#



Python

Offensive and Defensive Tool Construction

Table of Contents

[EVALUATION: 3](#_Toc72189222)

[Objectives 4](#_Toc72189223)

[Background Reading 4](#_Toc72189224)

[Important Information 4](#_Toc72189225)

[Problem 1 (7pts) 5](#_Toc72189226)

[Introduction: Why Python? 5](#_Toc72189227)

[Problem 2 (7pts) 6](#_Toc72189228)

[Problem 3 (11pts) 6](#_Toc72189229)

[Questions (show a screenshot for each of your answers if applicable): 7](#_Toc72189230)

[Problem 4 (12pts) 7](#_Toc72189231)

[Problem 5 (15pts) 9](#_Toc72189232)

# EVALUATION:

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | Problem 1 | 7 |  |
| 2 | Problem 2 | 7 |  |
| 3 | Problem 3 | 11 |  |
| 4 | Problem 4 | 12 |  |
| 5 | Problem 5 | 15 |  |
|  | TOTAL MARK | 52 |  |

Offensive and Defensive Tool Construction

Python Programming I

Objectives

This lab focuses on the following objectives:

* Explain the purpose of scripting languages and Python.
* Explore the basic syntax of Python and compare it to C.
* Use variables, expressions and statements in Python.
* Explain function definitions and function calls (pure and with return value).

Background Reading

* Read chapters 1–5 in *How to Think Like a Computer Scientist: Learning with Python*, available at [www.greenteapress.com/thinkpython/thinkCSpy.pdf](http://www.greenteapress.com/thinkpython/thinkCSpy.pdf).
* <https://docs.python.org/3.8/>

# Important Information

**YOU MUST PRESENT IMAGES OF YOUR CODE BEING EXECUTED. DO NOT SUBMIT YOUR ANSWERS IN THE DOCUMENT. CREATE A BLANK DOCUMENT AND SUBMIT YOUR ANSWERS THERE.**

**YOU WILL LOSE MARKS FOR NOT FOLLOWING THE ABOVE REQUIREMENTS.**

All scripts must have the following elements:

1. File and Header comments, which follows the following format:

***# Filename: m##XXX.py***

***# Author: Taylor Swift***

***# Course: ITSC203***

***# Details: This program calculates the rate at which users allow themselves # to be hacked.***

***# Resources: https://www.cs.siue.edu/programming-style-guide***

1. Comments on lines where you used some unique computation that might be tricky to comprehend a month later.

***list1 = [x for x in range(20) if x % 4 == 1] # Using list comprehension to ….***

# Problem 1 (7pts)

## Introduction: Why Python?

Both C and Python are high-level languages, and both need to be translated into machine code to execute.

In the table below, list 3 differences between the two languages. Also draw a simple diagram of the path from source code to program execution: **5pts**

|  |  |
| --- | --- |
| **C Program** | **Python Program** |
| No classes  Pointers  Small library of built in functions  C program > preprocessor > expanded source code > compiler > assembly code > assembler > object code > linker > executable code > loader > execution | Classes  No pointers  Large library of built in functions  Python program > compiler > byte code > virtual machine > execution |

You have learned how to program in C. Examine this simple program in C:

#include <stdio.h>

void main (int argc, char \*\*argv)

{

printf ("Hello World!\n");

}

1. Write the equivalent program using Python. **2pts**

#!/usr/bin/env python3

print(“Hello World!”)

# Problem 2 (7pts)

Convert the following C program to the python equivalent:

#include<stdio.h>

int main()

{

int arrint[ ] = {100, 200, 300, 400, 11};

printf("\n\* \* \* \* %d\n", arrint[0]**);**

printf("\n \* \* \* %d\n", arrint[1]**);**

printf("\n\* \* \* \* %d\n", arrint[2]**);**

printf("\n \* \* \* %d\n", arrint[3]**);**

}

1. After converting the program, execute the code and provide screenshots of the program running. **2pts**
2. You will also provide your completed python code. **5pts**

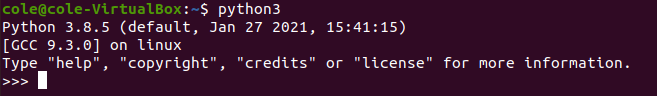
# Problem 3 (11pts)

