Lab 3.0
Loops

Z. Hutchinson
zachary.s.hutchinson@maine.edu

September 27, 2022

Goals

The goal of this lab is to practice using Python 3. Specifically, you will practice:

- Loops

Instructions

All work is due at the **end of your lab** and must be submitted to Brightspace in the proper place. Unless otherwise instructed, submissions must be python files (e.g. files that end with *.py*). Any other format, even if it is plain text, will **not** be graded. Messy or otherwise unreadable code will lose points. Lab submissions can be all in the same file, but please label with comments to which task code belongs. IMPORTANT: Any code that is commented out will not be graded. **RUN YOUR CODE TO MAKE SURE IT WORKS!!!**

Task 1 - Loops: Easy Difficulty

Convert the following to Python code.

1. Write a while-loop that prints all numbers between -5 and 5, inclusive.
2. Write a for-loop that prints all numbers between -5 and 5, inclusive.

Task 2 - Loops: Medium Difficulty

For some of the following subtasks you will need the following lists. The list you need will be referenced in the subtask.

```
L = ["A", "B", "C", "D", "E"]
```

1. Write a pair of nested for-loops (loop inside a loop). The outer for-loop should iterate (loop) 10 times. The inner loop should iterate 5 times for each outer iteration. Your program should contain a counter that counts how many total times the inner loop iterates. Print the value in the counter after both loops finish.
2. Write a pair of nested while-loops. The outer while-loop should iterate 4 times. The inner loop should iterate 8 times for each iteration of the outer loop. Your program should contain a counter that counts how many total times the inner loop iterates. Print the value in the counter after both loops finish.
3. Use list *L*. Write a while-loop that prints all the values in *L* starting with "A".
4. Use list *L*. Write a for-loop that prints all the values in *L* starting with "A".

Task 3 - Loops: Hard Difficulty

For the following subtasks you will need the following lists. The list you need will be referenced in the subtask.

```
mylist = [2, 11, 12, 45, 52, 808, 7, 68, 91, 1013]
first_names = ['Lisa', 'Bob', 'Carl', 'Mohammed', 'Vlad', 'Aina']
last_names = ['Smith', 'Zhang', 'Karlson', 'Lee', 'Numan', 'Musa']
```

1. For this subtask, use the list *mylist*. Use a for-loop to iterate through mylist. The loop should divide all even numbers by 2 and store them back in the list in the same place. Print the list after the loop ends. The final loop should be: mylist = [1, 11, 6, 45, 26, 404, 7, 34, 91, 1013].
2. Write a while-loop that combines the first and last names (from the *first_name* and *last_name* lists) and prints each full name. The names should be combined in order. For example, the first full name printed should be: Lisa Smith. NOTE: You do not need to store the combined names anywhere...just print them.
3. Repeat the previous task, except start with the first name, "Carl". When your loop reaches the end of the first names it should start back at the beginning. The last names should stay in the same other. The first name printed should be, "Carl Smith", and the last name printed should be "Bob Musa".

Task 4 - Loops: Insane Difficulty

For the following subtasks you will need the following lists. The list you need will be referenced in the subtask.

```
alist = [4, 5, 88, 32, 99, 88, 73, 68, 91, 1024]
```

1. Using the list *alist*, iterate through *alist*. Continually divide each number by 2 until it is not even. Print the final odd number. You do not need to store the odd number, just print it. HINT: Nested loops.
2. Ask the user for a whole number greater than 2. Print all prime numbers from 2 up to and including the number the user entered.
3. Write a program that models a menu system which contains two different games. This uses several nested sentinel loops. Each loop should have its own control variable. There is an outer loop for the main menu. And there are two independent inner loops for each game. Each inner should should also print "Playing [GAMENAME]...You win!" Here are the questions asked by each loop (NOTE: it is possible to quit the entire program from within each inner loop. This is challenging. See example):
   (a) Outer loop: "Do you want to play Galaga or Pacman or Quit to desktop [G/P/Q]?
   (b) Galaga Inner loop: "Galaga Menu: [c] continue playing Galaga, [q] quit playing, [x] exit to desktop?"
   (c) Pacman Inner loop: "Pacman Menu: [c] continue playing Pacman, [q] quit playing, [x] exit to desktop?"

Example of the final subtask of Task 4:

```
Do you want to play Galaga or Pacman or Quit to desktop [G/P/Q]? G
Playing Galaga...You win!
Galaga Menu: [c] continue playing Galaga, [q] quit playing, [x] exit to desktop? q
Do you want to play Galaga or Pacman or Quit to desktop [G/P/Q]? P
Playing Pacman...You win!
Pacman Menu: [c] continue playing Pacman, [q] quit playing, [x] exit to desktop? x
<program quits>
```