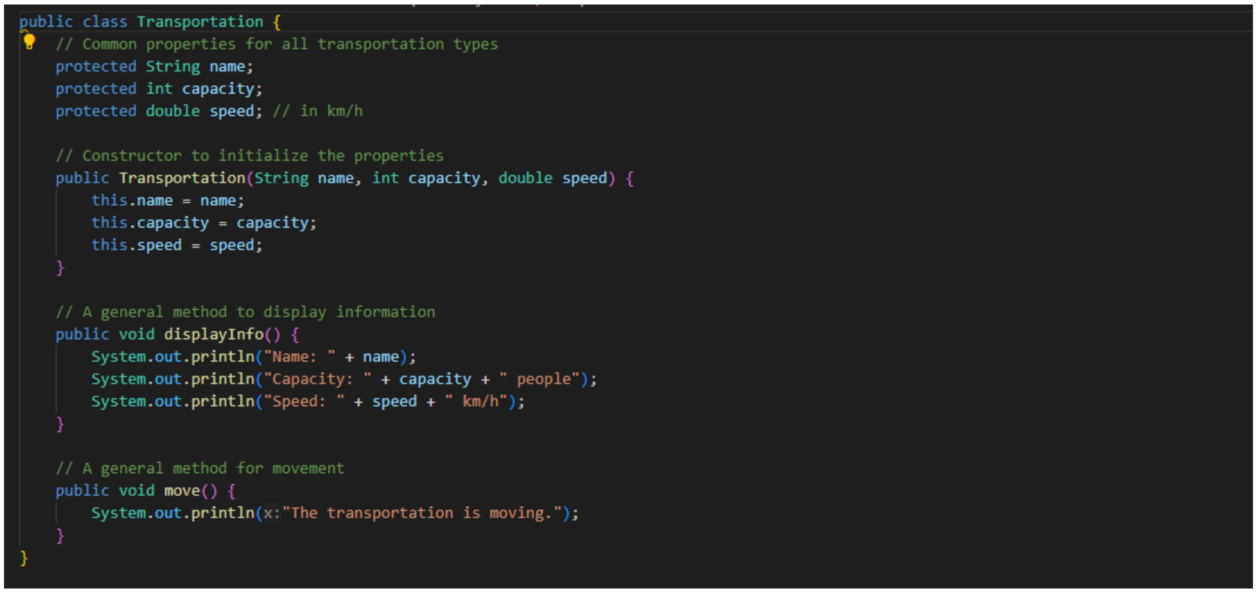
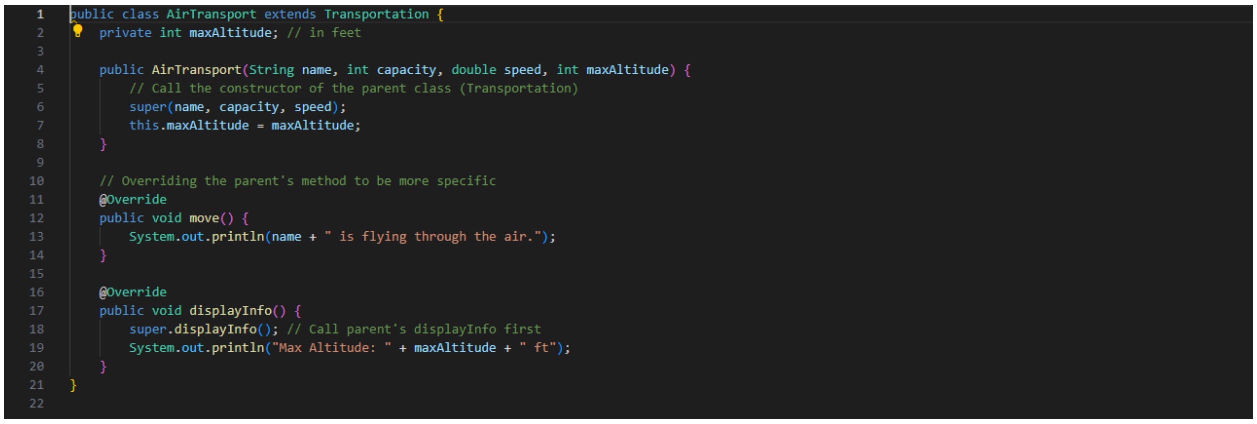
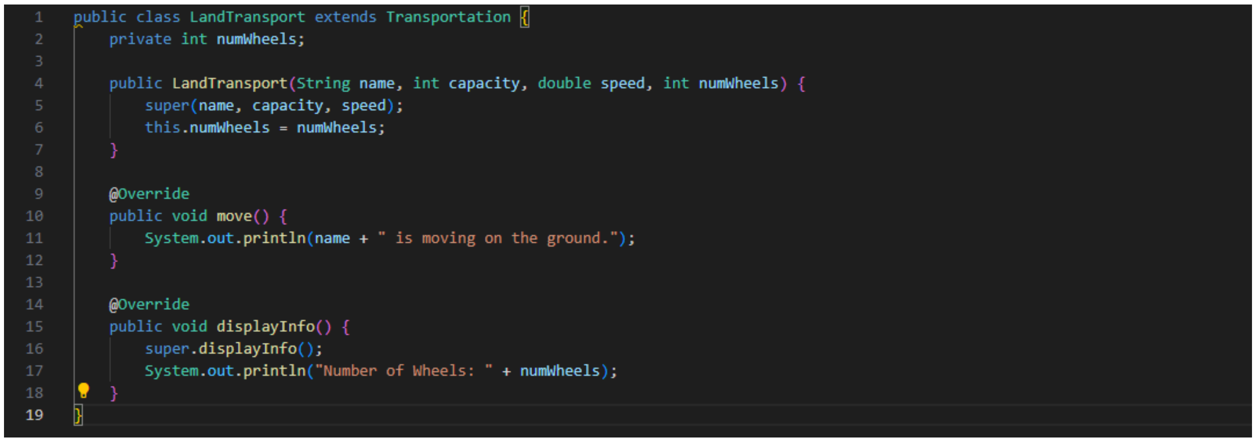
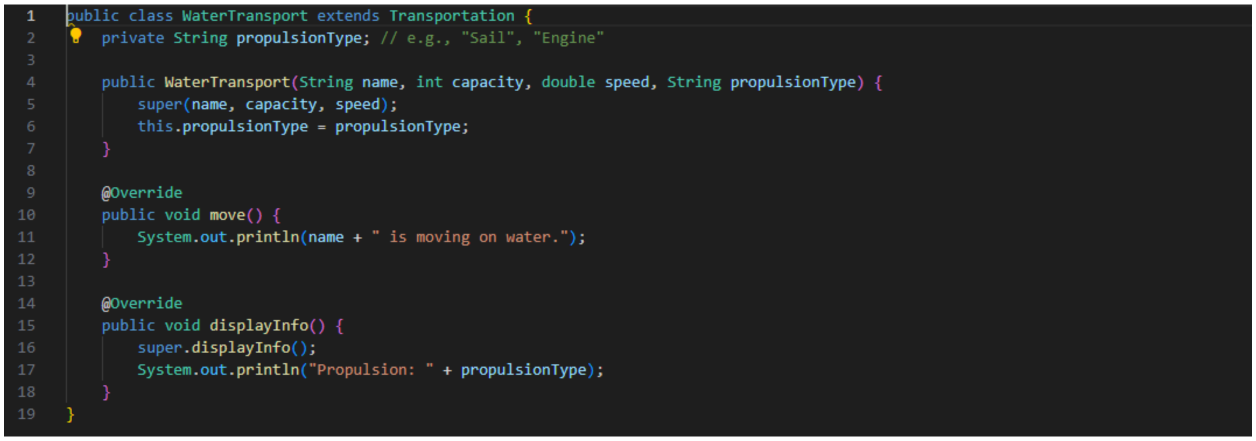
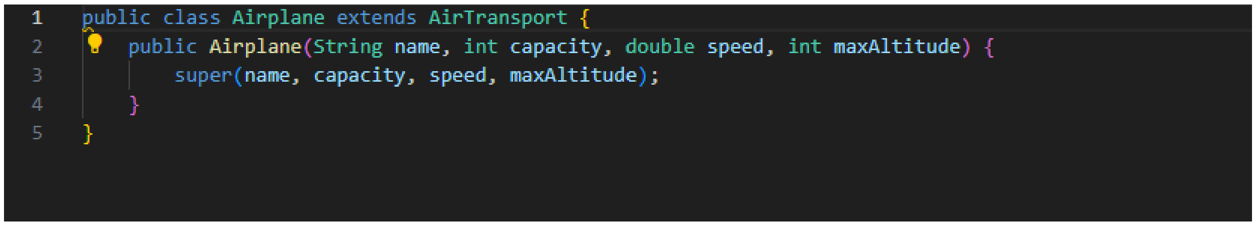

```java
public class Transportation {
    // Common properties for all transportation types
    protected String name;
    protected int capacity;
    protected double speed; // in km/h

    // Constructor to initialize the properties
    public Transportation(String name, int capacity, double speed) {
        this.name = name;
        this.capacity = capacity;
        this.speed = speed;
    }

    // A general method to display information
    public void displayInfo() {
        System.out.println("Name: " + name);
        System.out.println("Capacity: " + capacity + " people");
        System.out.println("Speed: " + speed + " km/h");
    }

    // A general method for movement
    public void move() {
        System.out.println(x:"The transportation is moving.");
    }
}
```


```java
public class AirTransport extends Transportation {
    private int maxAltitude; // in feet

    public AirTransport(String name, int capacity, double speed, int maxAltitude) {
        // Call the constructor of the parent class (Transportation)
        super(name, capacity, speed);
        this.maxAltitude = maxAltitude;
    }

    // Overriding the parent's method to be more specific
    @Override
    public void move() {
        System.out.println(name + " is flying through the air.");
    }

    @Override
    public void displayInfo() {
        super.displayInfo(); // Call parent's displayInfo first
        System.out.println("Max Altitude: " + maxAltitude + " ft");
    }
}

