Mr. Jason Myers

Wayne state university | information technology

Weekly Individual Assignment - # 5

CSC 4110, due February 17TH, 2022

|  |  |
| --- | --- |
| **Table of Contents** | **page** |
| **Assignment Instructions / Objectives**  **Software Requirements**  **General Requirements (sample commenting/ PEP8)**  **Deliverable Instructions**  **Debriefing Form**  **Rubric**  **Assignment** | **1**  **1**  **1**  **3**  **4**  **5**  **6 through End of Doc** |
| **Written 01-27-22** |  |

**Assignment Instructions:**

Do problem(s)/ resolve issues, following instructions explicitly (see Rubric). Any suspicion of group work (unless specifically stated) will result in a grade of 0. Problems / issues on final page.

**Due:**

See upload folder; due date is firm; NO EXCEPTIONS.

**Objectives:**

**Fulfill customer request(s) / resolve customer issues (see Assignment).**

**Software Allowed/ Required:**

Python3.x “script mode”, Github (desktop)

Note: Word, Winzip, any text editor, as needed. This does not include any modules, such as *matplotlib, math, or system/sys*.

**GitHub:**

GitHub Video 1: <https://www.youtube.com/watch?v=fJtyf62yAb8>

GitHub Video 2: <https://www.youtube.com/watch?v=GqNAD4XoZ6k>

**Reference following article to create repository so you can load this assignment output:**

**https://docs.github.com/en/desktop/installing-and-configuring-github-desktop/overview/getting-started-with-github-desktop**

**References:** Course videos, supplied resources.

**General Requirements:**

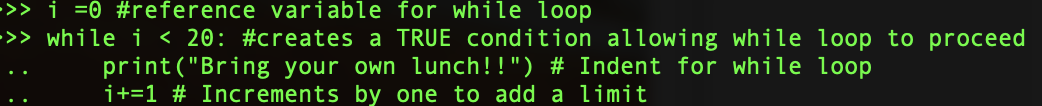
**Add labeling/ comments (name, date, revision #); add in-line requirements where appropriate (such as syntax usage).**

#Indicate coding begin and end

Example acceptable code comment:

**# Revision number BEGIN/ START DATE**

**## Begin John D. Student here (date)**



**# Revision number FINAL DATE**

**## End John D. Student here**

**# Group / manager/ lead tech/ project #**

**Adhere to the following coding style (from PEP8):**

1. Wrap lines so that they don’t exceed 79 characters.
2. Use blank lines to separate functions and classes, and larger blocks of code inside functions
3. When possible, put comments on a line of their own.
4. Name your classes and functions consistently; the convention is to use UpperCamelCase for classes and lowercase\_with\_underscores for functions and methods.

**Deliverable Instructions:**

Upload .py file (not image) to your GitHub (pls change extension to .txt)

Upload link to your .py file in the COMMENTS section of the UPLOAD Folder on CANVAS

See rubric for complete delivery details.

**Debriefing Form:**

|  |  |
| --- | --- |
| Group/ manager / lead tech/ project # | Enter here |
| Planning (Hours /min worked on (billing)) TOTAL | Enter here |
| Execution (Hours/ min worked on (billing)) TOTAL | Enter here |
| Host OS | MAC/ Windows 11/ etc… |
| Platform (if any) | DragonFly, OpenBSD, bare metal, etc… |

|  |  |  |
| --- | --- | --- |
| **Item** | **RUBRIC** | **Pts** |
| 1 | Customer request fulfilled (all issues completed and output consistent with customer expectation) | 60 |
| 2 | PEP8, lines 1 through 4, as appropriate and where necessary | 10 |
| 3 | .py (as .txt) file uploaded to GitHub repository | 10 |
| 4 | On-time (submitted on due date; not before or after) 🡨Note: only LATE submissions penalized | 10 |
| 5 | GitHub Link placed in upload comments section | 10 |
| 6 | Code images and OUTPUT images pasted placed in original assignment, uploaded to course shell (with appropriate comments, etc…) | 10 |
| 7 | Comments are appropriate and explanatory; contain all necessary information | 10 |
| 8 | Debriefing form filled out properly and completely | 10 |
| 9 | Only software mentioned is used (i.e. no extra modules imported) | 10 |
| 10 | External packages not imported, unless explicitly allowed | 10 |

