Track 2 ~ Java Basics

Exercise 1: Pseudocoding, Classes, Identifiers, Data types, Methods *Instructions: This is a hands-on graded exercise. Follow the instructions given below. Apply Java Coding Standards. Don't forget to write your pseudocode before coding on Eclipse IDE.*	Hours: 4.0 hours Name your project as <FamilyName>Ex1

A. Correct the following codes by applying the proper *Java Coding Standards* and fixing the syntax errors using the Eclipse IDE. Place all the corrected classes inside **org.acumen.training.codes.itema**.

Erroneous Code 1

```
public class Oops {
    public static void main(String args[]) {
        int a = 7, b = 42;
        int smaller = minimum(a, b);
        if (smaller == a) {
            System.out.println("a is the smallest!");
        } else if (smaller == b) {
            System.out.println("b is the smallest!");
        }
    }

    /*
         returns which int is smaller
    */
    public static int minimum(int a, int b) {
        int smaller = 0;
        if (a < b) {
            smaller = a;
        } else if (a >= b) {
            smaller = b;
        }
        return smaller;
    }
}
```

Erroneous Code 2

```
public class LotsOfErrors {
  public static void main(String[] args) {
     System.out.println("Hello, world!”);
     message();
   }

   public static void message() {
      System.out.println("This program surely cannot");
      System.out.println("have any so-called \"errors\" in it");
    }
 }
```

Erroneous Code 3

```
 public class Temperature {
   public static void main(String[] args) {
         double tempf = 98.6;
         double tempc = 0.0;
         tempc = ftoc(tempf);
         System.out.println("Body temp in C is: " + tempc);
   }

   /* Converts Fahrenheit temperatures to Celsius.
    */
   public static double ftoc(double tempf) {
         return (tempf - 32) * 5 / 9;
    }
}
```

Erroneous Code 4

```
public class problems
{
 public Static main (string[] args)
 {
 System.out.println ("!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!");
 System.out.println (This program used to have lots of problems,");
 System.Out.println ("but if it prints this, you fixed them all.")
 System.out.println (" *** Hurray! ***);
 System.out.println ("!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!");
 }
}
```

Erroneous Code 5

```
public class Parameters {
    public static void main() {
        double bubble = 867.5309;
        double x = 10.01;
        printer(double x, double y);
        printer(x);
        printer("barack", "obama");
        System.out.println("z = " + z);
    }

    public static void printer(x, y double) {
        int z = 5;
        System.out.println("x = " + double x + " and y = " + y);
        System.out.println("The value from main is: " + bubble);
    }
}
```

Erroneous Code 6

```
public class exercise {
        private int x = 101;
        private void f(int x) {
             x++;
             g();
        }
        private void g() {
             System.out.println(x);
        }
        public static void main(String[] args) {
             exercise e = new exercise();
             int x = 200;
             e.f(x);
        }
}
```

Erroneous Code 7

```
public class Tricky
    public static main(String args) {
        System.out.println(Hello world);
        system.out.Pritnln("Do you like this program"?);
        System.out.println()

        System.println("I wrote it myself.";
    {
}
```

Erroneous Code 8

```
public class Oops {
    public static void main(String[] args) {
        int x;
        System.out.println("x is" x);

        int x = 15.2;   // set x to 15.2
        System.out.println("x is now + x");

        int y;          // set y to 1 more than x
        y = int x + 1;
        System.out.println("x and y are " + x + and + y);
    }
}
```

B. Study the following problem:

Before implementing your solution in Java syntax, write the pseudocode of a class **SalesTax** that has a method **computeCost(itemCost)** that will compute the amount of state sales tax due, and the amount of county sales tax due. The state sales tax rate is constant at **4%**. The county sales tax rate is constant at **2%**.

The program should then display:

- The cost of the item purchased
- The amount of state sales tax due.
- The amount of county sales tax due
- The total amount due from the user (cost of item + amount of state sales tax + amount of county sales tax).

Create a test class **TestSalesTax** to run **computeCost(itemCost)** with varying **itemCost**.

Do the following:

1. Applying the flowcharting discusses in *Track 1*, create your initial solution in a flowchart form using *draw.io*. Save the PNG format of your flowchart in **`src/pseudocode`**.
2. Implement your flowchart using Java syntax discussed today from declaration of variables, choosing the right data types, and writing the proper codes. Place your solution inside **`org.acumen.training.codes.itemb`**.

C. The following program is legal under Java's syntax rules, but it is difficult to read because of its layout and lack of comments. Reformat it using the rules given in this chapter and add a comment header at the top of the program regarding what the class is supposed to do. Place the class inside **`org.acumen.training.codes.itemc`**.

```
//****************************************************************
// Place here your description
//****************************************************************
public class SumProd { public static void main(String[] args){
final int INT2=8; final int INT1=20; System.out.println(
“The sum of “ + INT1 + “ and ”
+ INT2 + “ is “ + (INT1+INT2)); System.out.println (
“Their product is “ + (INT1*INT2)); }}
```

D. Write a complete Java class **`WellDoneFormat`** that prints the following *exact* output. Place the class inside **`org.acumen.training.coded`**.

```
A well-formed Java program has a main
method with { and } braces.
A System.out.println statement has ( and )
and usually a String that starts and ends
with a " character. (But we type \" instead!)
```

E. Write a Java class **`AssignMe`** that has an entrypoint **`main()`** which performs the following steps:

a) Declare two int variables named x and y.
b) Assign 3 to x.
c) Assign twice the value of x to y.
d) Interchange the value of x and y (without explicitly assign 3 to y).
e) Print the values of both variables on screen.

Drop the class inside **`org.acumen.training.codes.itemf`**.

F. Write a Java class **`DeclareMe`** that has an entrypoint **`main()`** which performs the following steps:

a) Declare a float variable called f.
b) Declare an int variable called k.
c) Assign 22 to k.
d) Assign the value of k to f.
e) Print the values of both variables on screen.

Drop the class inside **`org.acumen.training.codes.itemg`**.

G. Write a Java class **`Fraction`** and that has instance variables **`numerator`** and **`denominator`** that can be accessible anytime during testing. It also has the following methods:

1. **`printRational()`**, that will print the fraction in the format like “3/4”.
2. **`add(Fraction fraction)`**, that returns a Fraction object after adding 2 fractions
3. **`multiply(Fraction fraction)`** ,that returns a Fraction object after multiplying 2 fractions
4. **`greaterEqual(Fraction fraction)`**, that checks the first fraction is greater than the other

Here are some sample scenarios:

- For instance, if a *Fraction (f1)* object has a **`numerator`** variable with value 10 and **`denominator`** variable with value 23, its **`f1.printRational()`** should return the string like "10/23" to the console
- If another *Fraction (f2)* has a numerator of 2 and denominator of 3, calling **`f2.multiply(f1).printRational()`** should return the String like "20/69"
- Calling **`f2.add(f1).printRational()`** should return the String like "76/69"
- Calling **`f2.greaterEqual(f1).printRational()`** should return the boolean **`true`**

Create a test class **`TestFraction`** to test and execute all these functions.