COMP202 Winter 2005 Assignment #1

Distributed: **Jan 10**, Due: **Jan 24** (23:55)

Question 0: Preliminaries (0/20 points)

This question doesn't count for marks but will help you get started. Note the following exception: for this question and **this question only** you are allowed and encouraged to work together (in teams).

1. If you plan to work from home, visit `http://java.sun.com/`, download and install Sun's Java Software Development Kit (SDK) version 1.5 (AKA version 5.0, hereon referred to as 1.5). It's part of the Java 2 Standard Platform Edition (J2SE 5.0). It is important that you use get this version as your book uses new concepts from Java 1.5 without mentioning that they are new to 1.5. If you plan to work from the SOCS labs at school (3rd floor Trottier), you must make Java 1.5 the default compiler (currently by default it's Java 1.4.2). The correct way to do that is by following the instructions on `http://www.cs.mcgill.ca/socsinfo/java/`. If you use the FreeBSD machines, please ask the lab attendant how to make Java 1.5 the default. To check that you have the correct versions, you can run `javac -version` and `java -version`.

2. Familiarize yourself with an editor or IDE. Netbeans is now bundled with the Java SDKi and there's also Eclipse (`http://www.eclipse.org`) which work in most operating systems. In Windows, JCreator (`http://www.jcreator.com`) is nice and simple and jEdit (`http://www.jedit.org`) has lots of nice features. You'll have to download these and install them because Windows doesn't come with them. In Unix and/or Gentoo GNU/Linux, I suggest using `nano`, `pico`, `kate`, or `gedit`. There are also `vim`, `gvim`, `emacs`, `xemacs` but they're all a bit harder to learn how to use. Most of the lab machines should already have these installed. You don't have to install or try all of these options: just find one you like because you'll be using it for every assignment.

3. Open a Web Browser and go to the WebCT (`http://webct.mcgill.ca`) web site. WebCT is the tool you will be using to submit assignments. You login to it by using the same login information that you use on Minerva. Click on COMP202 and browse through the various pages. It has several features like chat, discussion, grade listings, "email" etc. In particular, find out how to submit an assignment.

Question 1: What Goes up Must Come Down (3/20 points)

Thanks to Newton, humanity was ascertained that indeed what goes go up must come down. This is an obvious fact that most people take for granted. What's not so obvious is *how up* do things go? Newton proved that there is a physical relationship between the initial velocity of an object thrown upwards and the height it reaches. This program calculates the height reached by an object thrown straight up, with a given initial velocity.

Retype the following program, compile and run it. Hand in both your Java source file and the Java bytecode (.class) files. Feel free to experiment and modify the value of the initial velocity.

```
// Name: <<put your name here>>
// ID number: <<put your ID number here>>
```

```
// Date: <<put the date here>>
//
// A class that computes the distance traveled by an object
// thrown directly upwards.

public class NewtonianPhysics
{
   public static void main(String[] args)
   {
      // equation:  (v_t)^2 - (v_0)^2 = 2 g delta_y

      // velocity in meters per seconds
      double v_0= 4.2;

      // velocity of the object at the top is 0
      double v_t= 0.0;

      // constant acceleration wrt. gravity (in m/s^2)
      double g= -9.8;

      // now do the calculation!
      double LHS = (v_t*v_t - v_0*v_0);
      double delta_y = LHS/(2*g);

      System.out.println("The object travels " + delta_y + " meters upwards.");
   }
}
```

Question 2: Family Member Age Statistics (7/20 points)

Jenny R. Bentley has recently taken a course on sample statistics and is now enthused by calculating statistics of pretty much anything. Jenny isn't the greatest programmer, and she'd like to automate the process. She used Google to learn Java (instead of attending the COMP202 lectures), and created the following monstrosity. Your job is to fix Jenny's program so that it compiles and works.

There are 7 errors (commonly called "bugs") in the following program. Find them and fix them. Recompile the program, execute it, and make sure it is outputting the right answers. Try to find as many bugs as you can before compiling the program: the quicker you get at recognizing bugs in your code, the easier it will be when you have to compile it.

Note that Jenny told you that she's 100% sure that her Math calculations in the program are correct and that the only trouble is that she's unfamiliar with writing Java programs. Also note that abbreviations and/or missing words in comments are not considered bugs. Bugs are really what prevent the program from compiling and/or working properly.

For this question, you need only hand in the Java source file.

```
// Name: <<put your name here>>
// ID number: <<put your ID number here>>
// Date: <<put the date here>>
//
// A class that computes the sample statistics of the ages of
// family members.

public class FamilyStats
```

```
{
   public static void main(String[] args)
   {
      /*
                     math equations obtained from:                  */
            http://mathworld.wolfram.com/SampleVariance.html


      // define some ages
      int momsAge= 42; dadsAge= 43;
      int myAge= 22, sistersAge= 16;
      int dogsAge= 6;

      // get the mean
      double ageSum = (momsAge + dadsAge + myAge + sistersAge + DogsAge);
      double average = ageSum / 5

      // calculate the sample variance
      double variance= 0.0;
      variance += (momsAge - average)*(momsAge - average);
      variance += (dadsAge - average)(dadsAge - average);
      variance += (myAge - average)*(myAge - average);
      variance += (sistersAge - average)*(sistersAge - average);
      variance += (dogsAge - average)*(dogsAge - average);
      variance = variance / 4;

      // get the std. dev
      double standardDev= Math.sqrt(variance);

      // output the results
      System.out.println(The sample age mean is:  + average);
      System.out.println("The sample age variance is: " + variance);
      System.out.println("The sample age standard deviation is: " + standardDev);
   }
```

Question 3: Conversion Madness (10/20 points)

JoeBloe FromIdaho (JBFI or just "Joe", for short) has recently written a paper for a conference in Europe that got accepted. So, Joe applied for his passport, booked his flight and is almost all ready to go. Luckily, Joe knows a bit of French because he studies in Quebec and so language will not be a problem for him on his trip. However, Joe is absolutely terrible at converting between the English system and the metric system. Joe still weighs himself in pounds, uses the Fahrenheit scale for weather, uses US dollars for currency, measures his height in inches, and counts calories instead of kilojoules.

You know Joe personally, and he's a real swell guy.. so you want to help him out by applying a few things that you learned in your COMP202 class. Joe, though, is a bit scared of computers and he doesn't want to type much. In fact, he wants a program that only reads one single number and does all the conversions on that one number. That makes things easy for poor Joe.

Your program should use the following conversion rates: 1 US dollar = 0.765840 euros; Fahrenheit = 1.8*Celsius + 32; 1 kg = 2.2 pounds; 1 inch = 2.54 centimeters; 1 calorie = 4.1868 joules.

Your task is to build a program that reads a single number and outputs the converted numbers. An

example output of a single run of the program is found below.

```
Please enter a number: 45.7

45.7 Euros = 59.673038754831303 US dollars
45.7 Celsius = 114.26 Fahrenheit
45.7 kilograms = 100.54 pounds
45.7 centimeters = 17.99212598425197 inches
45.7 joules = 10.915257475876565 calories
```

Again, you must submit Java source files and class files.

Marking Scheme

If you do not submit a source file, you automatically get 0 for the question.

- Question 1: 2 points for the Java source, 1 point for the class file
- Question 2 depends on how many bugs you manage to find in the program, each bug being worth 1 point.
- Question 3: 3 points for calculating the values correctly, 4 points if the program is structured nicely (nice comments, good variable names, not hard to understand, etc.), 2 points for nicely formatted output, and 1 point for including the executable class file along with the source file.