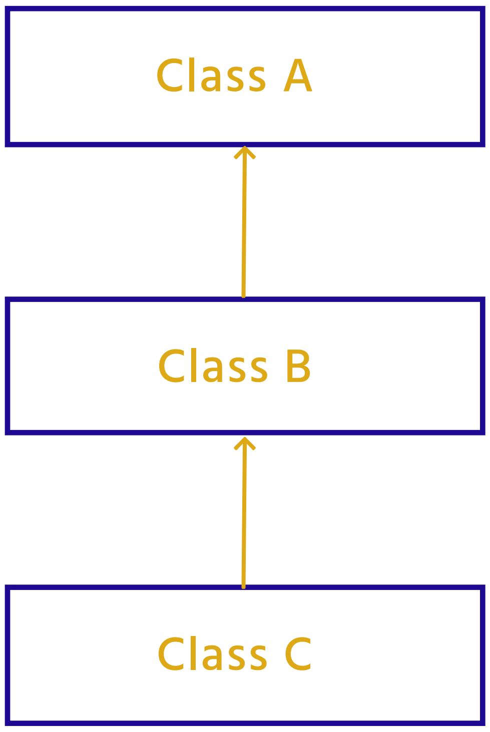
```
class Animal{
    void eat(){System.out.println("eating...");}
}
class Dog extends Animal{
    void bark(){System.out.println("barking...");}
}
class TestInheritance{
  public static void main(String args[]){
    Dog d=new Dog();
    d.bark();
    d.eat();
  }
}
```

