CSE 231 Spring 2016

# Programming Project #1

## Assignment Overview

This assignment involves coding and testing of a program using Python arithmetic from Chapter 1 based on the “Hello World” program from the first lab (pre-lab).

This assignment is worth 15 points (1.5% of course grade), and must be completed before 11:59 PM on Monday, January 18th.

## Assignment Deliverable

The deliverable for this assignment is the following file:

`proj01.py` – the source code for your Python program

Be sure to use the specified file name and to submit it for grading via the **handin** system before the project deadline.

## Background

The national debt is always an interesting topic as we enter a presidential election. Just how big is the debt? It is a very big number, and its size makes it difficult to comprehend. Let's write a program that tries to help us understand this big number.

## Program Specifications

1. Your program will prompt for two numbers (hint: use the `input` function):

- The size of the national debt in dollars (an integer).
  (You can look up this value online and then enter the value you find at the prompt: try http://www.brillig.com/debt_clock )
- The denomination of U.S. currency bills you want to use (an integer).
  ($100 is the largest, common U.S. bill—the curious can find more detail at http://en.wikipedia.org/wiki/Large_denominations_of_United_States_currency
- There is no need to check for erroneous input—we learn that later.

2. Your program will perform some simple arithmetic calculations.

3. Your program will print out two numbers (hint: use the `print` function):

- The height in **miles** of your chosen bill denomination required to equal the debt number that was input (output a floating point value).
- The calculated distance in miles translated to a multiple of the average distance to the Earth's moon (output a floating point value).

## Useful Facts

Some useful facts:

- The thickness of every U.S. denomination bill is 0.0043 inches (http://www.trivia-library.com/a/money-in-the-us-history-of-the-dollar-bill.htm)
- The average distance to the moon is 238,857 miles (http://en.wikipedia.org/wiki/Lunar_distance_(astronomy))

- The word **string** in computer science has a special meaning. It means "a sequence of characters", delimited by quotes. We'll see it used often in our work. It differs from a number: "12" is different from 12.

**Assignment Notes:**

To input the numbers it is necessary to use the `input` function. The `input` function takes a string, a sequence of characters between quotes, as a prompt to print to the user. It then waits until the user types a response, terminated by the user typing the Enter key. A string, again as a sequence of characters, is returned.

The returned string must be converted to a number. Integers for this assignment (hint: use `int`). A typical interaction would be something like:

```
num_str = input('Enter a number: ')
my_int = int(num_str)
```

`print` is a command that will print on the Python shell window any combination of variables, values and strings. Each item to be printed must be separated from other items by a comma. All the items will be printed together, followed by a new line. For example:

```
bills_float = 3.14159
print('The number ',bills_float,' times two is ', bills_float*2)
```

This command has 4 items to print: a string ('The number '), the value in the variable bills_float (3.14159), another string (' times two is ') and the result of an expression (3.14159 * 2). Thus the statement will print:

```
The number 3.14159 times two is 6.28318
```

Look at the program numberInput.py in the proj01 directory as an example of using `input`, `print` and `int`.

Once converted to numbers, the operations on these numbers are, respectively: **+** (sum), **-** (difference), ***** (product), **/** (division) and **%** (remainder). See Chapter 1 for details.

To clarify the problem specifications, we provide at the end of this document a snapshot of interaction with the already written program.

**Getting Started**

1. Work on the problem-solving concept first, i.e. solve the problem with pencil and paper without considering Python. Use the values in the Sample Interaction below to check your work. If you cannot solve it with pencil and paper, you certainly cannot solve it using Python. Collaboration is encouraged on this step.
2. Often an easy next step is to use a sample program we provide as a starting point.

   a. In Spyder you can open the sample program and save it with specified assignment name (proj01.py) or create a new program (proj01.py) and copy and paste in the sample contents.
   b. Change the variable names to be more appropriate for this assignment.
   c. Change the arithmetic using the formulas you developed when you solved the problem using pencil and paper.
   d. Change the print statements.
3. Test your program. Begin using values from the Sample Interaction, but do more tests than that.