```

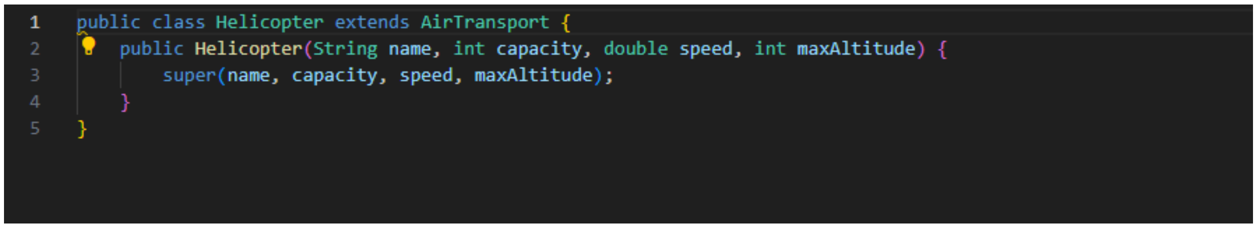

```java
public class LandTransport extends Transportation {
    private int numWheels;

    public LandTransport(String name, int capacity, double speed, int numWheels) {
        super(name, capacity, speed);
        this.numWheels = numWheels;
    }

    @Override
    public void move() {
        System.out.println(name + " is moving on the ground.");
    }

    @Override
    public void displayInfo() {
        super.displayInfo();
        System.out.println("Number of Wheels: " + numWheels);
    }
}
```



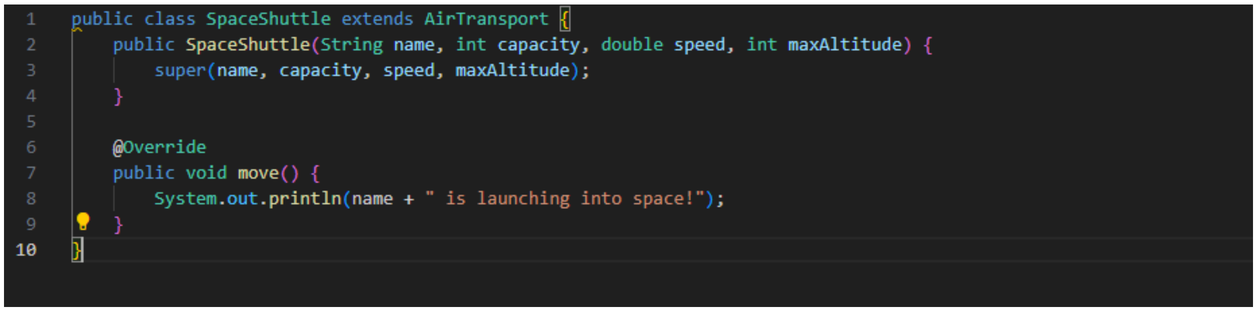
```java
public class WaterTransport extends Transportation {
    private String propulsionType; // e.g., "Sail", "Engine"

    public WaterTransport(String name, int capacity, double speed, String propulsionType) {
        super(name, capacity, speed);
        this.propulsionType = propulsionType;
    }

    @Override
    public void move() {
        System.out.println(name + " is moving on water.");
    }

    @Override
    public void displayInfo() {
        super.displayInfo();
        System.out.println("Propulsion: " + propulsionType);
    }
}
```



