Assignment No: 01

Department of Computer Science

Iqra University Islamabad

Object Oriented Programming (Lab)

Maqsood Ahmed

ID: 38186

Problem # 1

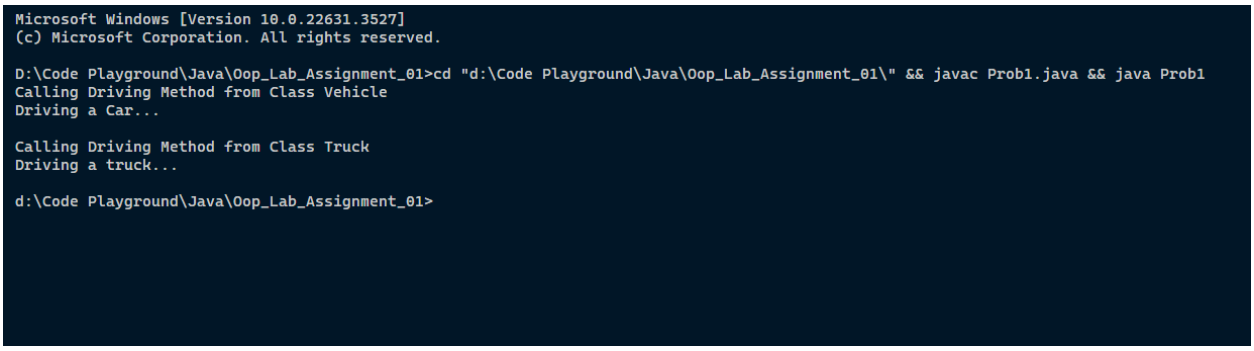
Create a base class Vehicle with a method drive(). Create a subclass Car that overrides the drive() method to print "Driving a car." Create another subclass Truck that overrides the drive() method to print "Driving a truck.".

Source Code:

```
public class Prob1 {
    public static void main(String args[]) {
        System.out.println("Calling Driving Method from Class Vehicle");
        Vehicle vehicle = new Vehicle();
        vehicle.drive();


        System.out.println("\nCalling Driving Method from Class Truck");
        Truck truck = new Truck();
        truck.drive();
    }
}


class Vehicle {
    public void drive() {
        System.out.println("Driving a Vehicle...");
    }
}
class Car {
@Override
    public void drive() {
        System.out.println("Driving a Car...");
    }
}
```

```
class Truck extends Vehicle {
    @Override
    public void drive() {
        System.out.println("Driving a truck...");
    }
}
```

OUTPUT:



Problem # 2

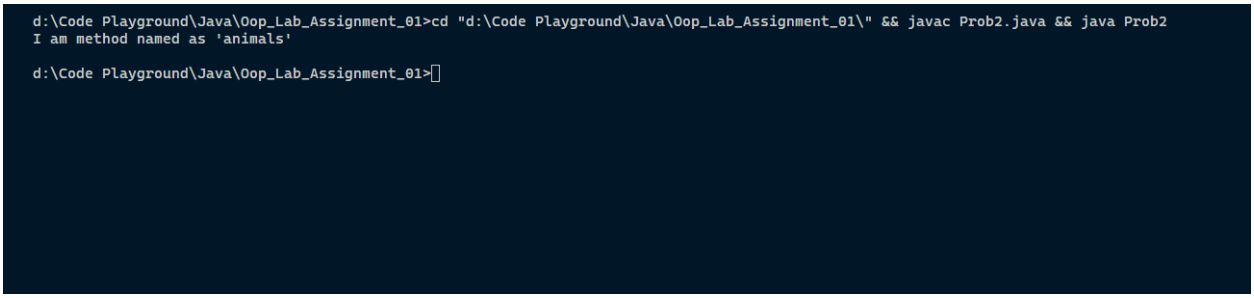
Create a class Zoo with a protected method Animals(). Create a subclass Lion in a different package that attempts to call the Animals() method. Investigate and explain what happens.

Source Code:

```
package package1;
public class Zoo {
    protected void Animals() {
        System.out.println("I am a method named as ‘animals’ ");
    }
}
```

```
package package2;

import package1.Zoo;

public class Lion extends Zoo {
    public void showAnimals() {
        Animals();
    }
}

import package2.Lion;

public class Main {
    public static void main(String[] args) {
        Lion myLion = new Lion();
        myLion.showAnimals();    }
}
```

OUTPUT:



Problem # 3

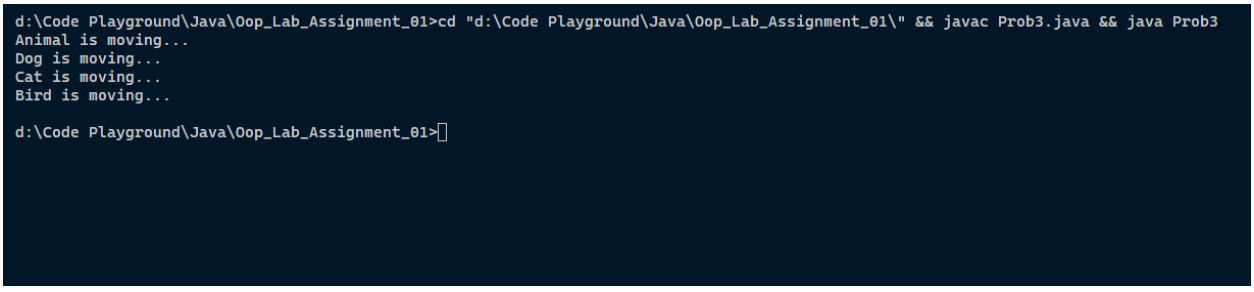
Create a class Animal with a method move(). Create subclasses Dog, Cat, and Bird that inherit from Animal and override the move() method with appropriate behaviors. Demonstrate polymorphism by creating an array of Animal references and calling move() on each element.

Source Code:

```
public class Prob3 {
    public static void main(String args[]) {

        Animal animal = new Animal();
        animal.move();

        Dog dog = new Dog();
        dog.move();

        Cat cat = new Cat();
        cat.move();

        Bird bird = new Bird();
        bird.move();

    }
}

class Animal {
    public void move() {
        System.out.println("Animal is moving...");
    }
}
```

```
class Dog extends Animal{
    @Override
    public void move() {
        System.out.println("Dog is moving...");
    }
}
class Cat extends Animal{
    @Override
    public void move() {
        System.out.println("Cat is moving...");
    }
}
class Bird extends Animal{
    @Override
    public void move() {
        System.out.println("Bird is moving...");
    }
}
```

OUTPUT:



The End