Task 1: Create a Basic Inheritance Structure

Description: Create a superclass named 'Vehicle' with basic attributes and methods. Then, create a subclass named 'Car' that inherits from 'Vehicle'.

Solution:

```
class Vehicle {
    String type;
    void startEngine() {
        System.out.println("Engine started");
    }
}
class Car extends Vehicle {
    int wheels = 4;
}
```

Task 2: Overriding Methods

Description: In the 'Car' class, override the 'startEngine' method to customize the output.

Solution:

```
class Car extends Vehicle {
    @Override
    void startEngine() {
        System.out.println("Car engine started");
    }
}
```

Task 3: Using Protected Access Modifier

Description: Add a protected attribute to the 'Vehicle' class and access it from the 'Car' subclass.

Solution:

```
class Vehicle {
    protected String brand = "Generic";
}
```

```
class Car extends Vehicle {
    void displayBrand() {
        System.out.println("Brand: " + brand);
    }
}
```

Task 4: Package Access Level

Description: Create two classes in the same package, where one class accesses a method of the other class without any access modifier (default/package level access).

Solution:

```
class FirstClass {
    void displayMessage() {
        System.out.println("Hello from FirstClass");
    }
}
class SecondClass {
    void accessFirstClass() {
        FirstClass obj = new FirstClass();
        obj.displayMessage();
    }
}
```

Task 5: Understanding the Object Class

Description: Demonstrate the use of getClass() and instanceof in identifying the class of an object.

Solution:

```
class Demo {
    public static void main(String[] args) {
        Car car = new Car();
        if (car instanceof Vehicle) {
            System.out.println("Car is a Vehicle");
        }
```

```
        System.out.println("Class of car: " + car.getClass().getName());
    }
}
```