**CMSC 140 Programming Project 2**

**Chapter Covered:**

* Chapters 1- 4

**Concepts tested in this project**

* [ALL concepts from previous projects]
* Selection control structure
  + The if statement
  + The switch statement (optional)
* Relational operators and logical operators
  + Use of relational and logical operators to check numeric ranges
* User friendly
  + Use of user friendly user prompt and simple input validation
  + Use of meaningful output labels and format
  + Use of output manipulators
* Working with string type
* Use of getline()

**Project Description**

There are two main systems for measuring distance, weight and temperature, the Imperial System of Measurement and the Metric System of Measurement. Most countries use the Metric System, which uses the measuring units such as meters and grams and adds prefixes like kilo, milli and centi to count orders of magnitude. In the United States, we use the older Imperial system, where things are measured in feet, inches and pounds.

Write a program that shows the following menu options and lets the user to convert from Metric to Imperial system:

Converter Toolkit

--------------------

1. Temperature Converter

2. Distance Converter

3. Weight Converter

4. Quit

* If the user enters 1, the program should ask for the temperature in Celsius and convert it to Fahrenheit
* If the user enters 2, the program should ask for the distance in Kilometer and convert it to Mile
* If the user enters 3, the program should ask for the weight in Kilogram and convert it to Pound
* If the user enters 4, the program should end.

**Project Specifications**

Input for this project:

* the user must enter an number to select a menu option
* the user must enter temperature in Celsius
* the user must enter distance in Kilometer
* the user must enter weight in Kilogram
* the user must enter a country name

Input Validation:

* Do not accept a number outside the range of 1 through 4 for the menu option. Be sure to display appropriate error message if the input is invalid.
* Do not accept negative numbers for distance and weight. Be sure to display appropriate error message if the input is invalid.

Output: The program should display the following:

* a menu for Converter Toolkit
* temperature in Fahrenheit, distance in miles or weight in pounds
* a country name
* Programmer’s full name
* project number
* a due date

**Processing Requirements**

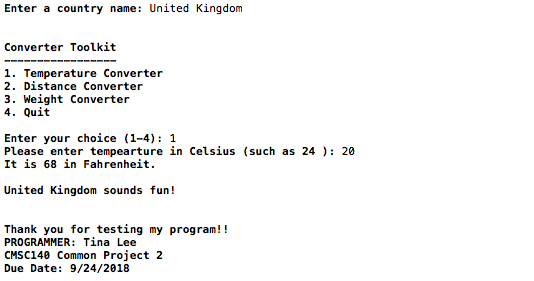
1. The program should use at least one selection control structure (if – else statement)
2. Be sure to convert as specified. For example, convert temperature from Celsius to Fahrenheit, not the other way around.
3. Use the following for converting input:

* 1 kilometer = 0.6 mile,
* 1 kilogram = 2.2 pounds,
* The formula for converting Celsius degree to Fahrenheit is:

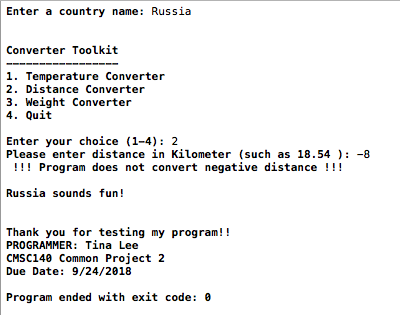
***F = (9/5)\*C + 32*** where F is the temperature in Fahrenheit and C is the temperature in Celsius

1. Convert temperature to a whole number such as 78, distance to two positions after decimal point (for example 84.56) and weight to one position after decimal point (For example 121.6).
2. Use C++ constants to store all permanent data.
3. Don’t use numbers of literals, such as 9, 5, 32. Instead use constants.
4. Your program should do all calculations using C++ expressions (Don’t use calculators for any calculations)

**Sample Screen Output #1:**



**Sample Screen Output #2:**



**Special Project 2 Submission Requirements**

**Deliverables:**

* ***Intermediate deliverable:***

Program design- Flowchart and/or pseudo code for the Program due one week after project is given. An intermediate assignment will be created for submission. Flowchart/pseudo code can be submitted electronically in the following format: word document, pdf or handwritten flowchart/pseudo code picture saved as .jpg or png.

* ***Project 2 deliverables*** (due one week after Project Design submitted).

**CMSC140 Project 2 Submission Requirements**

**Important NOTE:**

**-If your program does not compile, project will get a grade of “0”.** Contact your instructor prior to the project submission due date if you have compilation issues.

-If your project has compilation errors and you are not able to resolve them, try to see your instructor during his/her office hour or go to tutoring session for help **prior to project’s due date**.

