

COMPS258

Computer Programming and Problem Solving

Autumn 2021 Presentation

Assignment 1 (TMA01)

Please submit this Assignmentassignment by 17th November 2021

Preamble

Hello COMPS258 students. This assignment allows you to demonstrate your ability in basic programming in Python. Hopefully it will help you to pace your venture in the COMPS258 course. This assignment mainly covers Units 1 to 3.

Your tutors and the COMPS258 Teaching team will be on your side to help you learn how to learn programming. This assignment is pitched at a level so that everyone can make a reasonable attempt and demonstrate your ability. It is important for you to get some marks from this assignment. If you could only finish part of it or even a few questions, submit the assignment anyway. You are not expected to finish the whole assignment at all. Just hand in anything you have done by the cut-off date.

We take a very serious view towards plagiarism. Please study the document Academic writing: acknowledging your sources for details. You are encouraged to discuss academic matters with your course-mates, but your assignment submission must be solely your own work.

Also, we expect you to rigorously observe the various rules of course administration including the submission date and the files to attach to your submission, take initiative to download the assignment and supplements, and make sure your assignment has not been lost. All our actions must comply with the rules of the university.

So wish you Happy Computing and keep the good work going!

Programming Style

Your program solutions should follow a good programming and formatting style, which are outlined in this section. Marks are adjusted for good/poor programming and formatting style.

General Notes

- The programs should be written in a way that they are readable to fellow programmers.
- Comments should be added for programming code written in unusual or difficult ways.
- Insert blank lines to separate logical program segments in a file, and use comments for their description.

```
period = int(input("Enter the number of months to analyse: "))
count = 1
total = 0
# each each month's sales and calculate
while count <= period:
    sales = float(input("Enter sales in month #{}: ".format(count)))
    total = total + sales
    count = count + 1

# print results
print("Total sales = $", total)
print("Average monthly sales = $", total / period)
print("Largest monthly sales = $", largestSales)
```

Program Header

- The program header gives a general and overall description at the beginning of a program file.
 - Description of the program (name of the file, purpose, and known errors and issues)
 - Authoring information (name of the author, date of authoring, versioning if any)

```
#
# OCS Calculator (ocs_calculator.py)
#
# Purpose: This program can calculate the OCS from the OCAS and examination score.
#
# Limitation: This program does not handle negative score
#
# Written by Andrew Kwok Fai Lui
# On 1/9/2020
# For Assignment 1
#
```

Naming Variable and Functions

- The names (i.e. identifiers) given to variables and functions should be descriptive. Readers should be able to guess the significance, such as the roles and purposes, from the names.
- Long identifiers may be abbreviated.
- All uppercase names should be reserved for constants or carrying special meaning.
- Capitalized names may be used for classes.
- Lowercase should be used for variables, functions, and modules.
 - Follow either camel case or use underscore to join multiple names.

```
# Carmel case examples
#
carCounter = nameOfStudents = numberOfOrders = 0
#
# Underscore examples
#
car_counter = name_of_students = number_of_orders = 0
```

Function Headers

- Functions are important programming structure for organization of programming logic. A function should perform a task for a program.
- Include a comment, called a function header, about the purpose of the function, the return value, and some key parameters is helpful for readers.
- The function header may be omitted if the function to write is specified in the assignment.
- You may add additional functions in your programming solutions, and in this case, add a function header.

```
# This function returns the N fibonacci number
# N must be greater than or equal to 0
def fibonacci(N):
    if N == 0:
        return 1
    elif N == 1:
        return 1
    return fibonacci(N-1) + fibonacci(N-2)
```

Question 1 Programming (44%)**Theme: Basic Input and Output**

- (a) Write a Python program that prints the word “MU” in a dot-matrix format made up of asterisks. The size should be at least 8 (height) x 16 (width). Your program needs not print in exact the same style as the example below.

Some of you may think that looks like a “V”, but it is more like a “U”. ☺

Submit your solution in **q1a.py**. [4 marks]

Hint: dot-matrix format means that the characters or symbols are drawn with “dots” made up of a character (such as the asterisk in the following examples).

```
*      * *      *
**     ** *      *
* *    * * *      *
* *    * * *      *
* * *  * * *      *
* * *  * * *      *
*      * * *      *
*      * * *      *
*      *      * * * * *
```

- (b) This question wants you to use the input function to read a string into a program. Then format a message with the string. Write a program that should first ask for you name (as an input string) and then print a happy birthday message for you! The birthday message is shown in the following illustration that you should follow.

