CSE 231 Summer 2024

Programming Project #1

Assignment Overview
This assignment involves coding and testing of a program that uses Python arithmetic.

The basic design of your first program in this class: prompts for input, echo the input, performs arithmetic on that information and then displays the results.

Important Note about submission

This assignment is worth 10 points (1.0% of course grade), and must be completed before **11:59 PM EST on Monday, May 20th**. Note: Codio (where you submit your project) uses your computer's time zone to display time so if your computer is set to a different time zone, the wrong due time will be displayed.
After the due date, your score will be deducted by 10% for every 5 hours late or a fraction of it. No submissions will be accepted after 24 hours from the due date. The penalty will be applied on the full project score.
The project will be automatically submitted by the due date (05/20). If you want to keep working on it after the deadline with penalty, you must manually click on "Mark as uncompleted", acknowledge the penalty message and manually click on "Mark as completed" when done.

Deliverables
The deliverable for this assignment is the following file:

`proj01.py` -- your source code solution

Be sure to use the specified file name and to submit it for grading via Codio before the project deadline. You can also just copy and paste your code into the coding window in Codio.

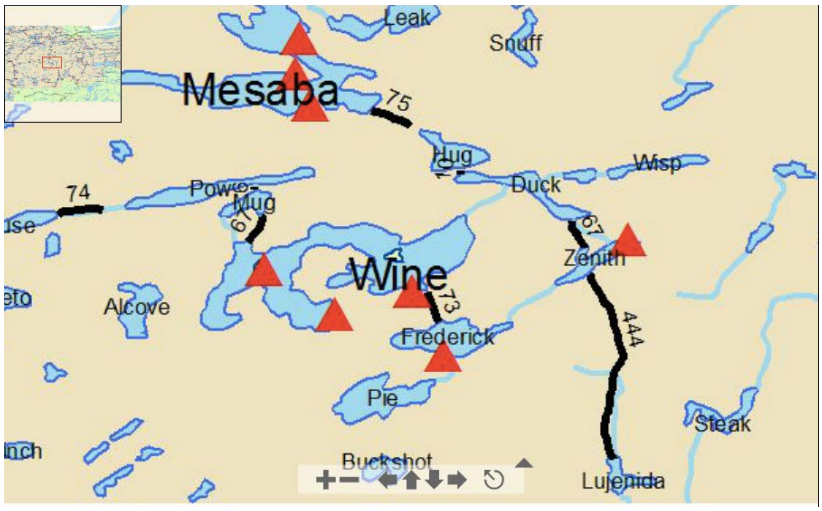
We provide a zip file in the course website that you need to download into your computer, unzip the file, and open it in PyCharm (similar to the lab) to access the starter code.

Background
This programming project will use the `input` and `print` functions along with some simple mathematics for conversion. The important part of the project is to learn the skills needed to access the class web site to download a project description, create a new program in Python and finally to hand it in.

Program Specifications
Conversions are useful both in science and daily life. Here we examine some obscure, but useful conversions as well as some silly ones.

If you canoe in the Boundary Waters Canoe Area (BWCA) you portage (carry) your canoe and gear between lakes as you wander deeper into the wilderness. BWCA maps label distances on portages in rods, an old unit of measurement that is 5.0292 meters, which is approximately the length of a canoe so it is a useful measurement in the wilderness.

Here is a piece of a BWCA map showing portages between lakes (black lines) with the length of the portage labeled in **rods.** (Red triangles are campsites.)



Your program will prompt the user for a floating-point value representing a distance in rods. You will reprint that value along with that value converted to the following values. The most important value for planning a trip is the time to walk the portage.

- meters
- feet
- miles
- furlongs
- the time in minutes to walk that distance.

