PHY1112: Assignment 1

> Documentation and [Please Send] Help

Assigned: January 9th, 2024

Due: January 16th, 2024

Learning Objectives

1. Learn about good documentation practices.
2. Learn how to use the `print` function
3. Learn to accept user input to your programs
4. Learn to use optional (keyword) arguments to functions

Grade Breakdown

Part	1	2	3	Total
Points	4	4	2	10
Score				

Question 1: File Headers and Documentation

File headers are comments are found at the top of a computer script, and are used in any language. They are extremely useful to have from a documentation standpoint, and relay very simple information:

- The *Filename* (a good check to see it is properly up to date)
- The *Author*
- The *Date* the file was created
- The *Date* the file was last modified
- A *Description* of what the program is meant to do.

This is important information, especially when you set aside a script and don't look at it for an extended period of time. It also gives you opportunity to informally cite any resources used, for example:

> "This program uses refractive index data for gold (Johnson and Christy 1972) to determine…"

You will be expected to include a file header for every script you hand in for your assignments. This will not apply to labs as they are time-limited.

It is best to use the "triple quote" form of python comment to create the file header, for example:

```
'''
Filename:           FitData.ph
Author:             Jane Doe (jdoe123@uottawa.ca)
Date Created:       2023-10-03
Date Modified:      2023-10-04
Description:        Peforms a linear regression fit to experimental data
'''
```

An alternate method of commenting is to use the "#" character, as any text following the "#" is treated as a comment and not parsed by the python interpreter.

Good documentation practices use both *regular* comments, and *inline* comments:

- A regular comment typically describes what a series of lines of code that follows will be doing.
- An inline comment is typically used to describe what a variable is. For physics codes, it is also useful for specifying the units being used.

If you select a descriptive name for your variable, an inline comment to describe the variable is often unnecessary, for example:

```
c = 2.998e8        # speed of light (m/s)
```

versus

```
speed_of_light = 2.998e8    # m/s
```

The second method reduces the amount of documentation that is required, and makes the code easier to understand, so is generally preferred. If using the first method, you would have to "remember" what "c" is when you are reading code that is farther down from this comment, and it could be confusing.

This variable naming approach lies within the philosophy referred to as **self-documenting code** since the code itself provides the necessary information without need for additional information through commenting.

For easy readability, inline comments should be preceded by a minimum of two spaces, and all comments should have a single space after the "#" symbol (as illustrated above).

Though it is not required, we recommend the naming convention based on the official PEP8 standard, https://peps.python.org/pep-0008/, which is set by the python creators. For variable naming, any letters used are lower-case, with underscores to separate words, for example, "speed_of_light".

For this question, please do the following:

a) Include appropriate file headers in questions 2 and 3 of this assignment.
(2 marks)

b) Make use of both a regular and inline comments at least once in questions 2 and 3 of the assignment.
(2 marks)

Question 2: Please send `help()`

For this question, we will learn about the `help` function. While not used *in* the code, it is useful overall to gain information regarding a function you would like to use.

When you want to use a function you may not be familiar with, the `help` function will display what the function is for, how it is used, and a summary of the available *arguments* (*i.e.*, what goes inside the brackets of a function call).

a) Recalling your first lab, open a Python terminal. In the terminal, use the `help` function to obtain information about the `print` function, that is type:

```
$ help(print)
```

Take a screenshot of the output and include this in your submitted homework.
(1 mark)

b) In your web browser, find the official documentation for the `print` function on the python website. Include a screenshot of this in your assignment and compare the two.
(1 mark)

c) Now you are introduced to your first *keyword argument*. These are arguments that are optional, and often might have a default.

In the case of the `print` function describe in your own words what the `sep` argument is used for and what the default value is.
(1 mark)

d) Using the `sep` argument, write a "Hello World" program that separates the words using a newline (done using a "\n") and a tab (done using a "\t").
(1 mark)

Question 3: Give me your `input()`

For this question, we will learn about the `input` function. This function is used to obtain input from the user while a script is running. When your script reaches this function, everything pauses while it waits to receive input from the user.

When the `input` function is called, a specified prompt is displayed in the terminal (this is the *standard output* location or *stdout*). The user can then enter their input and hit the enter/return key to finalize.

a) Using the `help` function, or any other resource you see fit, determine how to use the `input` function. Describe how it is used and what arguments it takes.
(1 mark)

b) Using the `input` function and the `print` function, write a script that takes in a user message, and displays it to the terminal.
(1 mark)