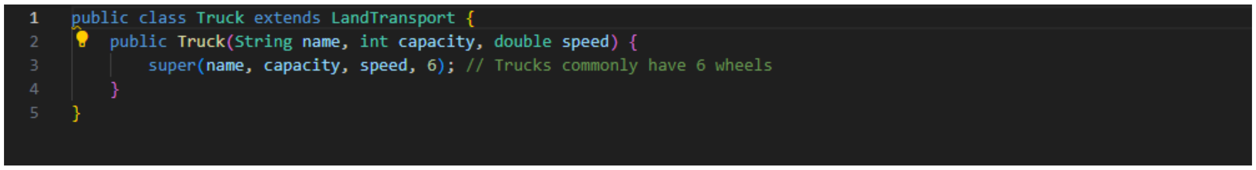
```java
public class Airplane extends AirTransport {
    public Airplane(String name, int capacity, double speed, int maxAltitude) {
        super(name, capacity, speed, maxAltitude);
    }
}
```



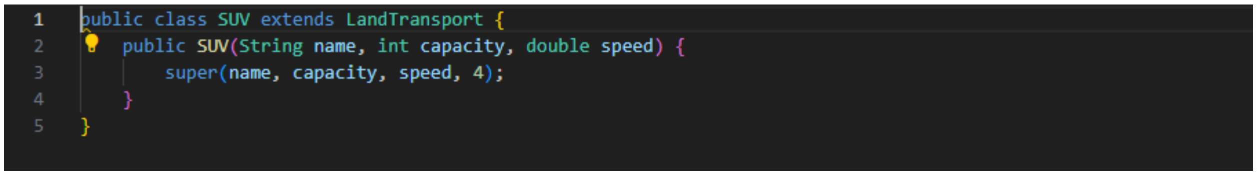
```java
public class Helicopter extends AirTransport {
    public Helicopter(String name, int capacity, double speed, int maxAltitude) {
        super(name, capacity, speed, maxAltitude);
    }
}
```



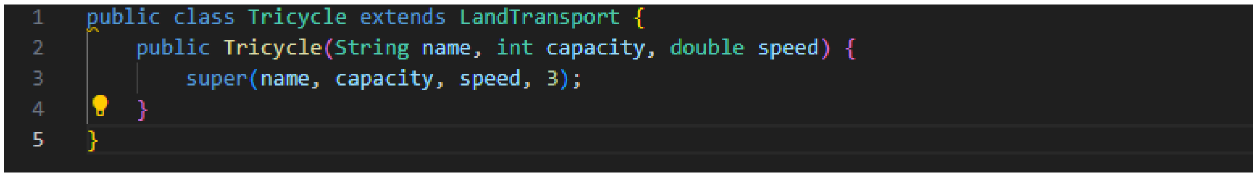
```java
public class SpaceShuttle extends AirTransport {
    public SpaceShuttle(String name, int capacity, double speed, int maxAltitude) {
        super(name, capacity, speed, maxAltitude);
    }

    @Override
    public void move() {
        System.out.println(name + " is launching into space!");
    }
}
```



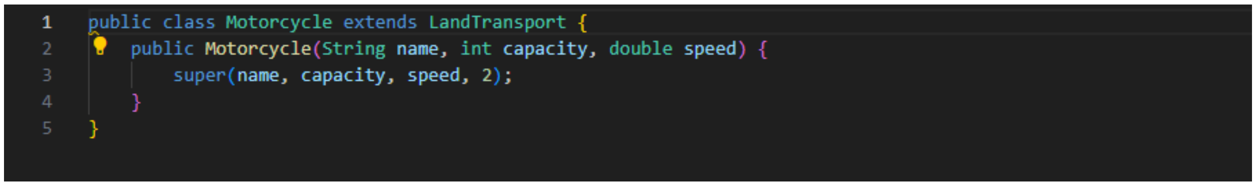
```java
public class Truck extends LandTransport {
    public Truck(String name, int capacity, double speed) {
        super(name, capacity, speed, 6); // Trucks commonly have 6 wheels
    }
}
```