You can find these measures on the web, but so everyone is using the same conversions, we require that you use the following so testing will yield the same results:

- 1 rod = 5.0292 meters
- 1 furlong = 40 rods
- 1 mile = 1609.34 meters
- 1 foot = 0.3048 meters
- average walking speed is 3.1 miles per hour

Assignment Notes:

1. To clarify the project specifications, sample output is appended to the end of this document.
2. To receive credit your program must take in input, echo the input, do some simple arithmetic based on that input, and then print results. The only input statement should use the following string:
   `"Input rods: "`
3. Round all floating-point output to three places, i.e. use `round( x, 3 ).` Important: do not round intermediate results because rounding differences will compound resulting in a wrong result. Round values only when you print them.
4. Testing on Codio does exact matching of output, so we provide a file `strings.txt` of strings for you to copy to make it easier to get the matching correct. The same strings are provided in the starter code also.

5. Items 1-6 of the Coding Standard will be enforced for this project.
6. The `input` function is used to accept a response from the user. The function accepts a string (a sequence of characters between quotes) as a prompt to display to the user. It then waits until the user types a response (terminated by the user touching the Enter key). Finally, the function returns the user's response as a string.

   If the user's response is supposed to be processed as a numeric value, the returned string must be converted into a number. When working with floating point values, a string is converted into a floating-point number using the `float` function. The function accepts a string as its argument and returns the floating-point number which the string represents. A typical interaction would be something like:

```
num_str = input( "Please enter a number: " )
num_float = float( num_str )
```

7. The `print` function is used to display any combination of variables, values, and strings in the output window. Each item to be displayed must be separated from another item by a comma. All the items will be displayed together, followed by a new line. For example:

```
print( num_int, "times two is", num_int*2 )
```

   Three items will be displayed when the `print` function is called: the value of the variable `num_int,` the string `"times two is",` and the result of the calculation (note that a space was automatically added between the items.

   Assuming that the value of the variable `num_int` is 26, then the output will be:

```
26 times two is 52
```

Getting Started

- *Solve the problem using pencil and paper first.* You cannot write a program until you have figured out how to solve the problem. This first step can be done collaboratively with another student. However, once the discussion turns to Python specifics and the subsequent writing of Python, you must work on your own.

- Use PyCharm to create a new program. Use the required file name (`proj01.py`).

- Write a simple version of the program, e.g. input the rods and print rods. Run the program and track down any errors.

- Use the **Codio** system to turn in the first version of your program.

- Cycle through the steps to incrementally develop your program:

  - Edit your program to add new capabilities.
  - Run the program and fix any errors.

- Use the **Codio** system to submit your final version.

Scoring Rubric

If you correctly run the four sample tests, you will automatically get 8 of the 10 points (details about the tests are in the sample interaction below). The other 2 points will be for the Coding standards. Here is the scoring rubric for this assignment.

```
Computer Project #1 Scoring Summary


General Requirements

  __0__   (2 pts)  Coding standard

                   Source code format
                   Source code header and descriptive comments
                   Mnemonic identifiers (variables and symbolic
constants)

Program Implementation

  __0__   (8 pts) Test cases including the blind test


Note that if you simply “hard code” the results you will earn
zero points for the project, e.g. if your code simply prints
values without any calculations, you are not really doing the
assignment.
```

Sample Interaction

Test 1

```
:~Input rods ~:1

Your value is 1.0 rods.

Conversions
Meters: 5.029
Feet: 16.5
Miles: 0.003
Furlongs: 0.025
Minutes to walk 1.0 rods: 0.06
```

Test 2

```
:~Input rods ~:10

Your value is 10.0 rods.

Conversions
Meters: 50.292
Feet: 165.0
Miles: 0.031
Furlongs: 0.25
Minutes to walk 10.0 rods: 0.605
```

Test 3

```
:~Input rods ~:444.44

Your value is 444.44 rods.

Conversions
Meters: 2235.178
Feet: 7333.26
Miles: 1.389
Furlongs: 11.111
Minutes to walk 444.44 rods: 26.882
```

Test 4

Blind test: you do not see results. If you fail this test make sure that you only round at the end and that you only used the conversion factors that we provided.