National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

Department of Computing

CS 212: Object Oriented Programming

Class: BSCS-6ABC

Lab 03: Fundamentals of OOP

Date: February 28th, 2017

Instructor:

Dr. Mian M. Hamayun / Dr. Anis ur Rahman

Learning Objectives

The learning objectives of this lab is to understand and practice the fundamentals of object-oriented programming, such as classes and objects, access specifiers, constructors, setters/getters, static members and passing objects to methods.

Activity #1.

What is wrong in the following code? Please rectify each, and provide a separate source file.

```
public class Test1
{
    public static void main(String[] args)
    {
        Test1 t1 = new Test1(5);
    }
}
```

```
public class Test2
{
    public static void main(String[] args)
    {
        Test2 t2 = new Test2();
        t2.x();
    }
}
```

```
public class Test3
{
    Public void method1()
    {
        Circle c;
        System.out.println("Radius is " + c.getRadius());
        c = new Circle();
    }
}
```

```
public class Test4
{
    public static void main(String[] args)
    {
        C c = new C(5.0);
        System.out.println(c.value);
    }
}

class C
{
    int value = 2;
}
```

```
public class Circle
{
    private double radius = 1.0;

    /** Find the area of this circle */
    public double getArea()
    {
        return radius * radius * Math.PI;
    }
}

public class Test5
{
    public static void main(String[] args)
    {
        Circle myCircle = new Circle();
        System.out.println("Radius is "+ myCircle.radius);
    }
}
```

Activity #2.
Can you invoke an instance method or reference an instance variable from a static method? Can you invoke a static method or reference a static variable from an instance method? What is wrong in the following code? Please correct it.

```
public class Example
{
    public static void main(String[] args)
    {
        method1();
    }

    public void method1()
    {
        method2();
    }

    public static void method2()
    {
        System.out.println("What is radius " + c.getRadius());
    }

    Circle c = new Circle();
 }
```

Activity #3.
What is the difference between passing a parameter of a primitive type and passing a parameter of a reference type? Show the output of the following program (*you may provide your answer and the output in commented form at the top of the program's source file*):

```
public class Count
{
    public int count;

    public Count(int c)
    {
        count = c;
    }

    public Count()
    {
        count = 1;
    }
}
```

```
public class Test
{
    public static void main(String[] args)
    {
        Count myCount = new Count();
        int times = 0;

        for (int i = 0; i < 100; i++)
            increment(myCount, times);

        System.out.println("count is " + myCount.count);
        System.out.println("times is " + times);
    }

    public static void increment(Count c, int times)
    {
        c.count++;
        times++;
    }
}
```

Activity #4.

Show the output of the following program (*you may provide the output in commented form at the top of the program's source file*):

```
public class Test
{
    public static void main(String[] args)
    {
        Circle circle1 = new Circle(1);
        Circle circle2 = new Circle(2);

        swap1(circle1, circle2);
        System.out.println("After swap1: circle1 = " +
            circle1.radius + " circle2 = " + circle2.radius);

        swap2(circle1, circle2);
        System.out.println("After swap2: circle1 = " +
            circle1.radius + " circle2 = " + circle2.radius);
    }

    public static void swap1(Circle x, Circle y)
    {
        Circle temp = x;
        x = y;
        y = temp;
    }

    public static void swap2(Circle x, Circle y)
    {
        double temp = x.radius;
        x.radius = y.radius;
        y.radius = temp;
    }
}

class Circle
{
    double radius;

    Circle(double newRadius)
    {
        radius = newRadius;
    }
}
```

You may like to have a look at: http://www.geeksforgeeks.org/swap-exchange-objects-java/

Task #1:
Design a class named `Rectangle` to represent a rectangle. The class contains:

- Two `double` data fields named `width` and `height` that specify the width and height of the rectangle. The default values are `1` for both `width` and `height`.
- A no-arg constructor that creates a default rectangle.
- A constructor that creates a rectangle with the specified `width` and `height`.
- A method named `getArea()` that returns the area of this rectangle.
- A method named `getPerimeter()` that returns the perimeter.

Draw the UML diagram for the class. Implement the class. Write a test program that creates two `Rectangle` objects—one with width 4 and height 40 and the other with width 3.5 and height 35.9. Display the width, height, area, and perimeter of each rectangle in this order.

Task #2:
Create a class called `Employee` that includes three instance variables—a first name (type `String`), a last name (type `String`) and a monthly salary (`double`). Provide a constructor that initializes the three instance variables. Provide a *set* and a *get* method for each instance variable. If the monthly salary is not positive, it should be set to 0.

Write a test application named `EmployeeTest` that demonstrates class `Employee`'s capabilities. Create two `Employee` objects and display each object's yearly salary. Then give each `Employee` a 10% raise and display each `Employee`'s yearly salary again.

Task #3:
Create a class called Date that includes three instance variables—a month (type int), a day (type int) and a year (type int). Provide a constructor that initializes the three instance variables and assumes that the values provided are correct. Provide a set and a get method for each instance variable. Provide a method displayDate that displays the month, day and year separated by forward slashes (/). Write a test application named DateTest that demonstrates class Date's capabilities.

Hand in

Hand in the source code from this lab at the appropriate location on the LMS system. You should hand in a single compressed/archived file named Lab_2_<Your CMS_ID. Your_NAME >.zip (without angle brackets) that contains ONLY the following files.

1) All completed java source files representing the work accomplished for this lab: ActivityOne(Test1.java, Test2.java, Test3.java, Test4.java, Test5.java); ActivityTwo.java; ActivityThree.java; ActivityFour.java; Task1.java, Task2.java. The files should contain author in the comments at the top.

2) A plain text file named **README.TXT** that includes a) author information at the beginning, b) a brief explanation of the lab, and c) any comments, or suggestions.

To Receive Credit

1. By showing up on time for lab, working on the lab solution, and staying to the end of the class period, only then you can receive full credit for the lab assignment.

2. Comment your program heavily. Intelligent comments and a clean, readable formatting of your code account for 20% of your grade.

3. The lab time is not intended as free time for working on your programming/other assignments. Only if you have completely solved the lab assignment, including all challenges, and have had your work checked off for completeness by your TA/Lab Engineer should you begin the programming/other assignments.