Output

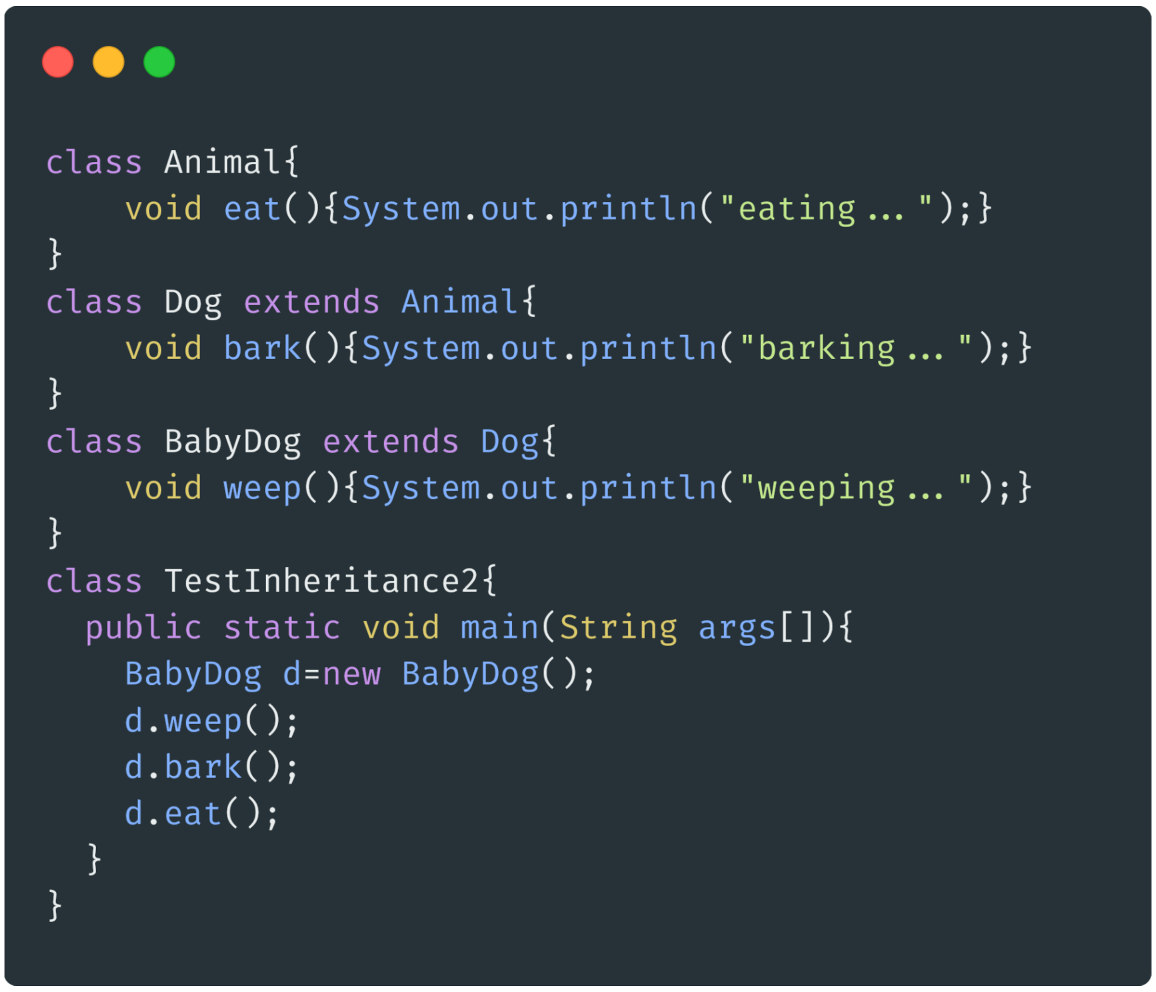
```
barking...
eating...
```

2] Multilevel Inheritance:

When there is a chain of inheritance, it is known as multilevel inheritance. As you can see in the example given below, the C class inherits the B class which again inherits the A class, so there is a multilevel inheritance.



E.g.



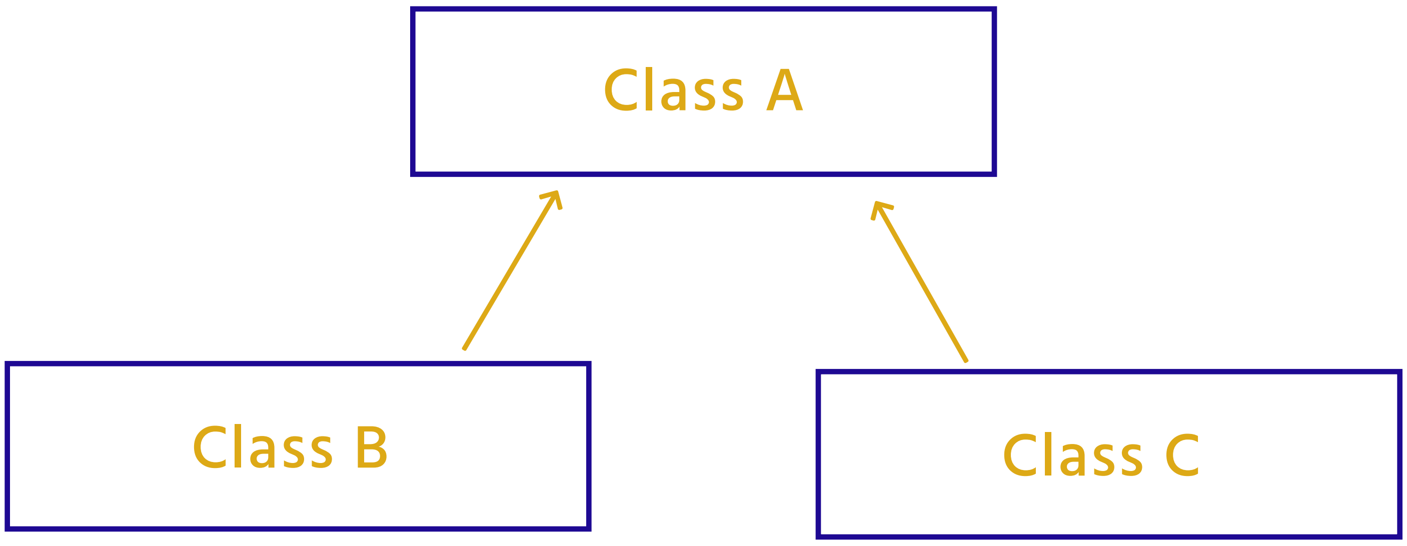
```java
class Animal{
    void eat(){System.out.println("eating...");}
}
class Dog extends Animal{
    void bark(){System.out.println("barking...");}
}
class BabyDog extends Dog{
    void weep(){System.out.println("weeping...");}
}
class TestInheritance2{
  public static void main(String args[]){
    BabyDog d=new BabyDog();
    d.weep();
    d.bark();
    d.eat();
  }
}
```

