**Institute of Technology Tralee**

**Ord/Hons BSc. in Computing with Specialism (Group C) - Year 1**

**Continuous Assessment #2**

**Date: 7/12/12**

**Time: 9 – 11 a.m.**

**Introduction to Programming**

**Instructions:** Attempt the following question. You should use the JCreator IDE for coding. When you are finished you must print out your code for correction.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Q1. (Contains modifications to original question)**

A Java program is required that will present the user with the following menu options:

1. Convert Ounces to Pounds and Ounces
2. Convert Pounds to Kilos
3. Quit

The user can choose any of the options above when the program runs and the main looping process will continue until the user selects option 3, the “Quit” option. You should code the main looping process here using a **do-while** loop.

The choice made by the user (1, 2 or 3) must be validated – although you can assume that a whole number has been entered for the choice. Any value entered outside of these 3 possibilities must be rejected and the user continually given the chance to re-enter their choice.

Should the user select option 1, then they will be prompted for a quantity in ounces. This quantity must be validated so that it is **greater than or equal to zero** – again, you can assume that a number has been entered for the ounces. Should an invalid value be entered, it will be rejected continually until a valid value is supplied. Once a valid value is supplied, the conversion to pounds and ounces should be performed and the result displayed. Use the fact that there are 16 ounces in one pound to help you in your calculations here.

Should the user select option 2, then they will be prompted for a quantity in pounds. This quantity must be validated so that it is **greater than zero**. Should an invalid value be entered, it will be rejected continually until a valid value is supplied. Once a valid value is supplied, the conversion to kilos should be performed and the result displayed to **3 decimal places**. Use the fact that 1 kilo equals 2.2 pounds to do the conversion here.

Should the user select option 3, then the main looping process should quit and the user should then be given a farewell message.

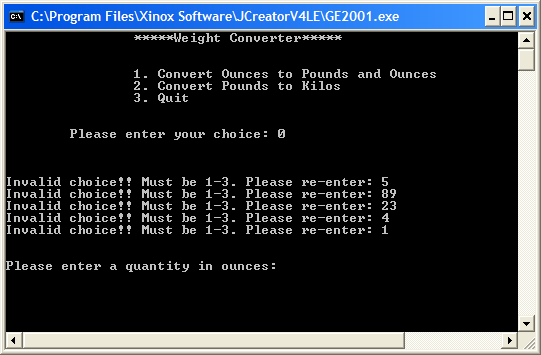
You can use **while** loops here for any validation processes and you should use a **switch** structure when dealing with the users choice (after it has been validated).

Using the test values as indicated in the screen shots below, the program should give you **exactly** the following output when it runs, including banners, blank lines, tabs etc.

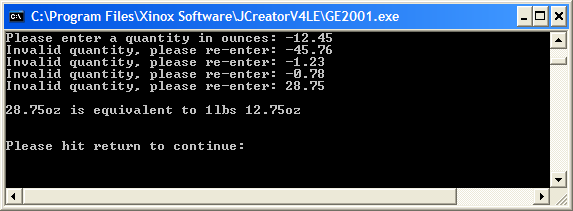
Also note that there will be a few marks awarded for having a **single-line comment** and a **meaningful multi-line comment** at the top of the program.

**Sample Screen Shots**

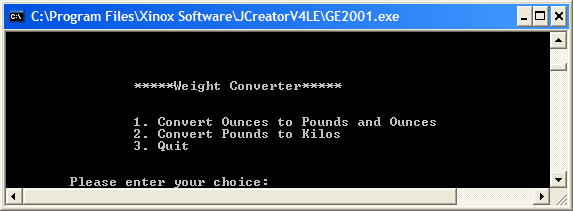
**In this first iteration of the main loop, the menu options appear and the user is prompted for their choice. As long as the choice entered is not one of the values 1, 2 or 3 , the value will be rejected and the user asked continuously to re-enter until a valid choice is supplied. When a valid choice gets supplied (1 in this case), the program then asks the user to supply a quantity of ounces.**



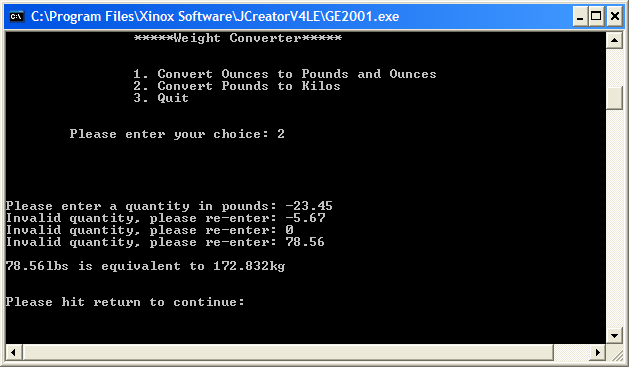
**The program halts waiting for the user to enter a quantity in ounces. This quantity is validated so that the user is continuously asked to re-enter the amount in ounces until a valid positive number (at least zero) has been supplied. In this case a valid value of 28.75 is finally entered and then the conversion to pounds and ounces occurs, with the result displayed. The program then halts until the user hits return to continue on.**



**Once the user hits return the main loop begins once more, displaying the menu-options as before:**



**This time, assuming the user enters 2 as their choice, then the user will be asked to enter a quantity in pounds. This will be validated to ensure it is a valid number greater than zero. The user will be continuously asked to re-enter the value until a valid one is supplied. When a valid value is given, then it will be converted to kilos and the result displayed to 3 decimal places. Again, the program will then halt until the user hits return.**



**Once the user hits return the main loop begins once more, displaying the menu-options as before. This time, assuming the user picks option 3, the program will simply display a farewell message and quit immediately.**

