**Lab 7: Modular Programming**

**Lab 7.1 – Pseudocode**

In this portion of the lab you will analyze the problem listed below. You will figure out what the inputs and outputs are as well as develop a problem statement. You will also determine what variables you will use and create some pseudocode.

**A good friend of yours is creating a web site to help elementary school children understand the metric system. Your friend has asked you for some help with the code. The program will offer the user a menu so that they can convert any unit of measurement into its metric equivalent. The user will select what they want to convert (i.e. feet to meters, inches to centimeters, etc.) and then enter a unit of measurement. Depending upon which conversion they select, the program will perform the proper conversion calculations and display the converted value to the user. The program will then ask the user if they want to continue converting values.**

**Your friend wants to start out simple, so for right now all you need to do is create a simple menu offering the user the chance to convert Fahrenheit to Celsius and feet to meters. Here are the equations you will need to use:**

**Fahrenheit to Celsius**

**(5 / 9) \* Fahrenheit – 32**

**Feet to Meters**

**0.305 \* foot**

**Make sure you solve this problem using modular programming techniques.**

**Step 1**: Analyze the problem. In the table below list the inputs, outputs and a one sentence description of the problem.

|  |  |
| --- | --- |
| Input: | Choice, Fahrenheit, Feet |
| Problem Statement: | Need to make a program that calls on selected functions to convert to metric. |
| Output: | Celsius, Meters |

**Step 2:** What variables are you going to need and what will be their datatype. (The number of spaces in this table does not necessarily mean these are the only variables\constants. This problem can be solved a number of different ways.)

|  |  |
| --- | --- |
| **Variable\Constant Name** | **Datatype** |
| Fahr | Float |
| Celsius | Float |
| Feet | Float |
| Meters | Float |
| Choice | String |

**Step 3:** Create a 30,000 view outline of the problem. How would you break this problem into numerous smaller units? Place your outline below:

|  |
| --- |
| * Make a while loop |
| * Create multiple if statements to look for the selected choice * Call on the appropriate function * Gather the input * Output the results * Continue the loop until “e” is chosen |

**Step 4:** In looking at your overall outline of the problem. Break apart your larger portions into smaller ones. Place your revised outline below:

|  |
| --- |
| * Make a while loop   + Look for cont to equal “no”   + Ask for a selection to get choice * Create multiple if statements to look for the selected choice   + If choice is C     - Ask for degrees fahrenheit     - Call on the conversion function   + If choice is F     - Ask for feet     - Call on the conversion function   + If choice is E     - Print “thanks for using the program”     - Set cont = “no” to exit the loop   + Else     - Print an error and have them try again * Output the results * Continue the loop until “e” is chosen |

**Lab 7.2 – Raptor**

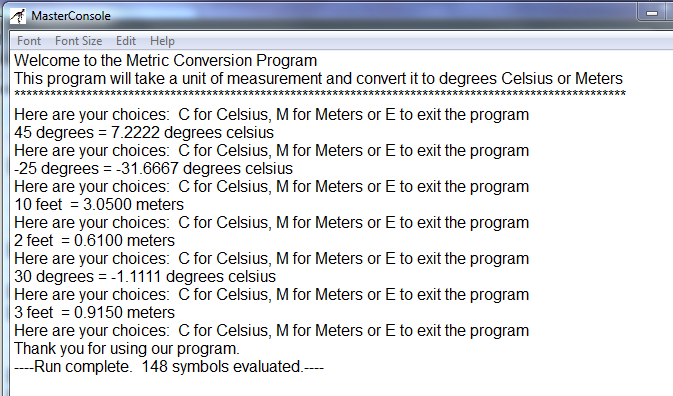
In this portion of the lab you will take the pseudocode you developed above and test it using Raptor.

**Step 1:** Code in Raptor. Open up Raptor and code this problem. Remember in Raptor the index starts at 1

**Step 2**: Test your program using the following data. At this point don’t worry about getting your numbers to display only 2 decimal places. We will learn how to do that later.

|  |  |  |  |
| --- | --- | --- | --- |
| Fahrenheit | Celsius | Feet | Meters |
| 45 | 7.2222 | 10 | 3.05 |
| -25 | -31.66667 | 2 | 0.6100 |
| 30 | -1.1111 | 3 | 0.9150 |

My output looked like the following:



**Step 3**: If your results match the test, then hand in your Raptor program and this lab. Otherwise go back over your Raptor program and correct any errors.