Output

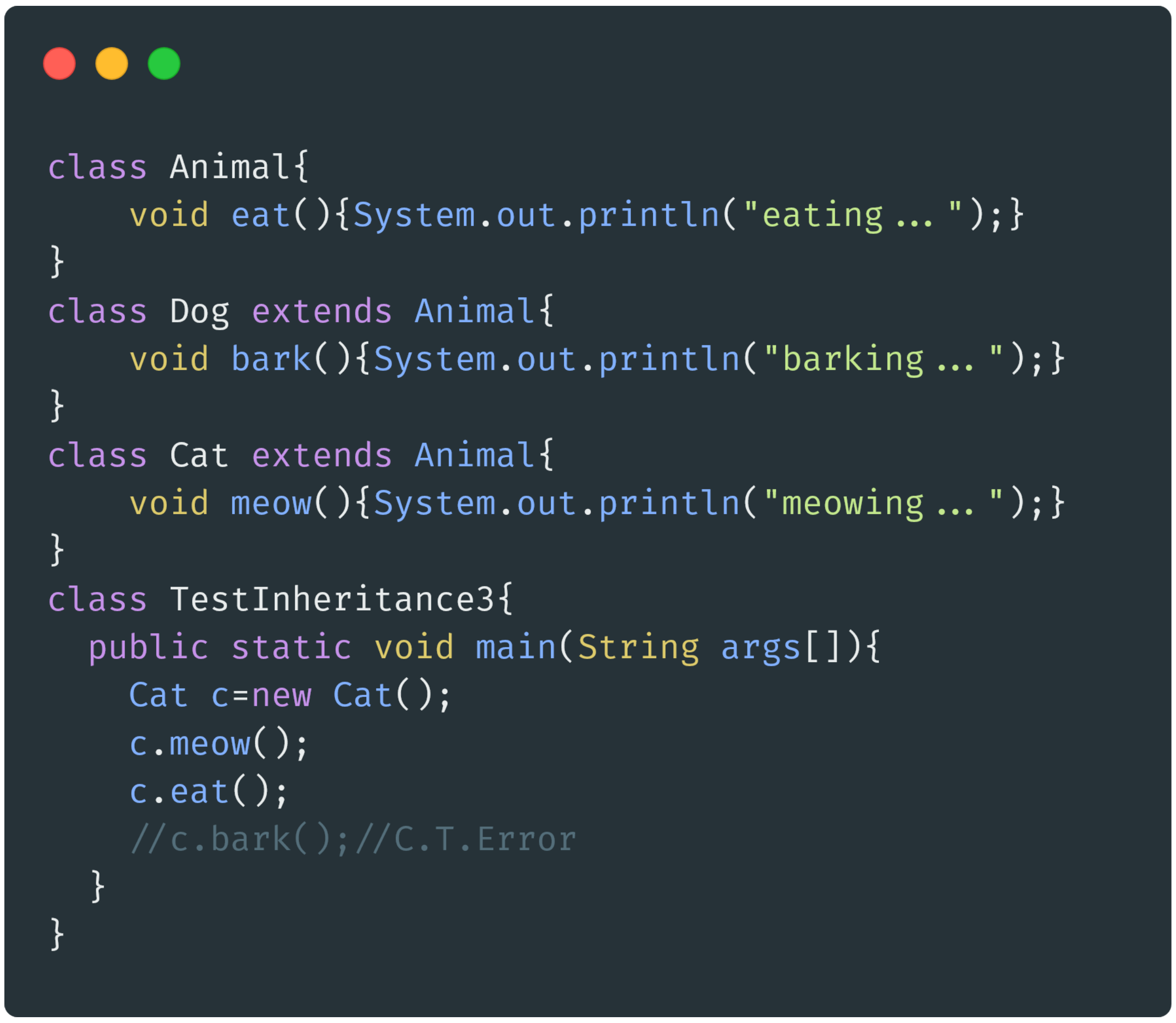
Weeping...
Barking...
eating...

