

Module-8 Modular Programming

```
package modularProramming;

//Create a simple Java class with composition, encapsulating functionalities using
other classes.
//Implement getter and setter methods for nested objects.

//Engine class with properties and methods
class Engine {
    private String type;
    private int horsepower;

    public Engine(String type, int horsepower) {
        this.type = type;
        this.horsepower = horsepower;
    }

    // Getter and setter for type
    public String getType() {
        return type;
    }

    public void setType(String type) {
        this.type = type;
    }

    // Getter and setter for horsepower
    public int getHorsepower() {
        return horsepower;
    }

    public void setHorsepower(int horsepower) {
        this.horsepower = horsepower;
    }

    public void start() {
        System.out.println("The engine is starting...");
    }
}

//Wheel class with properties and methods
class Wheel {
    private int size;
    private String material;

    public Wheel(int size, String material) {
        this.size = size;
        this.material = material;
    }

    // Getter and setter for size
    public int getSize() {
        return size;
    }
}
```

```

public void setSize(int size) {
    this.size = size;
}

// Getter and setter for material
public String getMaterial() {
    return material;
}

public void setMaterial(String material) {
    this.material = material;
}

public void rotate() {
    System.out.println("The wheel is rotating...");
}
}

//Car class with composition of Engine and Wheel
class Car {
    private Engine engine;
    private Wheel wheel;

    // Constructor for Car, initializing nested objects
    public Car(Engine engine, Wheel wheel) {
        this.engine = engine;
        this.wheel = wheel;
    }

    // Getter and setter for engine
    public Engine getEngine() {
        return engine;
    }

    public void setEngine(Engine engine) {
        this.engine = engine;
    }

    // Getter and setter for wheel
    public Wheel getWheel() {
        return wheel;
    }

    public void setWheel(Wheel wheel) {
        this.wheel = wheel;
    }

    public void drive() {
        engine.start();
        wheel.rotate();
        System.out.println("The car is driving...");
    }
}

//Main class to demonstrate composition and encapsulation
public class Assignment1Main {
    public static void main(String[] args) {
        // Create Engine and Wheel objects
        Engine engine = new Engine("V8", 450);
    }
}

```

```
Wheel wheel = new Wheel(18, "Alloy");

// Create Car object with Engine and Wheel
Car car = new Car(engine, wheel);

// Access and modify nested objects via getters and setters
System.out.println("Car Engine: " + car.getEngine().getType() + " with " +
car.getEngine().getHorsepower() + " HP");
System.out.println("Car Wheel: " + car.getWheel().getSize() + " inch " +
car.getWheel().getMaterial() + " wheel");

// Call methods of the Car (which in turn calls methods of Engine and Wheel)
car.drive();
}
}
```