The program q1bskeleton.py gives you an example.

```
Enter your name: Andrew Lui
Happy birthday to me. Happy birthday to me. Happy birthday to Andrew Lui.
Happy birthday to me!
```

Submit your solution in **q1b.py**. [4 marks]

- (c) Many online travel services encourages users to give feedback.

Write a program that reads in the name of a hotel(as a string) and the score (as a floating-point), and then prints out the data that has been entered in a specific format as shown below. There are two different executions of the same program.

```
Rate a Hotel System
Enter hotel name: Metro Hotel
Enter score (0-10) for the hotel: 9.8765
Thank you for given 9.9 to Metro Hotel

Rate a Hotel System
Enter hotel name: Hotel Del Luna
Enter score (0-10) for the hotel: 4.123
Thank you for given 4.1 to Hotel Del Luna
```

Specific requirements include:

- Hint: use both the print and input functions.
- Hint: remember to convert input string into a floating point number for the score.
- The output of the rating should be formatted to show rounding to 1 place after the decimal point. Hint: use the `format` function.

Submit your solution in **q1c.py**. [4 marks]

Theme: Operators and Calculation

- (d) Your friend Betsy is selling super mangos. Help her to write a program to calculate the amount to pay given the number of mangos in a purchase. Each super mango is sold at only \$10.50. Follow the input and output format shown in the example below. Assume that the input format is always correct. Submit your solution in **q1d.py**. [4 marks]

```
Enter the number of super mangos: 12
Amount to pay is $126.00
```

- (e) Have you seen the following thing? This is a bird seed block that attracts birds to come and pick their favourite food. This shape is called a cuboid with right angles at all the vertices.



A company offer made-to-measure seed block and it costs \$550,000 per cubic meter. Write a program that reads in the length, width, and height (all in meters) of the block, and then prints out the price. Note that the volume of a cuboid is simply $length \times width \times height$

Assume that the input format is always correct. Submit your solution in **q1e.py**. [8 marks]

```
Made-to-measure seed block
Enter the length(m): 0.1
Enter the width(m): 0.02
Enter the height(m): 0.1
The price is $110.00
```

- (f) Back to the super mango case in part (d). Betsy has decided to give a discount to MU students. MU students who buy 10 super mangos or more, one of the mango is free.

Modify the program so that the new rule applies. Submit your solution in **q1f.py**. [4 marks]

```
Enter the number of super mangos: 10
Amount to pay is $94.50

Enter the number of super mangos: 9
Amount to pay is $94.50
```

- (g) A new Korean Fried Chicken stall called *Metro Chicken* will be open in the university. The stall celebrates the opening with a promotion discount. Each chicken piece is originally priced at \$15. For purchases of 7 or more, each piece after the initial 6 is priced at \$12. If the total purchase amount is \$200 or more, a further 10% discount is given.

Write a program to calculate the final amount payable given the number of chicken pieces.

The program should check the validity of the input (i.e. negative or zero are invalid). In the case of invalid input, print a suitable error message and quit. Assume that the input is always an integer.

```
Metro Chicken
Enter the number of chicken pieces: 0
Sorry the input is not an integer
```

```
Metro Chicken
Enter the number of chicken pieces: 6
Amount to pay is $90.00
```

```
Metro Chicken
Enter the number of chicken pieces: 24
Amount to pay is $275.40
```

Submit q1g.py. [8 marks]

- (h) MU Name Cards is a company printing specially designed name cards. The rates for printing are given in the following table. The prices given are a lot of 500 cards.

Colours	Paper Type		
	A	B	C
1-colour	\$140.0	\$100.0	\$80.0
2-colour	\$180.0	\$120.0	\$90.0
4-colour	\$250.0	\$160.0	\$120.0

Help the company to write a program for calculating the amount to pay for a customer ordering name cards. The program should ask for the paper type (A, B, or C), the number of colours (an integer 1, 2, or 4), and the number of lots of 500 cards. The program should then print the amount to pay. The above input order must be followed.

Assume that the user always follow the required input format. After the input is finished, the program should check if an error input occurred, including a wrong paper type, incorrect number of colours, or zero or negative lot numbers. If there is an error, print a suitable message.

The example shows an execution of the program.

```
MU Name Cards
Enter the required paper type (A, B, C): A
Enter number of colours : 2
Enter number of lots of 500 : 5
The amount to pay is $900
```

```
MU Name Cards
Enter the required paper type (A, B, C): D
Enter number of colours : 3
Enter number of lots of 500 : 0
Error in the input
```

Submit your solution in q1h.py. [8 marks]

Question 2 Programming with Loops (48%)

In the following questions you are required to write loops that generate given output. You will receive no marks if you directly print the output.

