PROG 1090 – Java **Assignment 1** **Due: 0830 hours 27 Sep 2012**

Topic: Command Line Arguments and Conversion of String to double

Background:

The goal is to write a java program that will add up all the floating-point command line arguments passed to it and display a descriptive output including the sum of the numbers and the average of the numbers. The problem is that the command line arguments are **String** objects and not **double** primitives. Consider the following initial attempt:

```
public class ComdLineCalc {
   public static void main(String args[]) {
      // assume only two numbers are passed but your program must allow n of them
      double answer = args[0] + args[1];  // causes an error
      System.out.println(args[0] + " + " + args[1] + " = " + answer);
   }
}
```

However, when you try to compile this, you will get the following compile error: **Incompatible type for declaration. Can't convert java.lang.String to double** on the line assigning a value to answer.

To solve the problem you need to come up with a way to convert the String objects to numbers. One way to convert a String to a number is to use a method in the **Double** wrapper class called **parseDouble**.

Task:

Write a java class called **ComdLineCalc** which accepts **n** floating-point command line arguments and produces descriptive output which includes their sum and average.

For example if we ran your program at the command prompt as follows:

java ComdLineCalc 3.1 4.9 10.0

The resulting output would be:

3.1 + 4.9 + 10.0 = 18.0
average = 18.0/3 = 6.0

and if no command line arguments are passed then the output would be:

usage: java ComdLineCalc <double> <double> <double> …

and if any of the command line arguments is not a number then the output will indicate the offending argument, like …. **STOPPED - Arg # 3: six is not a number** (assuming the arguments were 4 5 six 7)

When complete, the following must be done to submit this assignment:

1. Attach an e-copy of a memo outlining your experiences with this assignment to Blackboard.

2. Attach an e-copy of your ".java" file to Blackboard so that your instructor can test it.

3. Hand in a printed copy of your source file as well as output screens when the command line arguments are as follows:

   a. Run 1: 3.1 4.9 10.0
   b. Run 2: 1 2 3 4 5 6 7 8 9 10
   c. Run 3:
   d. Run 4: 100.97 22.87 10 q234 97.3