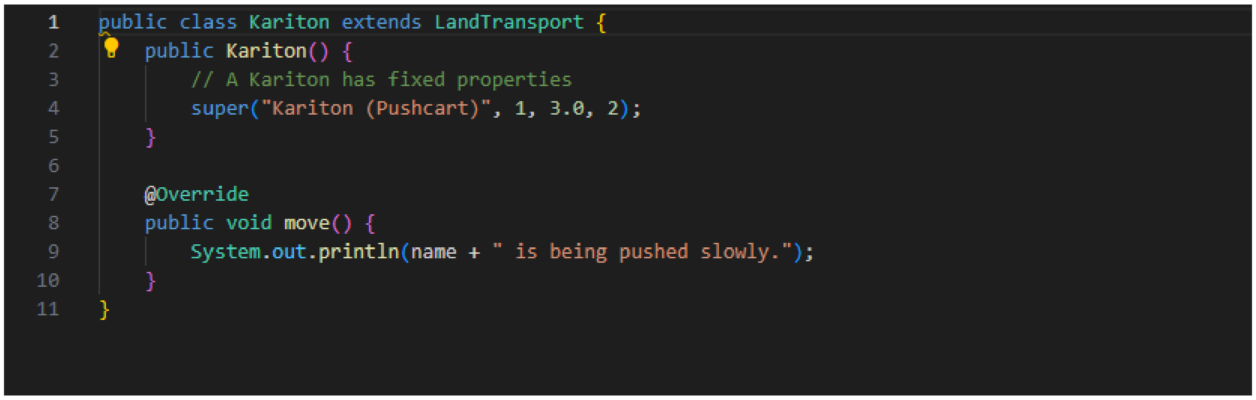
```java
public class SUV extends LandTransport {
    public SUV(String name, int capacity, double speed) {
        super(name, capacity, speed, 4);
    }
}
```



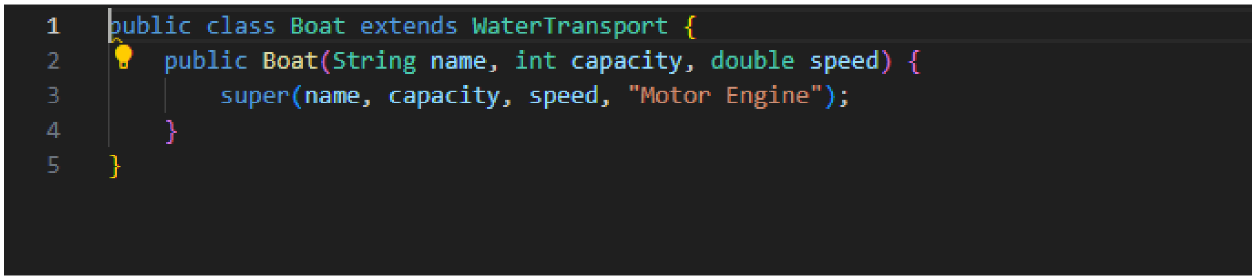
```java
public class Tricycle extends LandTransport {
    public Tricycle(String name, int capacity, double speed) {
        super(name, capacity, speed, 3);
    }
}
```



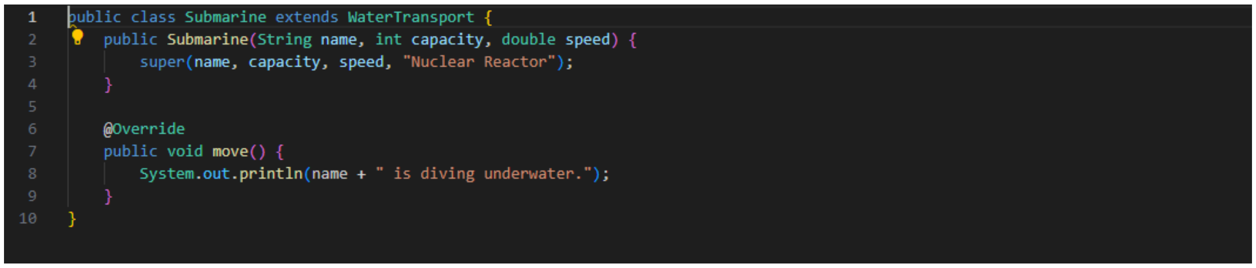
```java
public class Motorcycle extends LandTransport {
    public Motorcycle(String name, int capacity, double speed) {
        super(name, capacity, speed, 2);
    }
}
```