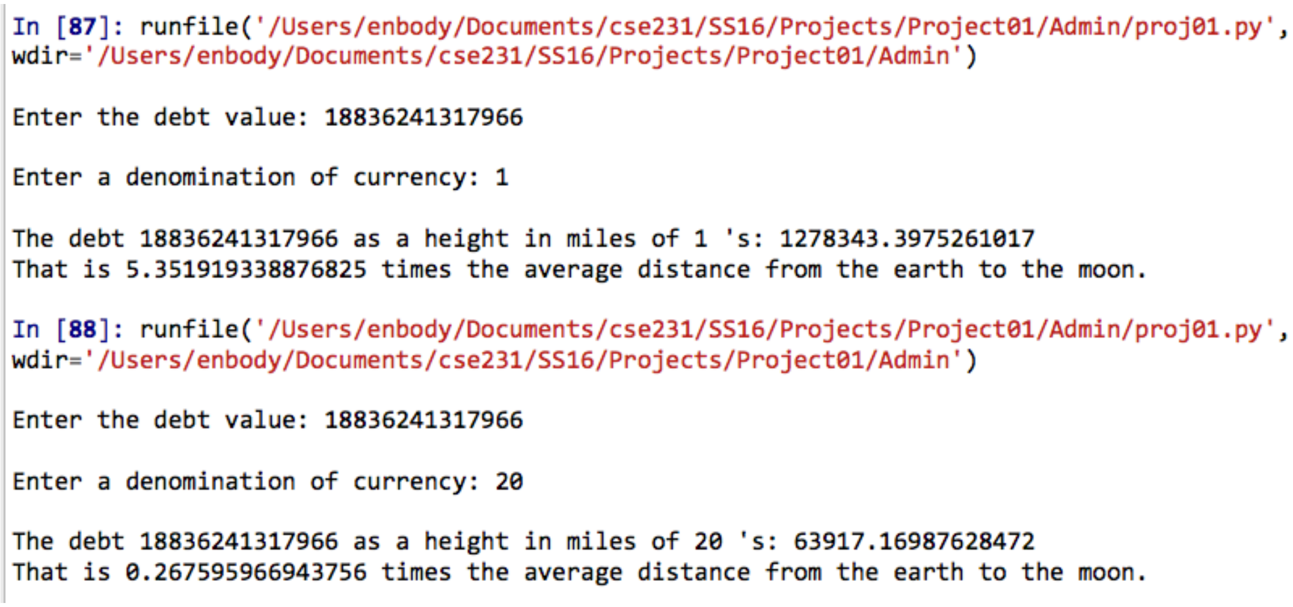
**Questions for you to consider (not hand in)**

1. What happens when you try to divide by zero when you run your program (for example, enter a 0 for the bill denomination)?
2. What happens when you enter a letter instead of a number at the prompt?

(Later in the course we will develop techniques to check for erroneous input.)

**Sample Interaction**

(Later in the course we will develop techniques for formatting the output to look nicer.)



```
In [87]: runfile('/Users/enbody/Documents/cse231/SS16/Projects/Project01/Admin/proj01.py',
wdir='/Users/enbody/Documents/cse231/SS16/Projects/Project01/Admin')

Enter the debt value: 18836241317966

Enter a denomination of currency: 1

The debt 18836241317966 as a height in miles of 1 's: 1278343.3975261017
That is 5.351919338876825 times the average distance from the earth to the moon.

In [88]: runfile('/Users/enbody/Documents/cse231/SS16/Projects/Project01/Admin/proj01.py',
wdir='/Users/enbody/Documents/cse231/SS16/Projects/Project01/Admin')

Enter the debt value: 18836241317966

Enter a denomination of currency: 20

The debt 18836241317966 as a height in miles of 20 's: 63917.16987628472
That is 0.267595966943756 times the average distance from the earth to the moon.
```