Fall 2020
Computer Science I — Maíra Marques Samary PhD
Section 5 – Tuesday and Thursdays – 12:00PM – Fulton 250

TA's Office Hours – online

James Monahan - monahajm@bc.edu - https://bccte.zoom.us/j/2822792652
Tuesdays 7:00 PM – 8:00 PM
Wednesdays 4:00 PM – 5:00 PM

Jennifer Joseph - - josephjz@bc.edu - https://bccte.zoom.us/j/5882755193
Wednesdays 11:00 AM – 12:00 PM
Thursdays 3:00 PM - 4:00 PM

Liam Murphy- - murpaue@bc.edu - https://bccte.zoom.us/j/3085424208
Tuesdays 2PM-4PM

Discussion Groups:

CSCI100701 - Tuesday 6:00 PM – 6:50 PM- Fulton Hall 220 (James Monahan)
CSCI100702 - Thursday 5:00 PM – 5:50 PM – Fulton Hall 220 (Jennifer Joseph)
CSCI100703 - Wednesday 4:00 PM – 4:50 PM – Gasson Hall 203 (Liam Murphy)

Homework 2

Due date – 09/29/20 11:59 PM

General Instructions

Create a folder named ***LASTNAME_FIRSTNAME***. You will populate the folder with **ALL** of the .py files you write for this homework. To submit the homework, verify the folder includes all your .py files, compress (zip) the folder then upload to Canvas. Remember to include the following comments at the **top of each** of your .py files:

```
# author:
# assignment:
# description:
```

What to submit in Canvas?

Make sure all your files are saved in the folder LASTNAME_FIRSTNAME, then compress (zip) the folder and upload to Canvas.

If you encounter any problems in completing the assignment or in the submission process, please don't hesitate to ask for help. The sooner, the better!

Problem

1.

Create a library/module (named **temperature.py**) that will serve as a function library (similar to the way the math/random library/module serves as a library for functions that perform mathematical/random operations). This library/module should contain the following three functions:

- o A function that accepts a temperature value in degrees Centigrade and returns the equivalent temperature value in degrees Fahrenheit.
- o A function that accepts a temperature value in degrees Fahrenheit and returns the equivalent temperature value in degrees Centigrade.
- o A function that accepts a temperature value in degrees Fahrenheit and a wind speed value in miles per hour, and returns the calculated wind chill factor for those parameters.

Formulas:

To convert between Fahrenheit (^{o}F) and degrees Celsius (^{o}C):

$$T_c = \frac{5}{9} \times (T_f - 32)$$

$$T_f = \left(\frac{9}{5}\right) \times T_c + 32$$

Where: T_c is temperature in Celsius

T_f is temperature in Fahrenheit

Wind chill = $91.4 - ((91.4 - T) \times \left(0.478 + \left(0.301 \times \sqrt[2]{W}\right) - (0.02 \times W)\right))$

Where: W (wind speed) is in mph

T (Temperature) is in ^{o}F

2.

Create a library/module (named **test_temp.py**) that will serve as a test program for the temperature module. This module will contain the following features:

o A function that prints a menu to standard output that provides the user with four options (convert a Centigrade temperature to a Fahrenheit temperature, convert a Fahrenheit temperature to a Centigrade temperature, calculate a wind chill factor, and exit the application).

o A main function that presents the user menu, requests appropriate data from the user according to the option selected, perform the selected operation, and display the result to the user.

o This module may also optionally include additional functions that perform each of the individual processes associated with each user selection option.

Tips:

- Code that resides in external libraries/modules must generally be imported in order to be used.

- The user interface should include appropriate prompts to clearly direct the user regarding system operation and should clearly present meaningful program results.

- Function headers (comments) should describe the operation of the function, the data required to be submitted as formal parameters, and the data to be returned

RUBRIC

	Excellent (100% of points)	Average (60% of points)	Needs Improving (40% of points)	Possible Points
Submission Details	• Both file name and file header meet stated spec	• Either file name is incorrect or file header is missing sections/details	• Both file name and file header is missing or are incorrectly implemented	**1**
Function module	• Functions are implemented properly • Function headers are complete and accurate • Values are validated (ex. when a number is expected, a string is not accepted)	• Functions are generally implemented properly, but exhibit minor errors. • Function headers are generally complete and accurate, but some minor details are missing.	• Functions are implemented improperly. • Function headers are sketchy or missing.	**3**
Test (driver) module	• Functions are implemented properly • Function headers are complete and accurate • Application operates exactly per spec.	• Functions are generally implemented properly, but exhibit minor errors. • Function headers are generally complete and accurate, but some minor details are missing. • System exhibits minor inconsistencies with spec.	• Functions are implemented improperly. • Function headers are sketchy or missing. • System does not operate per spec.	**4**
User interface	• UI formatting is appropriate. • Prompts are complete and concise. • Information is presented in a meaningful form.	• UI formatting exhibits minor flaws. • Prompts are not completely clear and concise. • Information presentation is slightly confusing.	• UI formatting is sketchy or haphazard. • Prompts are confusing or missing completely. • Information presentation is completely confusing.	**2**
FINAL SCORE				**10**