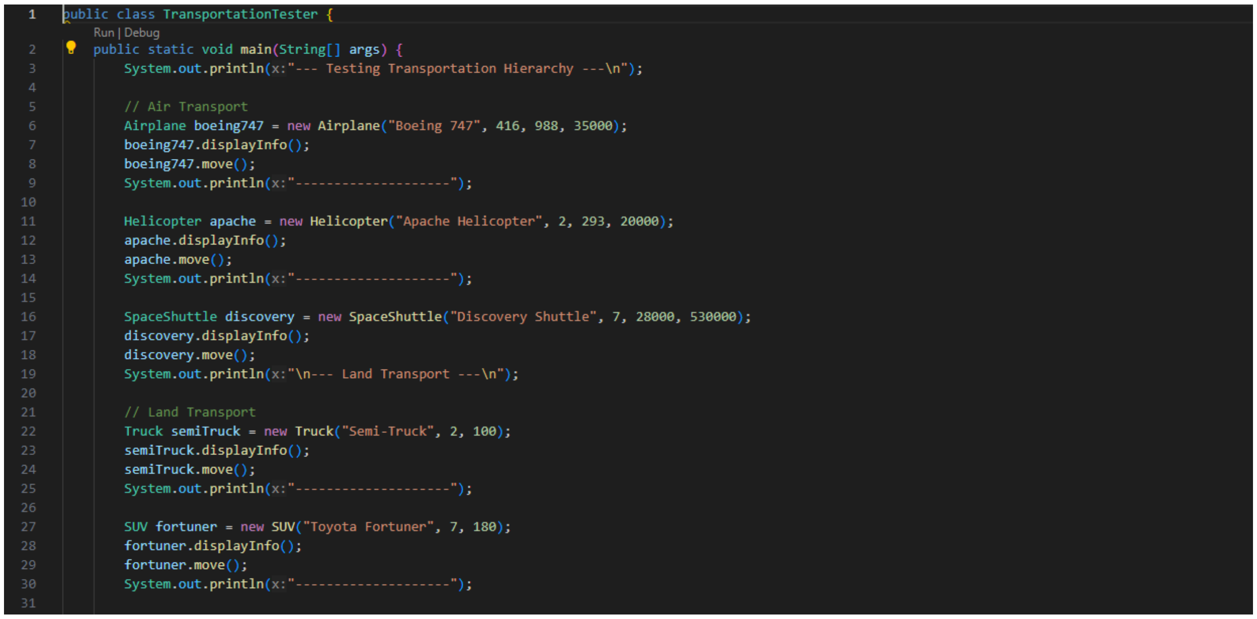
```java
public class Kariton extends LandTransport {
    public Kariton() {
        // A Kariton has fixed properties
        super("Kariton (Pushcart)", 1, 3.0, 2);
    }

    @Override
    public void move() {
        System.out.println(name + " is being pushed slowly.");
    }
}
```



