```java
// Base Class
class Vehicle {
    public void move() {
        System.out.println("Vehicle is moving...");
    }
}

// Intermediate Class
class Car extends Vehicle {
    public void fuel() {
        System.out.println("Car uses fuel or electricity.");
    }
}

// Derived Class
class SportsCar extends Car {
    public void speed() {
        System.out.println("SportsCar has high speed and performance.");
    }
}
```

Vehicle (Base/Super Class)
↑
Car (Intermediate Class - inherits Vehicle)
↑
SportsCar (Child Class - inherits Car)

```java
public class TestMultilevel {
    public static void main(String[] args) {
        SportsCar myCar = new SportsCar();

        myCar.move();    // from Vehicle
        myCar.fuel();    // from Car
        myCar.speed();   // from SportsCar
    }
}
```

Inheritance

Similar to Python, java has a concept of inheritance which allows a class to inherit attributes and methods from another class

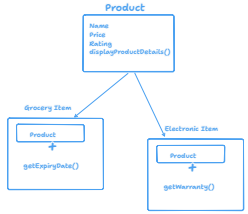
Inheritance in Java is when a class (child/subclass) inherits fields and methods from another class (parent/superclass).

inheritance is when properties & methods of base class are passed on to a derived class

Types of Inheritance

single level Inheritance

Multi level Inheritance



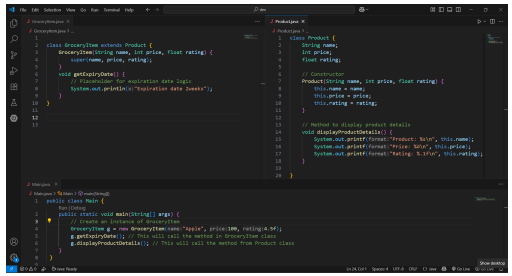

```python
class Product:
    # Block of code


class GroceryItem(Product):
    #  Block of code
```

```java
class Product {
    # Block of code
}

class GroceryItem extends Product(){
    #  Block of code
}
```

Creating class

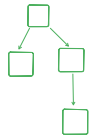
```java
class Product {
    String name;
    int price;
    float rating;

    // Constructor
    Product(String name, int price, float rating) {
        this.name = name;
        this.price = price;
        this.rating = rating;
    }

    // Method to display product details
    void displayProductDetails() {
        System.out.printf("Product: %s\n", this.name);
        System.out.printf("Price: %d\n", this.price);
        System.out.printf("Rating: %.1f\n", this.rating);
    }
}
```

```java
class Product {
    String name;
    int price;
    float rating;
}

class Base {
    public static void main(String[] args) {
        // You can create an object of Product here
        Product p = new Product();
        p.name = "Laptop";
        p.price = 50000;
        p.rating = 4.5f;

        System.out.println("Product Name: " + p.name);
        System.out.println("Price: " + p.price);
        System.out.println("Rating: " + p.rating);
    }
}
```

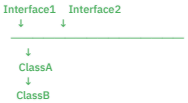


multiple inheritance

Hybrid Inheritance



Hybrid Inheritance is a combination of two or more types of inheritance (like single, multilevel, and hierarchical).
Keyword: Java does not support multiple inheritance with classes to avoid ambiguity, but we can achieve hybrid inheritance using interfaces.




```java
// Interface 1
interface Engine {
    void startEngine();
}

// Interface 2
interface Fuel {
    void fuelType();
}

// Base class
class Vehicle {
    void move() {
        System.out.println("Vehicle is moving...");
    }
}

// Child class (Hybrid Inheritance)
class Car extends Vehicle implements Engine, Fuel {
    @Override
    public void startEngine() {
        System.out.println("Car engine starts with a button.");
    }

    @Override
    public void fuelType() {
        System.out.println("Car uses petrol.");
    }

    void display() {
        System.out.println("This is a Car.");
    }
}
```

```java
public class HybridInheritanceExample {
    public static void main(String[] args) {
        Car myCar = new Car();

        myCar.display();      // Own method
        myCar.move();         // Inherited from Vehicle
        myCar.startEngine();  // From Engine interface
        myCar.fuelType();     // From Fuel interface
    }
}
```