3] Hierarchical Inheritance

When more than one classes inherit the same class then this is called hierarchical inheritance



E.g.



```
class Animal{
    void eat(){System.out.println("eating...");}
}
class Dog extends Animal{
    void bark(){System.out.println("barking...");}
}
class Cat extends Animal{
    void meow(){System.out.println("meowing...");}
}
class TestInheritance3{
  public static void main(String args[]){
    Cat c=new Cat();
    c.meow();
    c.eat();
    //c.bark();//C.T.Error
  }
}
```

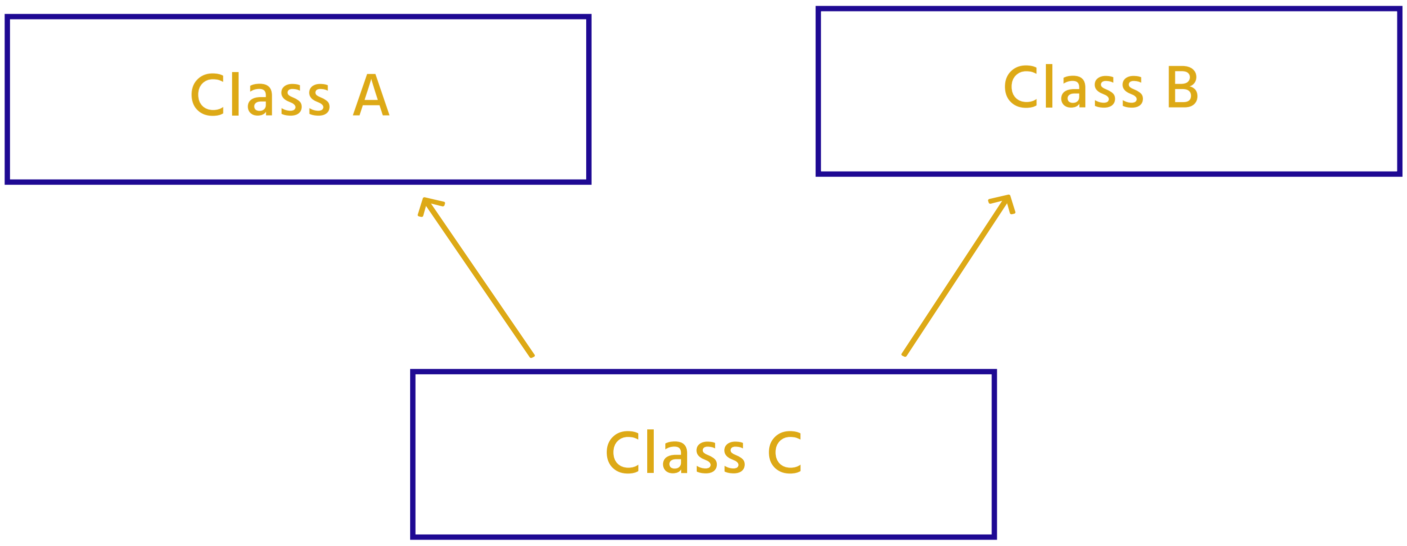
Output

```
meowing...
eating...
```

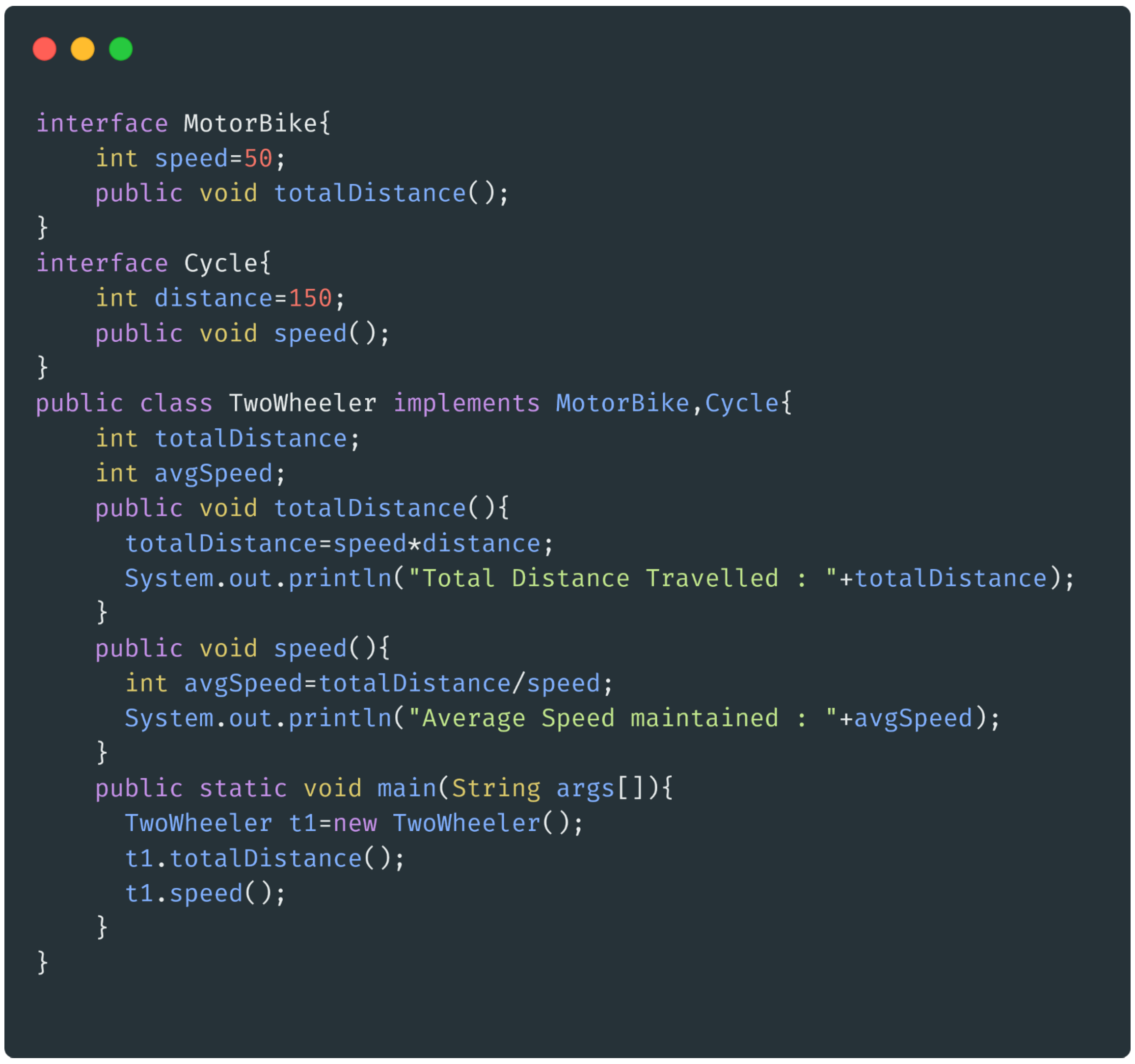
4] Multiple Inheritance

"Multiple Inheritance" refers to the concept of one class extending (Or inherits) more than one base class.

The programming language of java is unable to utilize this feature directly. It can be achieved indirectly through the usage of interfaces.



E.g.



```java
interface MotorBike{
    int speed=50;
    public void totalDistance();
}
interface Cycle{
    int distance=150;
    public void speed();
}
public class TwoWheeler implements MotorBike,Cycle{
    int totalDistance;
    int avgSpeed;
    public void totalDistance(){
      totalDistance=speed*distance;
      System.out.println("Total Distance Travelled : "+totalDistance);
    }
    public void speed(){
      int avgSpeed=totalDistance/speed;
      System.out.println("Average Speed maintained : "+avgSpeed);
    }
    public static void main(String args[]){
      TwoWheeler t1=new TwoWheeler();
      t1.totalDistance();
      t1.speed();
    }
}
```