**TOTAL POINTS: 140**

**ASSIGNMENT**

**Customer needs the following issues resolved by stated due date:**

|  |  |  |
| --- | --- | --- |
| WORK TICKET | **CMS DATA REQUEST** |  |
| 1 | Customer needs a program written in Python that performs the following, as appropriate:  For this CMS request, you need to submit a Python program that does the following:   1. Stores your first name as a variable. Use all lowercase letters when you declare it. 2. Stores your last name as a variable. Use all uppercase letters when you declare it. 3. Prints out, "Hello, <first name> <last name>" with the first name converted to uppercase letters and the last name converted to lowercase letters using string functions. 4. Prints out two newlines. 5. Creates a new variable that stores your first and last name together with a space between both parts. 6. Slices your last name from the variable you created in step 5 and prints it out. This must take place on one line. 7. Replaces your last name in the variable you created in step 5 with "<your last name>, Walsh College Student"; print out the new value of this variable. 8. Prints out the following:  "Start by doing what's necessary; then do what's possible; and suddenly you are doing the impossible - Francis of Assisi"Your output must have quotes at the beginning and the end of your outputted text. 9. Stores 2 decimal numbers as variables. 10. Stores one addition, one subtraction, one multiplication, and one division operation of these variables as variables. 11. Prints out each of the four results as:  <numeric value of variable 1> plus <numeric value of variable 2> equals <value of variable that stored the result of addition>  <numeric value of variable 1> minus <numeric value of variable 2> equals <value of variable that stored the result of subtraction>  ...etc. Each output should be on its own line and utilize a different technique for displaying the requested information (concatenation, string formatting expressions, string formatting method calls, f-Strings). 12. Creates a new variable called sq\_root that stores the square root of the variable that holds the result of the multiplication operation you performed in step 10 to two decimal places. Print out this value as:  The square root of <value of variable that stored the result of multiplication> equals <the variable you just calculated for this step>  You may pick your own method for displaying this information. 13. Stores the current month as a string variable (e.g. March, June, etc.) and day of the month as a numeric variable. 14. Outputs "Today is day <day of month> of the month of <month variable>." This should be on a new line and tabbed over two times. You may pick your own method for displaying this information, but it should be different than the technique you used in step 12. | This field for CMS use only |
|  | import math as math  import datetime  if \_\_name\_\_ == "\_\_main\_\_":      date = datetime.date.today()      firstn = input("Enter your first name: ").lower()      lastn = input("Enter your last name: ").upper()      print(f"Hello, {firstn} {lastn}\n\n")      wholename = firstn + " " + lastn      print([char for char in lastn])      wholename\_list = wholename.split(" ")      print(wholename\_list[1] + " Walsh College Student")      print(          "\"Start by doing what's necessary; then do what's possible; and suddenly you are doing the impossible - Francis of Assisi\""      )      num1 = 2.15      num2 = 3.19      add = num1 + num2      sub = num1 - num2      mul = num1 \* num2      div = num1 / num2      print(f"\n{num1} plus {num2} equals {add}")      print(f"\n{num1} minus {num2} equals {sub}")      print(f"\n{num1} times {num2} equals {mul}")      print(f"\n{num1} divided by {num2} equals {div}\n")      """      you could just as easily take it to the 1/2 slash 0.5 power, but importing math      is less lines of code/makes more sense to me personally      """      # Shout out to Carl Lentz, root boy, for this variable name ;)      my\_root = math.sqrt(mul)      print(f"The square root of {mul} equals {my\_root}")      month = str(date.strftime("%B"))      day = int(date.strftime("%d"))      print(f"\n\t\tToday is day {day} of the month {month}")  Enter hours/ min worked on HERE: 45 min  C:\Users\mckow\Desktop\assignment5>C:/Users/mckow/AppData/Local/Programs/Python/Python39/python.exe c:/Users/mckow/Desktop/assignment5/assignment5.py  Enter your first name: parker  Enter your last name: mckowen  Hello, parker MCKOWEN  ['M', 'C', 'K', 'O', 'W', 'E', 'N']  MCKOWEN Walsh College Student  "Start by doing what's necessary; then do what's possible; and suddenly you are doing the impossible - Francis of Assisi"  2.15 plus 3.19 equals 5.34  2.15 minus 3.19 equals -1.04  2.15 times 3.19 equals 6.858499999999999  2.15 divided by 3.19 equals 0.6739811912225705  The square root of 6.858499999999999 equals 2.618873803756111  Today is day 17 of the month February  C:\Users\mckow\Desktop\assignment5> |  |