

CS160 Python Programming Assignment

Spring 2017

Instructions

- A. For questions 1 through 9 simply type your responses in a Google Docs document and share it with your instructor. It is not necessary to save the code for questions 1 – 9 but you must save your answers!
- B. Questions 1 – 9 are designed to get you familiar with the Python environment, syntax, semantics, types, and methods of the language. There are two modes in the Python Integrated DeveLopment Environment (IDLE): interactive mode and script mode (Your instructor has demonstrated these two modes to you).
- C. The programs for questions 1 – 9 will be written in **interactive mode**. The programs for the remaining questions must be written in **script mode** and each one saved as a separate file with a **.py** extension.
- D. Submit your work by saving your files into the Google Drive folder you've shared with your instructor. Be sure to put the full names of everyone in your group at the top of the word processor file and put a comment block at the top of each of your Python files with your full names, class, and question number.
- E. Turn in your solutions by 5:00 p.m. on May 31. Late submissions are not allowed and will receive a zero grade.
- F. Use the python.org website (try the tutorial) to help you with this project. As an example: if you don't know what a string is, look it up in the tutorial on the python.org site. Also, remember the **Python Programming Beginner Tutorials** I posted a link to on the class web site. They should prove very valuable.

Good luck and do the best you can – it is not necessary to answer all the questions to get a passing grade. This assignment is designed to allow you to “educate” yourself through analysis, trial and error programming, and teamwork. Good Luck!

Part 1

Open the Python IDLE interactive mode, implement the code, and answer the following questions (create a Word or Google Docs file for your answers):

1. Input the following code segment in the interactive mode

```
number = 17  
print (number + 8)
```

What is the result?

2. Input the following code segment in the interactive mode

```
number = 17  
Print (number + 8)
```

What is the result? (1 point)

Why is this different from the result in Question number 1?

3. Input the following code segment in the interactive mode

```
number = '17'  
print (number+8)
```

What is your result? (1 point)

Explain what happened and why this result occurred.

4. Input the following code segment in the interactive mode

```
number = '17'  
print (number+'8')
```

What is the result? (1 point)

Explain what happened and why this result occurred. (1 point)

5. Input the following code segment in the interactive mode

```
count = 6  
print ('count is ' + count)
```

What is the result? (1 point)

Change the last line to be: `print ('count is' + str(count))`

Explain what happened and why this result occurred.

6. Input the following code segment in the interactive mode

```
print (25/10)
```

What is the result? Is this result accurate?

7. Input the following code segment in the interactive mode

```
print (25//10)
```

What is the result? (1 point)

Explain what happened and why this result occurred.

8. We discussed in class the difference between syntax and semantics. Briefly define each and give an example using the Python language for each.
9. Using the tutorial on the Python website explore and discover various uses of the syntax, mathematical operators, string methods, type conversions, operators, branch structures, loop structures, and user input to become familiar with their operation and purpose. Also, generate an executable program through the script mode. Ask questions, but try to find the answer first through experimentation, discussion with others, and analysis. In your word processor document describe both what you did and what you learned.

Part 2

Use IDLE's script mode to write your solution to the following question. Remember to save each solution separately in its own file with a .py extension. Also, remember to include a comment block at the top of each file with your full name, class, and question number.

10. First, write the pseudocode (put this in comment block in your Python file) and then use Python to write a program that accepts a single test score from 0-100 points and assigns a letter grade based on the scale of 90, 80, 70, 60, 59 and below is an 'F' grade.
11. Use the `for` and `while` iterative structures within Python to program a loop that generates a list of odd numbers from 1 to 33 inclusive (write two sets of code here, one `for` loop and one `while` loop).

Bonus: try doing this using `range()`.

12. Write the code to allow a user to input three integer numbers. The program will then produce the average of those three numbers. Design your program with text prompts to make clear to the user what they are to do and the results produced.
13. Using the pseudocode from problem 54 on page 256 of your book implement the program in Python, test the program segment to ensure it is correct using various inputs and answer the question in a comment block in your code file. If this code needs to be corrected what would you add to fix it?
14. Using the pseudocode from problem 5 on page 250 of your book, implement the program in Python and answer the question. How would you correct this code? Put these answers in your Python code file as comments.
15. Python contains a random number generator that is implemented by first importing the random module and then calling the method `randrange()` to generate a random number.

Input the following code:

```
import random
print ( random.randrange(5) )
```

Execute the print statement several times. Put the print line in a loop to print a random value at least 20 times and record your results.

Answer the following questions in comments in your Python code file. What range of numbers does the print statement produce? If you wanted to randomly generate the numbers from 1 to 5 what code would you have to add to the above print statement to do this? What would you change to get a random number in the range 1-100?

Part 3

Now for some more in-depth projects...

16. **Guess the Number**

Your program will first randomly generate a number unknown to the user. The user needs to guess what that number is. (In other words, the user needs to be able to input information.) If the user's guess is wrong, the program should return some sort of indication as to how wrong (e.g. The number is too high or too low). If the user guesses correctly, a positive indication should appear. You'll need functions to check if the user input is an actual number, to see the difference between the inputted number and the randomly generated numbers, and to then compare the numbers.

Concepts to keep in mind:

- Random function
- Variables
- Integers
- Input/Output
- Print
- While loops
- If/Else statements

17. **Mad Libs Generator**

The Goal: Inspired by Summer Son's Mad Libs project with Javascript. The program will first prompt the user for a series of inputs a la Mad Libs. For example, a singular noun, an adjective, etc. Then, once all the information has been inputted, the program will take that data and place them into a premade story template. You'll need prompts for user input, and to then print out the full story at the end with the input included.

Concepts to keep in mind:

- Strings
- Variables
- Concatenation
- Print

18. Hangman

Despite the name, the actual “hangman” part isn’t necessary. The main goal here is to create a sort of “guess the word” game. The user needs to be able to input letter guesses. A limit should also be set on how many guesses they can use. This means you’ll need a way to grab a word to use for guessing. (This can be grabbed from a pre-made list. No need to get too fancy.) You will also need functions to check if the user has actually inputted a single letter, to check if the inputted letter is in the hidden word (and if it is, how many times it appears), to print letters, and a counter variable to limit guesses.

Concepts to keep in mind:

- Random
- Variables
- Boolean
- Input and Output
- Integer
- Char
- String
- Length
- Print