- (a) Use the range function with a for structure to print all the even numbers between 0 and 500 (inclusive). Make sure that there is a space between two numbers.

```
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50 52
54 56 58 60 62 64 66 68 70 72 74 76 78 80 82 84 86 88 90 92 94 96 98 100 102
104 106 108 110 112 114 116 118 120 122 124 126 128 130 132 134 136 138 140
142 144 146 148 150 152 154 156 158 160 162 164 166 168 170 172 174 176 178
180 182 184 186 188 190 192 194 196 198 200 202 204 206 208 210 212 214 216
218 220 222 224 226 228 230 232 234 236 238 240 242 244 246 248 250 252 254
256 258 260 262 264 266 268 270 272 274 276 278 280 282 284 286 288 290 292
294 296 298 300 302 304 306 308 310 312 314 316 318 320 322 324 326 328 330
332 334 336 338 340 342 344 346 348 350 352 354 356 358 360 362 364 366 368
370 372 374 376 378 380 382 384 386 388 390 392 394 396 398 400 402 404 406
408 410 412 414 416 418 420 422 424 426 428 430 432 434 436 438 440 442 444
446 448 450 452 454 456 458 460 462 464 466 468 470 472 474 476 478 480 482
484 486 488 490 492 494 496 498 500
```

Submit your solution in **q2a.py** . [4 marks]

- (b) Chris is dreaming to be a millionaire. He has \$10000 savings now. He thinks if he can increase his savings by 15 percent every month, then he can be a millionaire in 50 months. You don't believe this. Doris suggests you to write a program to check.

Write a program that reads in a target amount (say, \$1000000), and then use a while loop structure to print the savings after each number of months. The loop should stop when the savings reaches the target. Finally, a message is printed as follows.

```
Enter target ($): 200000
After 1 months saving is $11500.0
After 2 months saving is $13224.999999999998
After 3 months saving is $15208.749999999996
After 4 months saving is $17490.062499999993
After 5 months saving is $20113.571874999999
After 6 months saving is $23130.607656249987
After 7 months saving is $26600.198804687483
After 8 months saving is $30590.228625390602
After 9 months saving is $35178.76291919919
After 10 months saving is $40455.57735707906
After 11 months saving is $46523.913960640915
After 12 months saving is $53502.50105473705
After 13 months saving is $61527.8762129476
After 14 months saving is $70757.05764488973
After 15 months saving is $81370.61629162318
After 16 months saving is $93576.20873536666
After 17 months saving is $107612.64004567165
After 18 months saving is $123754.5360525224
After 19 months saving is $142317.71646040076
After 20 months saving is $163665.37392946085
After 21 months saving is $188215.18001887997
After 22 months saving is $216447.45702171195
Reached $200000.0 after 22 months
```

Submit your solution in **q2b.py** . [8 marks]

- (c) Write a Python program that uses a for loop to generate a lookup table for the equivalent radians, sine (Sin) and cosine (Cos) of an angle in degrees. The table should contain rows for a consecutive degrees from 0 to 90 with an interval of 5 degrees. It should have 4 columns and each column should occupy 10 spaces. All values to be shown with 4 decimal places.

Degrees	Radians	Sin	Cos
0.0000	0.0000	0.0000	1.0000
5.0000	0.0873	0.0872	0.9962
10.0000	0.1745	0.1736	0.9848
15.0000	0.2618	0.2588	0.9659
20.0000	0.3491	0.3420	0.9397
25.0000	0.4363	0.4226	0.9063
30.0000	0.5236	0.5000	0.8660
35.0000	0.6109	0.5736	0.8192
40.0000	0.6981	0.6428	0.7660
45.0000	0.7854	0.7071	0.7071
50.0000	0.8727	0.7660	0.6428
55.0000	0.9599	0.8192	0.5736
60.0000	1.0472	0.8660	0.5000
65.0000	1.1345	0.9063	0.4226
70.0000	1.2217	0.9397	0.3420
75.0000	1.3090	0.9659	0.2588
80.0000	1.3963	0.9848	0.1736
85.0000	1.4835	0.9962	0.0872
90.0000	1.5708	1.0000	0.0000

Submit your solution in **q2c.py** . [8 marks]

- (d) Write a program that generates a square pattern. The program should first ask the user for the size of the pattern and then generate the pattern accordingly.