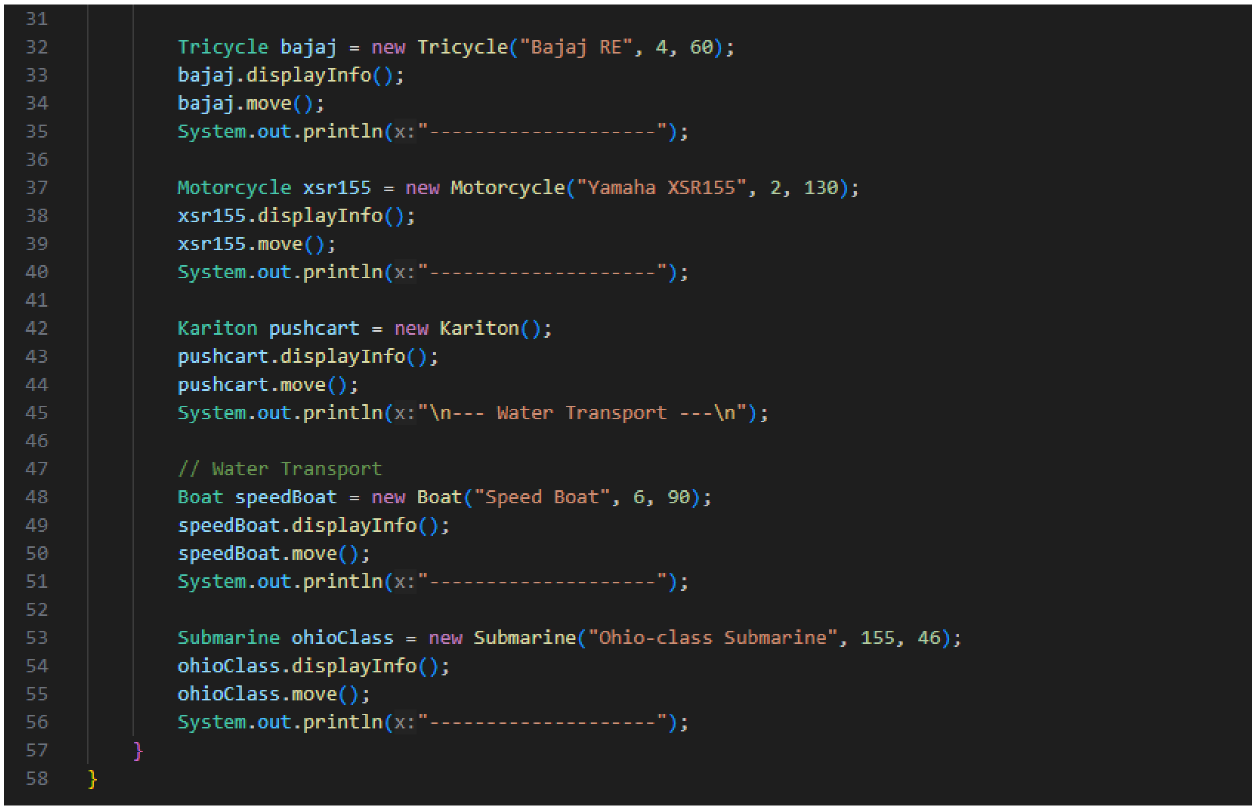
```java
public class Boat extends WaterTransport {
    public Boat(String name, int capacity, double speed) {
        super(name, capacity, speed, "Motor Engine");
    }
}
```



```java
public class Submarine extends WaterTransport {
    public Submarine(String name, int capacity, double speed) {
        super(name, capacity, speed, "Nuclear Reactor");
    }

    @Override
    public void move() {
        System.out.println(name + " is diving underwater.");
    }
}
```



```java
public class TransportationTester {
    public static void main(String[] args) {
        System.out.println(x:"--- Testing Transportation Hierarchy ---\n");

        // Air Transport
        Airplane boeing747 = new Airplane("Boeing 747", 416, 988, 35000);
        boeing747.displayInfo();
        boeing747.move();
        System.out.println(x:"--------------------");

        Helicopter apache = new Helicopter("Apache Helicopter", 2, 293, 20000);
        apache.displayInfo();
        apache.move();
        System.out.println(x:"--------------------");

        SpaceShuttle discovery = new SpaceShuttle("Discovery Shuttle", 7, 28000, 530000);
        discovery.displayInfo();
        discovery.move();
        System.out.println(x:"\n--- Land Transport ---\n");

        // Land Transport
        Truck semiTruck = new Truck("Semi-Truck", 2, 100);
        semiTruck.displayInfo();
        semiTruck.move();
        System.out.println(x:"--------------------");

        SUV fortuner = new SUV("Toyota Fortuner", 7, 180);
        fortuner.displayInfo();
        fortuner.move();
        System.out.println(x:"--------------------");

```



```java

        Tricycle bajaj = new Tricycle("Bajaj RE", 4, 60);
        bajaj.displayInfo();
        bajaj.move();
        System.out.println(x:"--------------------");

        Motorcycle xsr155 = new Motorcycle("Yamaha XSR155", 2, 130);
        xsr155.displayInfo();
        xsr155.move();
        System.out.println(x:"--------------------");

        Kariton pushcart = new Kariton();
        pushcart.displayInfo();
        pushcart.move();
        System.out.println(x:"\n--- Water Transport ---\n");

        // Water Transport
        Boat speedBoat = new Boat("Speed Boat", 6, 90);
        speedBoat.displayInfo();
        speedBoat.move();
        System.out.println(x:"--------------------");

        Submarine ohioClass = new Submarine("Ohio-class Submarine", 155, 46);
        ohioClass.displayInfo();
        ohioClass.move();
        System.out.println(x:"--------------------");
    }
}
```