**PSG COLLEGE OF TECHNOLOGY**

**DEPARTMENT OF COMPUTER APPLICATIONS**

**23MX26 - Java Programming Laboratory**

**Hands-on Worksheet 10 Analysis Practice Problems**

1. Consider the following two classes:

public class Car {

public void m1() {

System.out.println("car 1");

}

​

public void m2() {

System.out.println("car 2");

}

​

public String toString() {

return "vroom";

}

}

public class Truck extends Car {

public void m1() {

System.out.println("truck 1");

}

}

And assuming that the following variables have been declared:

Car mycar = new Car();

Truck mytruck = new Truck();

What is the output from the following statements?

Top of Form

|  |  |
| --- | --- |
| System.out.println(mycar); |  |
| mycar.m1(); |  |
| mycar.m2(); |  |
| System.out.println(mytruck); |  |
| mytruck.m1(); |  |
| mytruck.m2(); |  |

Bottom of Form

**2. Consider the following two classes:**

public class Car {

public void m1() {

System.out.println("car 1");

}

​

public void m2() {

System.out.println("car 2");

}

​

public String toString() {

return "vroom";

}

}

public class Truck extends Car {

public void m1() {

System.out.println("truck 1");

}

public void m2() {

super.m1();

}

public String toString() {

return super.toString() + super.toString();

}

}

And assuming that the following variable has been declared:

Truck mytruck = new Truck();

What is the output from the following statements?

Top of Form

|  |  |
| --- | --- |
| System.out.println(mytruck); |  |
| mytruck.m1(); |  |
| mytruck.m2(); |  |

Bottom of Form

**3. Suppose that the following two classes have been declared:**

public class Car {

public void m1() {

System.out.println("car 1");

}

​

public void m2() {

System.out.println("car 2");

}

​

public String toString() {

return "vroom";

}

}

public class Truck extends Car {

public void m1() {

System.out.println("truck 1");

}

public void m2() {

super.m1();

}

public String toString() {

return super.toString() + super.toString();

}

}

Write a class MonsterTruck whose methods have the behavior below. Don't just print/return the output; whenever possible, use inheritance to reuse behavior from the superclass.

|  |  |
| --- | --- |
| **Method** | **Output/Return** |
| m1 | monster 1 |
| m2 | truck 1  car 1 |
| toString | "monster vroomvroom" |

**4. Define a Critter class named Butterfly with the following behavior:**

|  |  |
| --- | --- |
| **constructor** | public Butterfly() |
| **color** | yellow (Color.YELLOW) |
| **eating behavior** | never eats (this is the default behavior) |
| **fighting behavior** | always forfeits (this is the default behavior) |
| **movement behavior** | moves N, W, N, E, then repeats |
| **toString** | alternates between "x" and "-" on each move |