**CSCI 1411: Fundamentals of Computing**

**Lab 10**

**Due Date: April 13, 2022**

**Name: Malachi Milton-Brewer**

**Goals:**

* Use of decision structure
* Use of functions and modular programming
* Exception handling
* Testing

**Development Environment:** IDLE

**Deliverables:**

1. This lab handout with 6 screen shots (3 for part I and 3 for part II).
2. Your Python code for Part I of this lab. Name the file using the following format:  
   YourlastnameFirstnameLab10a.py

Example: If your name is Jamal Jones then you will name the file as follows:  
JonesJamalLab10a.py

1. Your Python code for Part II of this lab. Name the file using the following format:  
   YourlastnameFirstnameLab10b.py  
   Example: If your name is Jamal Jones then you will name the file as follows:  
   JonesJamalLab10b.py

How to take a **screen shot**:

* For a Windows 10: Use Snipping Tool to copy and CTRL + V to paste screen shot.
* For Mac: Shift + Command + 4 to copy and CTRL + V to paste screen shot.

**Part I – Skill Practice (10 Points)**

* Start IDLE
* Create a new file.
* Type the following code in the file. …..**Do not cut and paste.** You will learn more by typing it in.
* Make sure that you read all comments to understand the code
* Remember to update the first 2 lines with your own name and the date of the lab.

"""

Author: Dr. Salim Lakhani

Date: October 16, 2020

This program will perform the following tasks:

1) It will ask user for three test score

2) It will calculate the mean of the three test scores

3) It will convert the mean into a letter grade (See

find\_letter\_grade function for the mapping)

4) It will display the mean score and letter grade

It will display an error message if the entered score is non-numerical

"""

def calculate\_mean (score1, score2, score3):

"""

Function Name: calculate\_mean

Description: Calculate mean of three test scores

Parameter: score1 - Score for test 1

score2 - Score for test 2

score3 - Score for test 3

Returns mean of the three test scores

"""

mean = (score1 + score2 + score3) /3

return mean

def find\_letter\_grade (mean):

"""

Function Name: find\_letter\_grade

Description: Convert the mean into letter grade as follows:

Mean Letter Grade

90 to 100 A

80 to 89 B

70 to 79 C

60 to 69 D

0 to 59 F

If the acore is above 100 or below 0 then it will return

letter grade to be "undefined"

Parameter: mean- mean of the test score

Returns list of numbers

"""

if mean > 100:

letter\_grade = 'Undefined'

elif mean >= 90:

letter\_grade = 'A'

elif mean >= 80:

letter\_grade = 'B'

elif mean >= 70:

letter\_grade = 'C'

elif mean >= 60:

letter\_grade = 'D'

elif mean >= 0:

letter\_grade = 'F'

else:

letter\_grade = 'Undefined'

return letter\_grade

def main():

"""

Function Name: main

Description: Ask the user for three test score, use calculate\_mean

function to calculate the mean, use find\_letter\_grade

function to convert mean into letter grade, and display

mean and letter grade.

It uses try/except construct to catch any errors and

displays appropriate error messages

Parameter: None

Returns None

"""

print ('This program will calculate the mean of three test grades and')

print ('convert the mean into a letter grade')

print ('=============================================================')

# Use tr/except block to catch errors

try:

score1 = int (input ('Enter score 1: '))

score2 = int (input ('Enter score 2: '))

score3 = int (input ('Enter score 3: '))

mean = calculate\_mean (score1, score2, score3)

letter\_grade = find\_letter\_grade (mean)

print ('Your mean score is {0:0.3f}'.format (mean))

print ('Your letter grade is:', letter\_grade)

# If the input is a non-numerical string then display an error message

except ValueError as err:

print ('One of you score is non-numerical')

print ('Please try again with correct scotes')

# Catch unexpected errors

except:

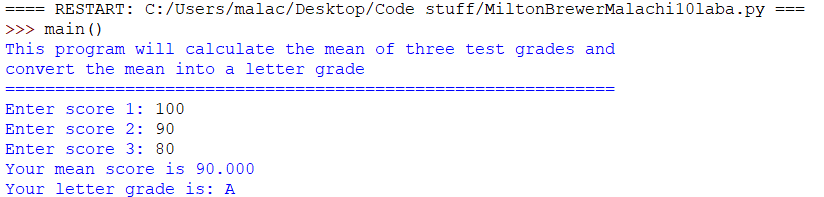
print ('Unknown error')