**Details for Deliverables:**

1. ***Word document(docx) or .pdf*** with a name FirstInitialLastName<number>.docx or .pdf should include:
   1. Title page including student name, project title and number
   2. Completed Test Plan table (given in the project’s description document)
   3. Screenshot of the program run (using test data from the Test Plan)
   4. Pseudocode
   5. Student Checklist
   6. Lessons Learned: Provide answers to the questions listed below:
      * + - Write about your Learning Experience, highlighting your lessons learned and learning experience from working on this project.
          - What have you learned?
          - What did you struggle with?
          - What would you do differently on your next project?
          - What parts of this assignment were you successful with, and what parts (if any) were you not successful with?
          - Provide any additional resources/links/videos you used to while working on this assignment/project.
2. ***Compressed file containing your C++ file*** with a name FirstInitialLastName\_Pr2\_Moss.zip (This folder should contain C++ source file FirstInitialLastName\_Pr1.cpp only).

Include the following in your code in C++ source file FirstInitialLastName\_Pr2.cpp:

Program Header: All programming projects’ source code need to have one block comment at the top of the program containing the course name and CRN, the project number, your name, project description, and the due date.

Following is a template of the required program header:

/\*

\* Class: CMSC140 CRN 20433

\* Instructor: Grigoriy A. Grinberg

\* Project 2

\* Description: The department planning to purchase a humanoid robot.

\* The Chairman would like us to we are creating a program

\* to show a greeting script the robot can use later.

\* Due Date: 10/04/21

\* I pledge that I have completed the programming assignment independently.

I have not copied the code from a student or any source.

I have not given my code to any student.

Print your Full Name here:

\*/

Comments: Provide any additional comments as necessary to clarify the program. Add comment to variables, formulas, or any part of the program with the purpose of making the source code easier to understand.

Indentation: It must be consistent throughout the program and must reflect the control structure.

Proper naming conventions: Variable and method names need to be descriptive to show the role of the variable or method. Avoid single letter names. Constant names should be all upper case, variable names should use “camel case” (i.e. start with lower case, with subsequent words starting with upper case: *hours****W****orked* for example) or underscores to separate words (i.e. items\_ordered).

**Notes:**

***Test Plan***

Test your program with at least two more test cases. Use the given data as an example. Record your data for input and output in the following table. **Make sure your tests cover all the possible scenarios.**

**A good test plan should be comprehensive**. This means you should have a few test cases that test when the input is out of range, division by 0, etc. For example:

Input validation requirements: Display an error message if the user enters for example: a number outside the range of 1 through 4 when selecting an item from the menu. No negative values accepted for the circle’s radius. You should add at least three cases:

Case 1: Check program if user enters a choice out of range: 0

Case 2: Check program if user enters a choice out of range: 5

Case 3: Check program if user enters a negative distance: -2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case #** | **Input** | **Actual Input** | **Expected Output** | **Actual Output** | **Did the test pass?** |
| 1 | Choice 1  Temp 20 |  | 68 |  |  |
| 2 | Choice 2  Distance -8 |  | Error message |  |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |

***Check List***

|  |  |  |  |
| --- | --- | --- | --- |
| **Nr** |  | **Y/N** | **Comments** |
| **1** | **Compressed Source Code .cpp file:** FirstInitialLastNam\_Pr2.zip | **Y** |  |
| **2** | **Documentation File:** FirstInitialLastNam\_Pr2.docx |  |  |
| **3** | **Program compiles** | **Y** |  |
| **4** | **Program runs with desired outputs related to a Test Plan** | **Y** |  |
| **5** | **Documentation file:** | **Y** |  |
|  | Title page | **Y** |  |
|  | Comprehensive Test Plan | **Y** |  |
|  | Screenshots based on Test Plan | **Y** |  |
|  | Algorithm (Pseudocode) | **Y** |  |
|  | Lessons Learned | **Y** |  |
|  | Student Check List | **Y** |  |

|  |  |
| --- | --- |
| **C140 Grading Rubric** | **Possible points deduct** |
| **TESTING** |  |
| Project must compile. If it doesn't compile | -100 |
| Project must run. If it's run- time error | -100 |
| Follows assignment document instructions | -25 |
| Doesn’t pass private instructor test | -75 |
| **Requirements** |  |
| Using integer or floating literals instead of constants or variables (for each literal -10 points) | -10 |
| Using calculator to calculate any expressions instead of using C++ calculation expressions (– 20 points for each missing expression) | -20 |
| **Documentation:** |  |
| Header comments at the top of the program are missing | -10 |
| Additional comments should be provided to clarify a code | -10 |
| Design: Pseudocode/algorithm missing or incorrect | -10 |
| Required output screenshots are missing | -10 |
| Lessons Learned are missing or incomplete | -10 |
| Student Checklist is missing or is not completed | -10 |
| FirstInitialLastName\_Pr1\_Moss.zip is missing | -100 |
| FirstInitialLastName\_Pr1.docx is missing | -50 |
| **Programming Style:** |  |
| Incorrect use of indentation, naming convention, etc: see coding/style standards | -10 |
| Output is not easy to understand | -5 |
| **Design:** |  |
| Does not print application header | -5 |
| Does not print the Programmer's name at the end | -5 |