



Course : Object oriented programming
Tutorial N°6 : Java inheritance and abstract classes

Exercise N1:

Let the following two classes be:

```
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");
    }
}
```

1- What is the result of the following instructions:

```
Truck mycar = new Truck();
System.out.println(mycar);
mycar.m1();
mycar.m2();
```

2- Assuming the Truck class has changed to the following:

```
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();
    }
}
```

What will be the result of the following instructions:

```
Truck mycar = new Truck();
System.out.println(mycar);
mycar.m1();
mycar.m2();
```

Exercise N1:

Assuming still the two classes `Car` and `Truck` from the first exercise, write a class `MonsterTruck` that inherits from the `Truck` class and produces the following output:

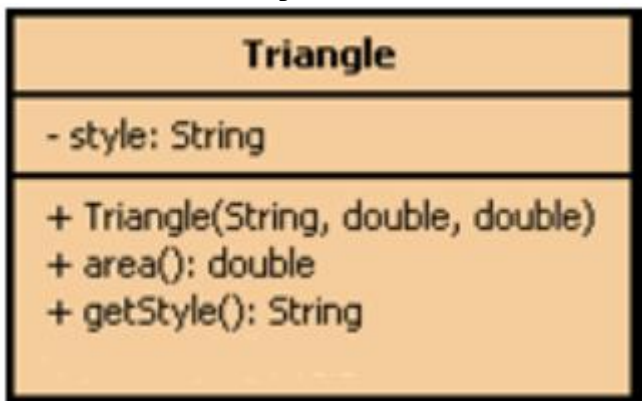
```
MonsterTruck bigfoot = new MonsterTruck();
bigfoot.m1(); // monster 1
bigfoot.m2(); // truck 1 / car 1
System.out.println(bigfoot); // monster vroomvroom
```

Exercise N3:

Let the following class **TwoDShape** (2D geometric shape) be:

```
class TwoDShape {  
    private double width;  
    private double height;  
    public TwoDShape(double width, double height) {  
        this.width = width;  
        this.height = height;  
    }  
    public double getWidth() {  
        return width;  
    }  
    public double getHeight() {  
        return height;  
    }  
    public void print() {  
        System.out.println("Width is " + width + " Height is " + height);  
    }  
}
```

And let the class **Triangle** inherit from the class **TwoDShape**.



```
// area = w * h / 2
```

```
// The style can be: isosceles, equilateral, or scalene
```

```
// getStyle returns the style.
```

- 1- Provide the corresponding **JAVA code** for the Triangle class.
- 2- Rewrite the TwoDShape class with an **abstract method** called area, then implement this method in the Triangle class that inherits from TwoDShape. Additionally, both classes should use the toString method to display their dimensions (the toString method of the Triangle class should also display the style of the triangle).