# Lab 7.3 – Translating into Python

The purpose of this portion of the lab is to take the problem you analyzed and coded in Raptor in part

7.2 and translate your Raptor file to Python code

**Step 1:** Open up your Raptor file and resize the window so that it takes up half of your computer screen.

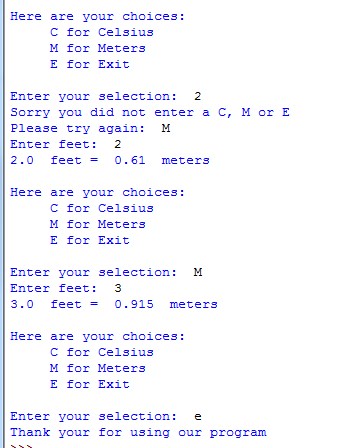
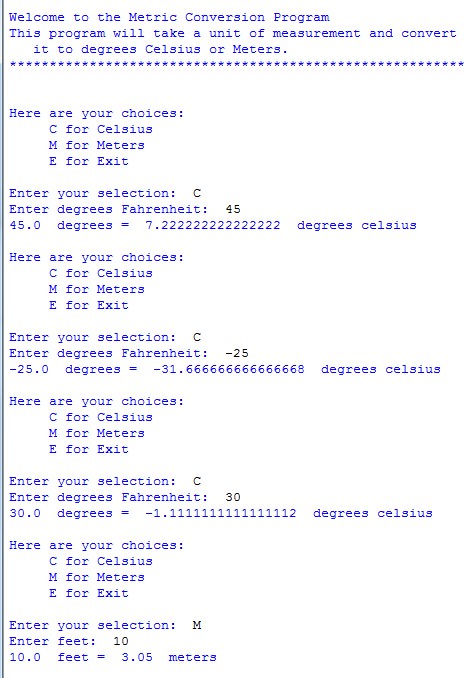
**Step 2:** Open IDLE (Python compiler). Click on File and select New Window. In the new window click on Save. Give your file a name. Resize it so that it takes up half of your screen just like you did previously in this lab.

**Step 3:** Translate your Raptor file into Python code.

**Step 4:** Test and Debug your Python code using the same test data you used for the Raptor file. The table below lists the same data. Since we know how to format numbers in Python, make sure you output is formatted to 2 decimal places.

|  |  |  |  |
| --- | --- | --- | --- |
| **Fahrenheit** | **Celsius** | **Feet** | **Meters** |
| 45 | 7.2222 | 10 | 3.05 |
| -25 | -31.66667 | 2 | 0.6100 |
| 30 | -1.1111 | 3 | 0.9150 |

My output looked like the following:



**Step 7**: If your results match the test, then you are done. Otherwise go back over your code and correct any errors.

**How Will I Be Graded**

You will be graded based off of the following rubric

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Exemplary** | **80**  **%** | **Competent** | **60**  **%** | **Developing** |
| 4 | Completed the analysis of inputs and outputs, developed a correct problem statement and developed thorough pseudocode. | 3.2 | Did not complete the input and output analysis or the problem statement was not detailed enough. | 2.4 | Did not develop the analysis of the inputs and outputs and problem statement or the pseudocode was lacking detail. |
| 6 | Correctly coded the problem in Raptor. Includes comments, variable declarations, input statements, process statements and correct output | 4.8 | Coded problem in Raptor but missing comments, an input statement or an output statement. | 3.6 | Completed the problem in Raptor but there was an error in the processing |
| 6 | Proper creation of functions to handle conversions, to display menu and handle other program operations. | 4.8 | Minor error in the creation of one of the functions. | 3.6 | Major error in one of the functions or code did not use enough functions |
| 4 | Main program procedures  works correctly. Very little code in main. All code in the procedures. | 3.2 | There is a minor error in program execution. | 2.4 | Major error in program execution. |
| 20 | TOTAL POINTS | 16 |  | 12 |  |

**Handing it all in**

1. Log on to Canvas and our course shell.
2. Modules\Lab#7
3. Upload this worksheet showing your 30,000 foot view and pseudo code along with your Raptor and Python files to the Lab #7 drop box.