You should also include code that checks the validity of the input value. A valid input is between 1 and 20 (inclusive). An invalid input should be rejected with a proper error message rather than printing the pattern.

Submit your solution in **q2d.py**. The example shows three different executions of the program. [4 marks]

```
Enter the size of the pattern: 5
$+++ $
$+++ $
$+++ $
$+++ $
$+++ $

Enter the size of the pattern: 8
$+++++ $
$+++++ $
$+++++ $
$+++++ $
$+++++ $
$+++++ $
$+++++ $
$+++++ $

Enter the size of the pattern: -1
The entered size should fall between 1 to 20
```

- (e) Write a program that again generates a square, this time of a different pattern. You should again include code that checks the validity of the input value. Submit **q2e.py**.

The example shows two different executions of the program. Your program code should be similar to the above program. Printing of digits should have wrap-around feature, i.e. after '9' is '0'. [4 marks]

```
Enter the size of the pattern: 5
```

```
$123$
$123$
$123$
$123$
$123$
```

```
Enter the size of the pattern: 15
```

```
$1234567890123$
$1234567890123$
$1234567890123$
$1234567890123$
$1234567890123$
$1234567890123$
$1234567890123$
$1234567890123$
$1234567890123$
$1234567890123$
$1234567890123$
$1234567890123$
$1234567890123$
$1234567890123$
$1234567890123$
```

- (f) Betsy has been feeling headaches for some time and she suspects the cause is poor indoor air quality. She bought an sensor that provides readings of Indoor Air Pollution Index (IAPI) (a floating point number). The next thing to do is to write a Python program to analyze the data collected over a period.

Write a program that should read the period of analysis (number of days). The program should then read the IAPI of each day. The program should finally print the following results:

- Average IAPI in the period.
- Maximum IAPI
- Minimum IAPI
- Number of days of which the IAPI is over 10.0 (very unhealthy)
- Number of days of which the IAPI is 2.0 or below (very healthy)
- Number of days of which the IAPI is worse (higher) than the previous day's IAPI.

Follow the input and output format as shown in the example below. You can assume that inputs are always a number. However, a valid IAPI is within the range 0 to 20.0 (inclusive). If the input IAPI is invalid, an error message should be printed and the IAPI for the day should be re-entered. Submit your solution in **q2f.py**. [12 marks]

Hint: you are suggested to use a while structure for better control over the loop.

	<div>IAPY Analyser Enter number of days of the IAPY data collection period: 7 Enter the daily IAPY of day 1 : 4.3 Enter the daily IAPY of day 2 : 12.2 Enter the daily IAPY of day 3 : 11.5 Enter the daily IAPY of day 4 : 5.3 Enter the daily IAPY of day 5 : 2.0 Enter the daily IAPY of day 6 : 1.8 Enter the daily IAPY of day 7 : 3.3 Average daily IAPY in the period: 5.7714285714285705 Maximum daily IAPY is 12.2 Minimum daily IAPY is 1.8 Number of days with Very Unhealthy IAPY is 2 Number of days with Healthy IAPY is 2 Number of days with IAPY higher than previous day is 2</div> <div>IAPY Analyser Enter number of days of the IAPY data collection period: 3 Enter the daily IAPY of day 1 : 2.5 Enter the daily IAPY of day 2 : 18.4 Enter the daily IAPY of day 3 : 20.3 Input Error: IAPY should be between 0 and 20.0 Enter the daily IAPY of day 3 : -1.5 Input Error: IAPY should be between 0 and 20.0 Enter the daily IAPY of day 3 : 20.0 Average daily IAPY in the period: 13.633333333333333 Maximum daily IAPY is 20.0 Minimum daily IAPY is 2.5 Number of days with Very Unhealthy IAPY is 2 Number of days with Healthy IAPY is 0 Number of days with IAPY higher than previous day is 2</div>															
(g)	<p>Evaluate the following expressions given in the following. List the order of execution of the operators in the following expression in the way described below, and show the final value of the whole expression.</p> <table><tr><td></td><td></td><td>Assumptions</td></tr><tr><td>(i)</td><td>$a - 4 * b$</td><td>$a = 2$ and $b = 3$</td></tr><tr><td>(ii)</td><td>$a + b / 2 == b + 4$</td><td>$a = 2$ and $b = 3$</td></tr><tr><td>(iii)</td><td>$a**2 > b$ and $3 < 4$</td><td>$a = 2$ and $b = 3$</td></tr><tr><td>(iv)</td><td>$a = b = \text{not}(a > 3)$</td><td>$a = 2$ and $b = 3$</td></tr></table> <p>An example is given below. Each sub-question is treated independently (i.e. use the same assumption values for each sub-question). Submit your work in a file named q2g.txt, which is given to you as the template for your answers. [8 marks]</p> <div><pre>x + y + x * 2 and assumption is x = 2 and y = 9 Order of execution is: 1. x * 2 2. x + y 3. x + y + x * 2 Final value is 15</pre></div>			Assumptions	(i)	$a - 4 * b$	$a = 2$ and $b = 3$	(ii)	$a + b / 2 == b + 4$	$a = 2$ and $b = 3$	(iii)	$a**2 > b$ and $3 < 4$	$a = 2$ and $b = 3$	(iv)	$a = b = \text{not}(a > 3)$	$a = 2$ and $b = 3$
		Assumptions														
(i)	$a - 4 * b$	$a = 2$ and $b = 3$														
(ii)	$a + b / 2 == b + 4$	$a = 2$ and $b = 3$														
(iii)	$a**2 > b$ and $3 < 4$	$a = 2$ and $b = 3$														
(iv)	$a = b = \text{not}(a > 3)$	$a = 2$ and $b = 3$														

