Lab Assignment Week 08

CSC/DSCI 1301 – Principles of CS/DS I

Week of March 3rd, 2025

Introduction

Welcome to the seventh programming lab of CSC/DSCI 1301! Today we will be covering the following topics:

- Nested Loops
- Loop Keywords
- For-Loop Else Statements
- Creating User-defined Functions

Lab policy reminders:

- Attendance is mandatory.
- Labs must be completed **individually.**
- TAs are here to help you. Ask them for help!
- Lab assignments are due at the end of each lab.

Comments

The lab assignment requires the inclusion of comments to enhance code readability and understanding. Specifically, a block comment at the beginning of the Python file is required. Your block comment should include the following:

- The program name
- The author's name (your name)
- A description of the program's overall purpose

Additionally, inline comments should be used throughout the code to explain specific lines or sections that might be less obvious to someone reading the code. These inline comments can clarify complex calculations, explain the purpose of certain variables, or provide additional context for specific code blocks.

Deliverables:

1. Python files for all 3 programs in the lab
2. Screenshots of program output for all 3 programs

If you have any questions, please do not hesitate to ask your TA!

Program 1: guess_my_number.py

Write a number-guessing game that generates a random number between 1 and 100. The player will have only 10 attempts to guess the number through the terminal. After each guess by the player, your game will need to provide feedback to the player about if their guess is greater than or less than the random number.

If the player guesses correctly, your game must acknowledge their success, and the game should end immediately. ***Hint***: You will need to use the ***break*** keyword.

If the player runs out of attempts, you will need to output a message that tells them that they have run out of attempts. ***Hint***: You will need to use the for-loop ***else*** statement to accomplish this.

You may assume that the player will only enter valid integers.

Example Output

```
I have generated random number between 1 and 100. You will have 10 attempts to guess that number.
Guess 1: 50
Your guess is greater than my random number. Try Again.
Guess 2: 25
Your guess is less than my random number. Try Again.
Guess 3: 40
Your guess is less than my random number. Try Again.
Guess 4: 45
Your guess is less than my random number. Try Again.
Guess 5: 47
You correctly guessed my random number!
```

Skills Covered

- Nested Loops
- Loop Keywords
- For-Loop Else Statements

Deliverables

For this program, you will need to provide the Python file containing your code and a screenshot of your program's output. Please name your files as follows:

- Python Files
  - lastname_firstname_filename.py
  - For example: **hawamdeh_faris_guess_my_number.py**
- Screenshots
  - lastname_firstname_filename.png
  - For example: **hawamdeh_faris_guess_my_number.png**

Program 2: steps.py

Fitness apps treat walking 1 step as walking 2.5 feet. Define a function named feet_to_steps() that takes a float as a parameter, representing the number of feet walked, and returns an integer that represents the number of steps walked. Then, write a main program that reads the distanced walked in feet as an input, calls function feet_to_steps() with the input as an argument, and outputs the number of steps as an integer.

Use floating-point arithmetic to perform the conversion.

Example Output

```
Please enter the distance travelled in feet: 183.4
73 steps
```

Skills Covered

- Creating User-defined Functions

Deliverables

For this program, you will need to provide the Python file containing your code and a screenshot of your program's output. Please name your files as follows:

- Python Files
  - lastname_firstname_filename.py
  - For example: **hawamdeh_faris_steps.py**
- Screenshots
  - lastname_firstname_filename.png
  - For example: **hawamdeh_faris_steps.png**

Program 3: text_filter.py

In many chat and messaging programs, automated programs or bots are used to filter banned words from being sent to users. Define a function named text_filter() that takes a string as a parameter, representing the message that is about to be sent, and returns a new string that has the words in the list of banned words removed. Then, write a main program that reads the message string as an input, calls the text_filter() with the input as an argument, and outputs the new filtered text.

For this program, your banned words list will only contain the following five words.

1. Turkey
2. Dog
3. Fox
4. Cat
5. Chicken

Hints

- The membership (in) operator can be used to test if a String is in the list of banned words.
- The built-in String method split() will separate a String at the whitespaces.
    - The split() method returns a list of words where each word is a list item
    - You can read more about this method here.

Example Output

```
>: The Quick Brown Fox Jumps
Input Message: The Quick Brown Fox Jumps
Output Message: The Quick Brown Jumps
```

Skills Covered

- Creating User-Defined Functions
- Using String Methods

Deliverables

For this program, you will need to provide the Python file containing your code and a screenshot of your program's output. Please name your files as follows:

- Python Files
    - lastname_firstname_filename.py
    - For example: **hawamdeh_faris_text_filter.py**
- Screenshots
    - lastname_firstname_filename.png
    - For example: **hawamdeh_faris_text_filter.png**