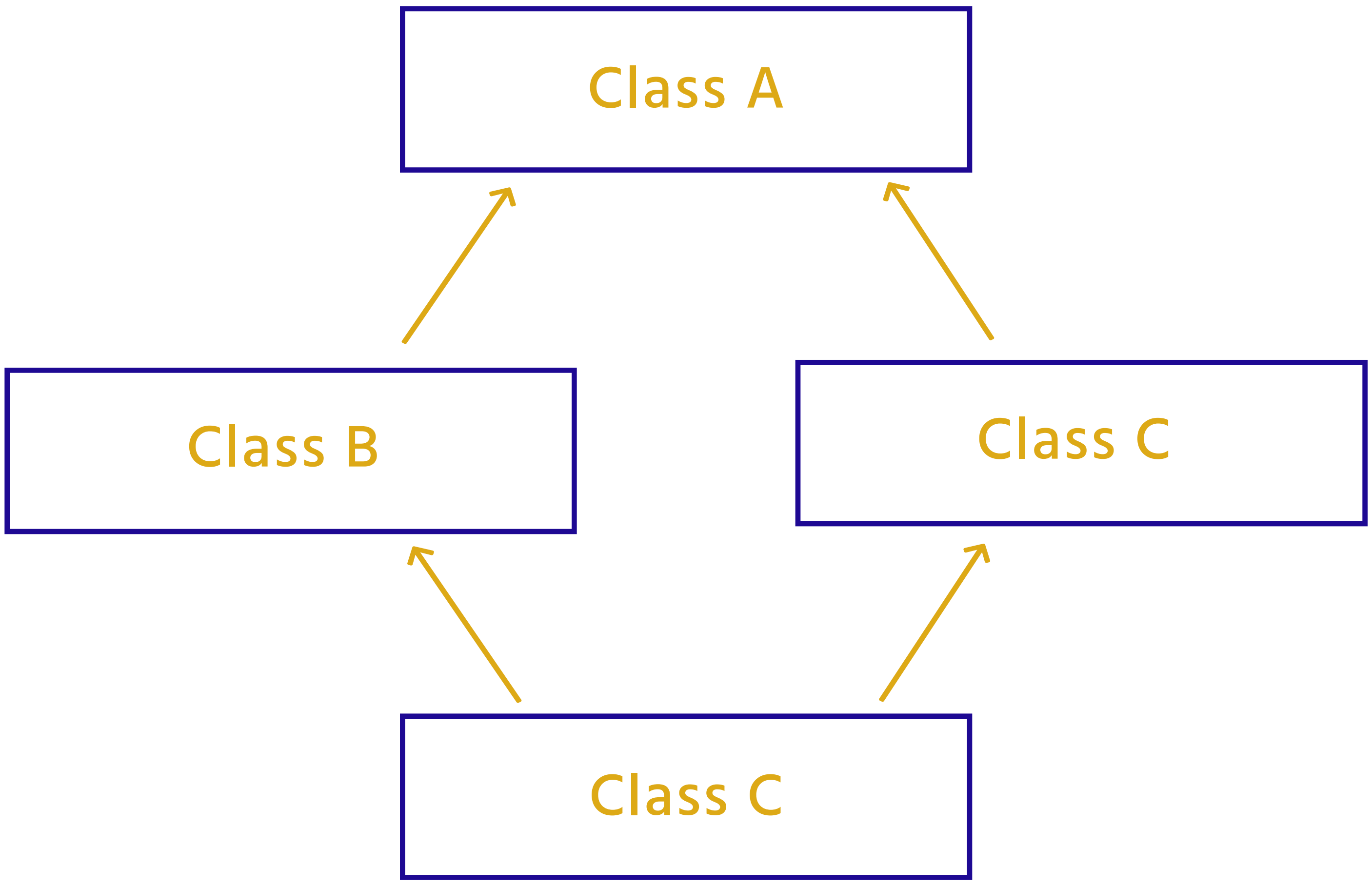
Output

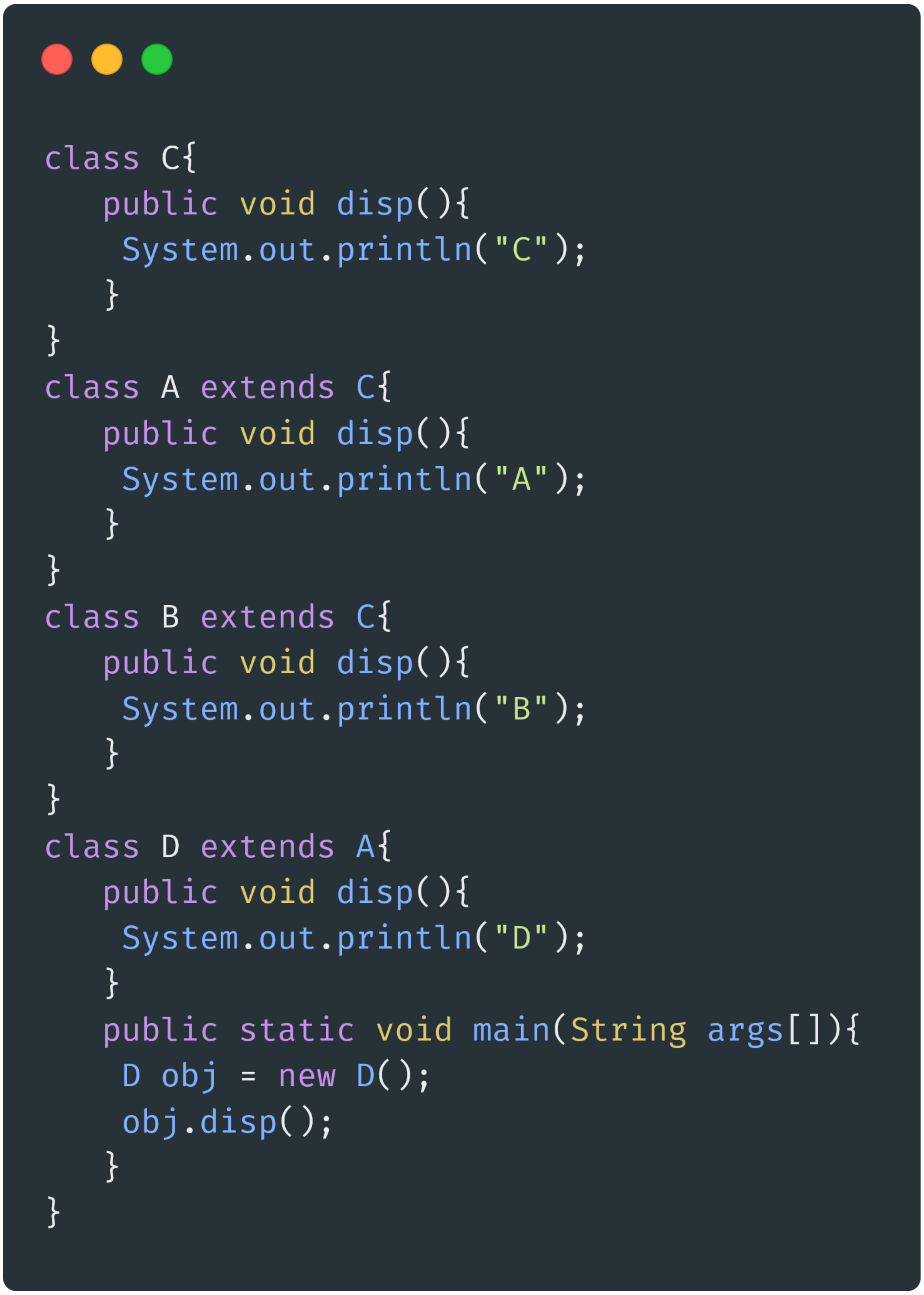
```
Total Distance Travelled: 7500
Average Speed maintained: 150
```

5] Hybrid Inheritance

Hybrid inheritance is a combination of more than one type of inheritance. For example when class A and B extend class C & another class D extends class A then this is a hybrid inheritance because it is a combination of single and hierarchical inheritance.

A hybrid inheritance can be achieved in the java in the same way as multiple inheritance using interfaces.





```
class C{
   public void disp(){
    System.out.println("C");
   }
}
class A extends C{
   public void disp(){
    System.out.println("A");
   }
}
class B extends C{
   public void disp(){
    System.out.println("B");
   }
}
class D extends A{
   public void disp(){
    System.out.println("D");
   }
   public static void main(String args[]){
    D obj = new D();
    obj.disp();
   }
}
```