Question 3 Mini-Project (8%)

This question is designed for students who want more difficult programming challenge. The 8 marks are relatively small tangible reward but the incentive is about solving the problem giving you satisfaction.

(a)	<p>This question is to consider the following equation, where a, b, and c are integers between 1 and 1000.</p> $a^2 + b^2 = c^3$ <p>The challenge is to write a program to find out how many combinations (a, b, c) that make the equation correct.</p> <p>Obviously if (a, b, c) = (1, 1, 1), the equation is incorrect because $1^2 + 1^2 \neq 1^3$. Here are some examples that the equation is correct: (2, 2, 2), (10, 30, 10), (18, 26, 10).</p> <p>Hint: use three for loops for each of the three variables a, b, and c.</p> <p>The CC and the tutors will not provide any further hint to this question, as it is good for the challenge. No mark is given if the programming solution is not correct. Submit your programming solution in q3a.py and the number of combinations in q3a.txt. [8 marks]</p>
-----	--

Appendix A Submission Summary

The following table shows the files that may be submitted. The submitted files will be computer-processed. Failure to use the correct file names will earn you zero mark. Please note that Python 3.8 will be used to mark your assignment.

Filename	Format	Description
q1a.py	Python Source	Question 1(a)
q1b.py	Python Source	Question 1(b)
q1c.py	Python Source	Question 1(c)
q1d.py	Python Source	Question 1(d)
q1e.py	Python Source	Question 1(e)
q1f.py	Python Source	Question 1(f)
q1g.py	Python Source	Question 1(g)
q1h.py	Python Source	Question 1(h)
q2a.py	Python Source	Question 2(a)
q2b.py	Python Source	Question 2(b)
q2c.py	Python Source	Question 2(c)
q2d.py	Python Source	Question 2(d)
q2e.py	Python Source	Question 2(e)
q2f.py	Python Source	Question 2(f)
q2g.txt	Plain Text	Question 2(g)
q3a.py	Python Source	Question 3(a)
q3a.txt	Plain Text	Question 3(a)

Appendix B Extension Application

Please submit the assignments on or before the cut-off dates.

Your tutor is authorized to grant extension of up to seven days.

Only in the circumstances of sickness (with medical proof), long business trips (at least a week long and happen within 2 weeks of the cut-off dates), and exceptional events would extension of 8 to 21 days be considered. We will reply your extension application through the OLE Assignment Records and Extension System.

No marks will be awarded for any late assignments without prior approval obtained from the Course Coordinator and/or the Dean.

If you are in an exceptionally difficult situation and unable to submit within 21 days, you may make an application to the Dean of School. In such situations, please email us at alui@ouhk.edu.hk and we will tell you what to do.

Appendix C Electronic Submission

Assignment submission is a three-step process.

Step 1. Start working on your assignment as early as possible, so you can submit your work by the cut-off date. Follow the file names specified in the assignment questions. Place all the questions in a folder.

Step 2. Package your work (files) into a ZIP file. ZIP files are a common way in the Internet era for packaging and sending multiple files. You will gather all your work files into one ZIP file using a package such as WinZip. Name your ZIP file with your student ID followed by the suffix .zip (for example, 01234567.zip). This is an additional measure to avoid mistakes.

Step 3. Submit the ZIP file at the OLE (Online Learning Environment) , Records & Extension System (<https://ole.hkmu.edu.hk>)