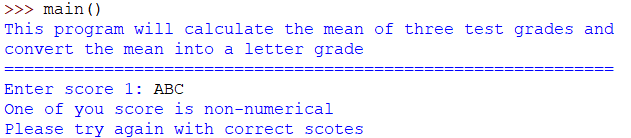
* Save the file as “YourLastNameYourFirstNameLab10a.py”
* Click Run -> Run Module
* Type main() to run your program
* If there are any syntax errors, then fix those errors and run your program again.
* Use the test scores given in the following table to test your program for different possible outcomes.
* If you get the correct result (shown in the last two column) then your program is working as expected.

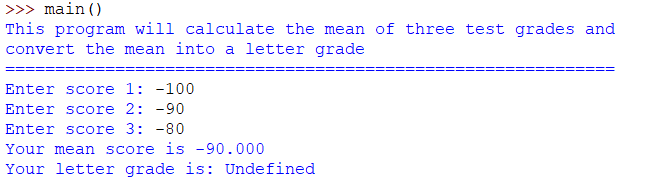
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Run Number** | **Test Score 1** | **Test Score 2** | **Test Score 3** | **Output** | |
| **Mean** | **Letter Grade** |
| 1 | 100 | 90 | 80 | 90.000 | A |
| 2 | 66 | 33 | 88 | 62.333 | D |
| 3 | ABC |  |  | Display Error Message | |
| 4 | -100 | -90 | -80 | -90.000 | Undefined |
| 5 | 101 | 102 | 103 | 102.000 | Undefined |

* Once you are satisfied with your results then take a screen shot of run number 1, 3 and 4 and past them below.

**Paste your screen shot below this line**

****

****

****

**Part II – Cost Calculator (15 Points)**

A software company sells a package that retails for $99. Quantity discounts are given according to the following table:

|  |  |
| --- | --- |
| **Quantity** | **Discount** |
| 10 to 19 | 20% |
| 20 to 49 | 30% |
| 50 to 99 | 40% |
| 100 or more | 50% |

Write a program that asks the user to enter the number of packages purchased. The program should then display the discount percentage, amount of the discount (can be 0) and the total amount of the purchase after the discount. Format the output to two decimal places, include % sign after the discount percentage and $ before the discount amount & total amount. Your program should include the following functions:

1. discount\_percentage – This function will receive number of packages purchased as a parameter and will return the discount percentage.
2. main – This function will perform the following tasks:
   1. Ask user for the quantity
   2. Use discount\_percentage function to calculate the discount percentage
   3. Calculate amount (quantity \* 99)
   4. Calculate discount amount (amount \* discount percentage /100)
   5. Calculate total amount (amount – discount amount)
   6. Display discount percentage, amount of discount, and total amount
   7. Display an error message if quantity is less than 0 (negative number)
   8. Use try/except construct to catch any errors
3. Click Run -> Run Module
4. Type main() to run your program
5. If there are any syntax errors, then fix those errors and run your program again.
6. Use the quantity given in the following table to test your program for different possible outcomes.
7. If you get the correct results (shown in the last three columns) then your program is working as expected.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Run Number** | **Quantity** | **Output** | | |
| **Discount Percentage** | **Discount Amount** | **Total Amount** |
| 1 | Abc | Display Error Message | | |
| 2 | -5 | Display Error Message | | |
| 3 | 5 | 0.00% | $0.00 | $495.00 |
| 4 | 19 | 20.00% | $376.20 | $1504.80 |
| 5 | 55 | 40.00% | $2178.00 | $3267.00 |
| 6 | 105 | 50% | $5197.50 | $5197.50 |

* Once you are satisfied with your results then take a screen shot of run number 1, 3 and 6 and past them below.

**Paste your screen shot below this line**

* Upload this lab handout with required screen shots and your code file to Canvas to submit the lab.

