COEN 160 **OO Analysis, Design and Programming** **Winter 2017**

Assignment 1 (100 pts) **Due: (31st Jan, 10 pm)**

In this first assignment, you will learn to use a **Java IDE** (**Eclipse, NetBeans** etc)

- To create a Java program with classes, compile and run the program.
- To define classes in a package
- To define Java arrays
- To test your classes

Question 1 (50 pts)

Note: Define the classes required in questions 1 and 2, in a package with name, your *lastname*.assign1

Create a **USMoney** class with two instance data members (integers), **dollars** and **cents**. Define the following methods:

- A **constructor** that takes two parameters (*dollars* and *cents*) and initializes the corresponding data members. The constructor should check that the cents value is between 0 and 99 and if not, transfer some of the cents to the dollars to make it between and 99.
- Define a default constructor.
- Define two **setter** methods, one to set dollars (**setDollars())** and one to set cents (**setCents**()).
- Define two **getter** methods, one to return the dollars and one to return the cents.
- Define a method called **addTo** (int *dollars*, int *cents*) that add the parameter values to the data members.
- Define a method called **add** (USMoney *money*) that creates and returns a new **USMoney** object representing the sum of the object whose add() is invoked and the parameter. Make sure it does not modify the value of the two existing objects.
- Define a method called **toString()** that returns a string representation of the object.
- Include **Javadoc** comments to identify the author and for identify the parameters and return type for methods, addTo() and add().

Given below is an example of using some of the methods of **USMoney** object.

Usage:

```
USMoney m1 = new USMoney (15,80);
System.out.println (m1);      // Should print $15.80
m1.addTo (25,100);
System.out.println (m1);    // Should print $41.80
USMoney m2 = m1.add( new USMoney (2.90));
System.out.println (m2);    // Should print $44.70
System.out.println (m1);    // Should print $41.80
```

Create a class called **USMoneyTester with the main().** Inside the main, include statements to create instances of USMoney (you must create instances using each of the constructors) and call the methods on each of the instances.

Include the following test case. Make sure that you are calling the methods with values that test the constraints and rules specified for the class.

// Creating instances

```
USMoney amt1 = new USMoney ();

System.out.println (amt1);

amt1.setCents (250);

System.out.println (amt1);

amt1.setDollars (10);

System.out.println (amt1);

System.out.println (amt1.getCents());

USMoney amt2 = amt1.add( new USMoney (2,90));

System.out.println (amt1);

System.out.println (amt2);

amt2.addTo(amt1.getDollars(), amt1.getCents());

System.out.println (amt2);
```

Question 2 (40 pts)

Write a class called **MyUtil** that includes the following 2 **static methods**:

- **String displayArray (int [] array)** that takes an integer array as parameter, uses a loop to create and return a new String that represents the contents of the array surrounded by braces and separated by commas.

  For example,

  ```
  int [] myArray = { 2,90,14,15};
  ```

```
String str = displayArray(myArray);

System.out.println (str);      // should display {2,90,14,15}
```

- **void reverseANumber(int aNumber)** that takes a positive integer parameter and displays it in reverse. For example,

  `reverseANumber(12345)` should display it as 54321.

- The above 2 methods are to be written as static methods. Do you see any reason to define them as instance methods, instead? Give your reasons as to why you think they should be instance (or static method).

General Requirements:

1. Include comments in the program (where applicable) to make your logic clear.
2. Use Java conventions for naming classes, methods and variables.

Point distribution:

Correctness: 90%
Program comments and Naming Conventions: 5%
Output Screen shots, reflections document: 5%

What to submit :

a) A zipped file (assign1_*yourfirstInitialLastName*) with the following:

- A readme document (include your name, course name, assignment number) identifying the IDE and Java version used.
- Source code
- A screenshot of your output.

References:

Java packages

Java Naming Conventions