Lab 4: Temp

Introduction

Lab 4 is your first Python script and it is a simple one. It is your first interactive script, meaning your script will prompt the user to enter a temperature in degrees Fahrenheit or Celsius and calculate the other, in a continuous loop until the user exits the program.

Requirements

Your script shall be named `temp` and be marked executable. Here is how you execute your script:

```
./temp
```

Your script shall interact with the user. The first thing it should print is exactly this:

```
Welcome to the CS 3030 Temperature Conversion Program
```

This welcome message should be printed only once.

Your program should then loop until the user enters `3`, followed by `ENTER` to exit the program. Your script should prompt the user with this main menu (note the spacing, colons and number of commas):

```bash
    Main Menu

    1:Fahrenheit to Celsius
    2:Celsius to Fahrenheit
    3:Exit program

    Please enter 1, 2 or 3:
```

The user then enters `1`, `2` or `3` (and hits `ENTER`)

If the user enter `3`, the program exits.

If the user enters a `1`, the user is prompted for degrees Fahrenheit with this prompt:

```
Please enter degrees Fahrenheit:
```

The user enters a number, which should not contain commas but can contain a decimal point or a minus sign. Assume the user enters `56.2`. The program should respond with:

```
56.2 degrees Fahrenheit equals 13.4 degrees Celsius
```

The program then reprints the main menu and prompts the user again, in an endless loop.

If the user enters a `2`, the user should be prompted for degrees Celsius with this prompt:

```
Please enter degrees Celsius:
```

The user enters a number as before, which should not contain commas but can contain a decimal point or a minus sign. Assume the user enters `91.8`. The program should respond with:

```
91.8 degrees Celsius equals 197.2 degrees Fahrenheit
```

If at any time the user enters anything non-numeric (including typing nothing and hitting just `ENTER`), your script should trap the error and at a minimum, print a message that includes:

```
Invalid entry
```

You should then reprint the main menu and start over at the beginning. Do not re-prompt for the degrees, start over at the top. Do not reprint the Welcome message.

Your script should include these two functions to calculate its results (yes, you should use floating point numbers):

`def fahrenheitToCelsius(fahrenheit)`

converts degrees fahrenheit to degrees celsius using this formula: `C = (F - 32.0) * (5.0/9.0)`

`def celsiusToFahrenheit(celsius)`

converts degrees celsius to degrees fahrenheit using this formula: `F = (9.0/5.0) * C + 32.0`

Hints

- Use `try/except` to trap errors when prompting the user or converting the incoming string to a number
- Always restart from the main menu when the user enters anything invalid
- Always restart from the main menu after calculating degrees
- Your script should expect the user to hit `ENTER` after each entry
- Exit immediately when the user enters `3` from the main menu
- Use whitespace and indenting to make your script readable
- Add comments to your script to document your logic
- Run cucumber to check your progress

- Because temp is interactive, the first thing you should implement and test is typing `3` to exit. The cucumber scripts rely on this feature and it **must** work.

Clone your private repo on github.com

In the assignment module in Canvas, find the link to the Canvas page entitled "Clone your Lab 4 Repo", click on it and follow the instructions.

Copy your private repo down to icarus

```bash
git clone https://github.com/cowancs3030<SEMESTER>/lab4-YOURGITHUBUSERNAME

cd lab4-YOURGITHUBUSERNAME
```

Write and test `temp`

Fire up your favorite text editor, and begin with your header:

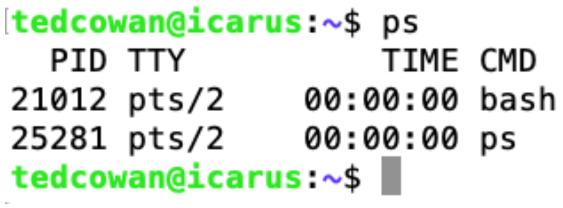
```text
#!/usr/bin/python
# (Your name)
# Lab 4 - Temp
# CS 3030 - Scripting Languages

(Add your amazing, perfect code here)
```

If your `temp` script hangs when you run it, and you have to close and reopen your terminal session, it is possible that your script will continue to run in the background on icarus. It will then slowly consuming all available disk space. When this happens, icarus crashes and the Lab assistants in Ogden will know it is you and hunt you down. If at any time you suspect this might be the case, execute the `ps` command to display all of your running processes:

```
ps
```

You should get something that looks like this:



```
tedcowan@icarus:~$ ps
  PID TTY          TIME CMD
21012 pts/2    00:00:00 bash
25281 pts/2    00:00:00 ps
tedcowan@icarus:~$
```

The 5 digit numbers on the left are process IDs (pids). If you see any other processes than the ones in the above example, issue the `kill` command to terminate each process, substituting the pid number for PID (don't kill the `bash` process or you will be logged out).

```
kill PID
```

The cucumber files for this lab implements a random number generator for the °C and °F values. To ensure your program always produces the correct output, test your program many, many times with various input. Then run cucumber many, many times.

```
./cucumber -s
```

Submit your assignment code for grading

Remember, you must push your script in your private repo to github.com to receive any points, for all assignments in this course.

```bash
git add temp
git commit -m"COMMIT MESSAGE"
git push origin master
```

Files created for this lab

`temp`

Grading

Here is how you earn points for this assignment:

FEATURES	POINTS
Must-Have Features	
Script is named correctly and found in its proper place in your repo	5
Script is executable	5
Required Features	
Script prints “Welcome to the CS 3030 Temperature Conversion Program”	5
Script prints the Main Menu	5
Script correctly converts degrees Fahrenheit to degrees Celsius with 1 decimal place with option 1	10
Script correctly converts degrees Celsius to degrees Fahrenheit with 1 decimal place with option 2	10
Script exits with option 3	10
Script uses function fahrenheitToCelsius()	10
Script uses function celsiusToFahrenheit()	10
Script recovers from non-numeric entry at main menu	10
Script recovers from non-numeric entry when prompting for degrees Fahrenheit	10
Script recovers from non-numeric entry when prompting for degrees Celsius	10
Grand Total	100