



Assessment Details and Submission Guidelines	
Trimester	T2 2019
Unit Code	HS1031
Unit Title	Introduction to Programming
Assessment Type	Group
Assessment Title	Group Assignment
Purpose of the assessment (with ULO Mapping)	Assess student's ability to develop algorithmic solutions to programming problems using Python language.
Weight	20 % of the total assessments
Total Marks	20
Word limit	N/A
Due Date	Week 10
Submission Guidelines	<ul style="list-style-type: none"> • There are three questions in this assignment which require you to write and submit three Python scripts. Please save each script in two different formats: .py and .txt prior to submission. For example, for the first question you need to submit < >.py and < >.txt. • Combine all .py and .txt files into one zip or rar file (6 files in total). Also, remember include a file that lists the names of students who contributed to the assignment (within the zip or rar file). Finally, upload your compressed file (zip or rar) to Black Board and proceed with submission. • Code must be appropriately commented. Make sure to add comments at each segment of your code to explain what it does. • Make sure that your code runs successfully for all possible entries. • Try to approach the solution with the least number of steps. Your code must be clear, logical, and easy to understand. • Your code must be written in Python 3. • This is a group assignment. Each member of the group must make an even contribution to the submission. Please make sure to submit the contributions document with names of those who contributed to the work. • All work must be submitted to Blackboard by the due date (Monday, 23 September 2019 11:55PM). • You are encouraged to avoid last minute submission so that you do not run into technical difficulties. • You are allowed up three attempts. All attempts must take place prior to assignment deadline. • Please note that the Self-check Safeassign link has been removed from Blackboard. You still check your work for plagiarism by directly submitting your assignment. If the score for plagiarism is high, you are welcome to resubmit which will count as a second or third attempt.

- | | |
|--|--|
| | <ul style="list-style-type: none">• Please note that plagiarism is treated seriously. All those caught plagiarising will get zero and get their names added to an institutional register. Scoring a lower mark is much better than falling under the plagiarism watch-list.• Please download the attached submission template and see the step-by-step instructions for submission. |
|--|--|

Group Assignment Specifications

Purpose:

This assignment evaluates your understanding of basic programming principles using Python language. In particular, it assesses your ability to develop algorithms to solve simple problems, successfully develop and run python programs, and your ability to write meaningful comments when necessary. The assignment also provides a platform for students to work together in groups to develop solutions, which resembles how complex programming problems are solved in real-life.

Marking criteria

Question	Marking criteria	Marks
Question 1	Appropriate commenting	2
	Sound logic	2
	Code running successfully	2
Total		6
Question 2	Appropriate commenting	2
	Sound logic	2
	Code running successfully	2
Total		6
Question 3	Appropriate commenting	2
	Sound logic	3
	Code running successfully	3
Total		8
Total Marks		20

1. Pythonize is a new language, that is similar to English, however the first letter of each word in Pythonize is moved to the end. Furthermore, each word in this new language ends with 'py'. Write a program that takes an English word as an input and prints the Pythonize equivalent on the screen. (Save your script as **pythonize.py** and **pythonize.txt**)

For example:

If the input is: "Python"

The output would be: Ythonppy

If the input is: "Programming"

The output would be: Rogrammingppy

Notice that the first letter in a Pythonize word is capitalized (as it is the case in English).

Marks Distribution

Criteria	Comments	Logic	Execution	Total
Mark	2	2	2	6

2. Write a program that calculates the average of numeric values in a string and ignores all other characters. Your program must print the sum and average on the screen. (Save your script as **spotNumbers.py** and **spotNumbers.txt**)

For example:

If the input is: myString= "Biology = 66.0 Python = 90.0 Economics = 80.0 Statistics = 70.0"

The output would be: Sum= 306.0 Average= 76.5

Hint: Consider using *type* build-in function in your code. Also, see documentation or exception handling in Python as it may prove to be handy in your code.

Marks Distribution

Criteria	Comments	Logic	Execution	Total
Mark	2	2	2	6

3. Write a program that takes a string of words, and prints out these words one per line in a rectangular format (using # shape). The output must be properly formatted. (Save your script as **tabularFormat.py** and **tabularFormat.txt**)

For example:

If the input is: “Hello World! Python rules!”

The output would be:

```
#####
# Hello  #
# World  #
# Python #
# rules  #
#####
```

If the input is: “Roses are red, sky is blue, I mind my own business, why can’t you!”

The output would be:

```
#####
# Roses  #
# are    #
# red    #
# skye   #
# is     #
# blue   #
# I      #
# mind   #
# my     #
# own    #
# business #
# why    #
# can't  #
# you    #
#####
```

Marks Distribution

Criteria	Comments	Logic	Execution	Total
Mark	2	3	3	8