CSIT111 Lab Exercises

Lab 7

Objectives

- Java code analysis and debugging
- Java classes and objects
- Object constructors
- Local variables, fields, methods

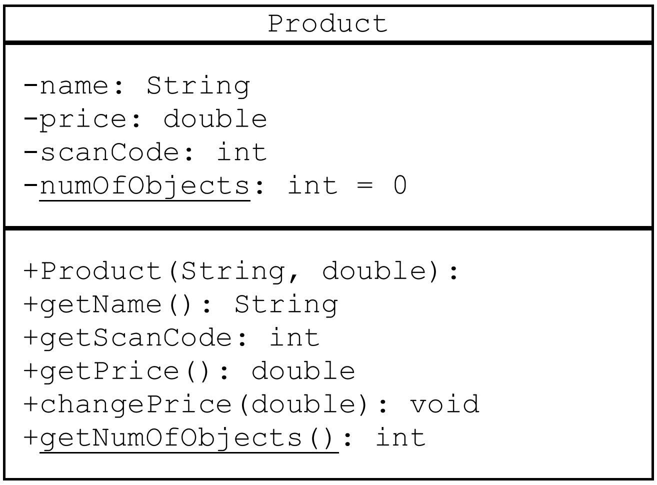
Exercise 1: Analyse, compile and fix bugs in the following program

```
class Circle
{
        private double r;

        public void Circle( double r )
        {
            r = r;
        }

        private double calCircumference()
        {
            return 2*Math.PI*r;
        }

        public static double calArea()
        {
            return Math.PI*r*r;
        }
}


class CircleApp
{
        public static void main( String[] args )
        {
                double rd = Double.parseDouble( args[0] );

                System.out.println( "Circle radius = " + rd );

                // create an object of Circle with the radius rd
                Circle circle1 = new Circle( rd );

                double cir = circle1.calCircumference();
                double area = circle1.calArea();

                System.out.println("Circumference = " + cir);
                System.out.println("Area = " + area);
        }
}
```

Exercise 2: Update the `Circle` class definition. You need to:

- Add a copy constructor to the class `Circle`.
- Add a public method `setRadius( double r )`
- Add a public method `double getRadius()`
- Create Object `circle2 as` a clone of `circle1` in the `main()`. Display its circumference and area.
- Use `setRadius(..)` to change radius of `circle1`. Does it affect `circle2`? Prove your answer by displaying relevant values.
- Declare another object reference variable
  `Circle circle3 = circle2;`
  Display the radius of `circle2` and `circle3`. Are these values the same?
  Change the radius of `circle3` to a different value. Display the radius of `circle2` and `circle3` again. As the radius of `circle2` has not been changed, it should be different from the radius of `circle3` that has been changed. Are they different? Explain the result.

Exercise 3: Implement a class according to its UML class symbol:



```
Product

-name: String
-price: double
-scanCode: int
-numOfObjects: int = 0

+Product(String, double):
+getName(): String
+getScanCode: int
+getPrice(): double
+changePrice(double): void
+getNumOfObjects(): int
```

The constructor shall take two parameters to initialise the `name` and `price` data fields.
The `scanCode` is a 4-digit integer value `1000+numOfObjects` that shall be calculated and initialised by the constructor.
The variable `numOfObjects` shall be incremented by the constructor every time a new object of type `Product` is instantiated.

Test the defined class with the following Java program:

```
class TestProduct {
     public static void main(String[] args) {
        System.out.println("There are " + Product.getNumOfObjects() + “ items.”);

        Product pr1 = new Product("Computer", 1500.0);
        System.out.println("There are " + Product.getNumOfObjects() + “ items.”);

        Product pr2 = new Product("Printer", 600.0);
        Product pr3 = new Product("Monitor", 240.0);
        System.out.println("There are " + Product.getNumOfObjects() + “ items.”);

        pr2.changePrice( 549.99 );

        System.out.println(" -- Product info -- ");
        System.out.println( "Name: " + pr2.getName() );
        System.out.println( "Scan code: " + pr2.getScanCode() );
        System.out.println( "Price: " + pr2.getPrice() );
     }
}
```

1. Download the TestProduct.java from Moodle (in the same folder as the Lab exercise);
2. Make sure to save TestProduct.java and Product.java in the same folder;
3. Compile the two java programs, “javac TestProduct.java Product.java”;
4. Execute the TestProduct program, “java TestProduct” (why the program can’t be executed by using “java Product”).