Open a terminal and follow the instructions below:

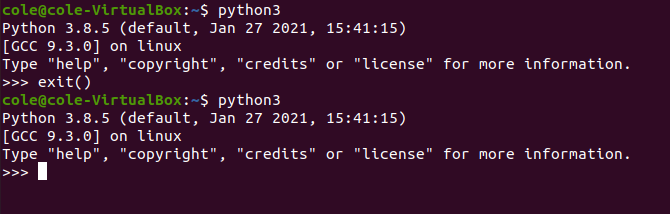
***Note you may not have python3 installed if not you will need to install it:***

***sudo apt-get install python3***

1. Type **python3**
2. Press Enter
   1. The interface presented is called REPL (Read Evaluaute Print Loop) interactive environment. It is a single line interface that allows you to type python command in a similar way to the linux terminal.
3. Take a screenshot of the information presented by the interpreter. **1pt**

****

1. Type **exit()**
2. Type **python3**
3. Press Enter
4. Take a screenshot of the information presented by the interpreter. **1pt**



1. Type **CTRL + D**
2. Enter the **python3** REPL interface
3. Type **help()**
   1. You can type keywords you want help with eg (**int, float, keyword**)
   2. Type **q** after you have finished reading the help
4. Exit the interactive help (do a bit of research)
5. Type help(**tuple**)
6. Scan through the list of items. Take note of the different **entries in the output**.

## Questions (show a screenshot for each of your answers if applicable):

1. What version(s) of REPL were you able to launch in the steps above? **1pt**

**Python 3.8.5**

1. How can you return to the Linux terminal after finishing your testing in REPL? **1pt**

**Exit() or CTRL + D**

1. How do you know that you are in the interactive shell and not at the linux command prompt? **1pt**

**The prompt is different, instead of cole@cole-VirtualBox:~$ its >>>**

1. How do you exit the interactive help? **1pt**

**q**

1. Using **help** what function within the **list** class would you use to remove all items from the **list**? **5pts**

**clear(mylist)**

**mylist.clear**

mylist = [“Really”, 45, (“on top”, “bottom”, 12), 12.45, 12, 12]

|  |  |
| --- | --- |
| Functionality Required | List function used |
| 1. Get the number of times 12 is in the list | count(mylist, 12), mylist.count(12) |
| 1. Get the current position of the number 12.45 | index(mylist, 12.45), mylist.index(12.45) |
| 1. Reverse the order of the list | Reverse(mylist), mylist.reverse |
| 1. And the Word Property to the beginning of the list | insert(mylist, 0, “Property”), mylist.insert(0, “Property”) |
| 1. Remove the word Really from the list | remove(mylist, “Really”), mylist.remove(“Really”) |

# Problem 4 (12pts)

Write a Python program named **m1p4.py** that prints a table of the provided user information provided below.

You have been given the following list that contains:

1. The user’s full name and the home directory
2. The table must be created such that the width of each column is only 2 spaces wider than the longest string.
3. Notice that the names in the table have to be title case.
4. Notice that the folder names capitalize the first letter of each name and also only use the first initial of the first name.
5. Submit your code and an image showing the table output.

**[['kenny rogers', '/home/users/KRogers'],[ 'tony robbins, '/home/TRobbins'],[ 'johnny cash', '/home/users/JCash'],[ 'tito jackson', '/home/hut/TJackson'],[ 'tim tzuyu', '/home/users/TTzuyu'], ['kareena kapoor', '/home/users2/KKapoor']]**

commandprompt$ m1p4.py

+--------------+---------------------+

| Kenny Rogers | /home/users/krogers |

| Tito Jackson | /home/hut/titoj |

+--------------+---------------------+

# Problem 5 (15pts)

Analyze the following code and complete the following steps:

1. Find and fix any errors in syntax or logic **10pts**
   1. You can also remove lines that will not work, but you must explain why you removed that line
2. Document what you did to make the program work: **3pts**
   1. Explain what was wrong
   2. Explain what **you** did to fix the issue
3. Submit the corrected, commented code along with a screenshot of the program’s output. **2pts**

S