Module Code
COM00015C

UNIVERSITY of York

BEng, BSc, MEng and MMath Degree Examinations 2020–2021

Department Computer Science

Title Software 1: Formative Assessment 1

Time Allowed 24 Hours (NOTE: Late papers will not be marked)

Time Recommended TWO hours

Word Limit Not Applicable

Allocation of Marks:

All questions are worth 25% each.

Instructions:

Candidates should answer **all** questions using Python 3. Failing to do so will result in a mark of 0%. All questions are independent and can be answered in any order. The solution submitted must work on the VDS, failing to do so will result in a mark of 0%.

Download the paper and the required source files from the VLE, in the "Assessment>Formative Assessments" section. Once downloaded, unzip the file. You **must** save all your code in the `ClosedExamination` folder provided. **Do not** save your code anywhere else other than this folder.

Submit your answers to the Department's Teaching Portal as a single **zip** file containing the `ClosedExamination` folder and its sub-directories. Failing to include the `ClosedExamination` folder will result in a loss of 10 marks.

If a question is unclear, answer the question as best you can, and note the assumptions you have made to allow you to proceed.

A Note on Academic Integrity

We are treating this online examination as a time-limited open assessment, and you are therefore permitted to refer to written and online materials to aid you in your answers.

However, you must ensure that the work you submit is entirely your own, and for the whole time the assessment is live you must not:

- communicate with departmental staff on the topic of the assessment
- communicate with other students on the topic of this assessment
- seek assistance with the assignment from the academic and/or disability support services, such as the Writing and Language Skills Centre, Maths Skills Centre and/or Disability Services. (The only exception to this will be for those students who have been recommended an exam support worker in a Student Support Plan. If this applies to you, you are advised to contact Disability Services as soon as possible to discuss the necessary arrangements.)
- seek advice or contribution from any third party, including proofreaders, online fora, friends, or family members.

We expect, and trust, that all our students will seek to maintain the integrity of the assessment, and of their award, through ensuring that these instructions are strictly followed. Failure to adhere to these requirements will be considered a breach of the Academic Misconduct regulations, where the offences of plagiarism, breach/cheating, collusion and commissioning are relevant: see AM1.2.1 *(Note this supercedes Section 7.3 of the Guide to Assessment)*.

1 (25 marks) Functions and Strings

All code for this question must be written in the file `question_1.py` provided.

Write a function `create_triangle(n)` that when given an integer n, returns a string representing a triangle n levels high when printed. To draw the triangle use the $-$ and x characters. If $n = 0$, the function should return an empty string, if n is not positive it should return `None`.

For examples:

```
>>> create_triangle(3)
'x--\nxx-\nxxx\n'
>>> print(create_triangle(3))
x--
xx-
xxx

>>> create_triangle(1)
'x\n'
>>> create_triangle(-1)
None
```

2 (25 marks) Functions and Strings

All code for this question must be written in the file `question_2.py` provided.

Write a function `create_chequerboard(n)` that when given an integer n, returns a string representing a chequerboard of size $n \times n$. To draw the chequerboard use the $-$ and x characters. If n is less than 2, the function should return `None`.

For example:

```
>>> create_chequerboard(4)
'x-x-\n-x-x\nx-x-\n-x-x\n'
>>> print(create_chequerboard(4))
x-x-
-x-x
x-x-
-x-x

>>> create_chequerboard(5)
'x-x-x\n-x-x-\nx-x-x\n-x-x-\nx-x-x\n'
```

3 (25 marks) Basic Programming Structure

All code for this question must be written in the file `question_3.py`.

Write a function `sum_integers(start, end)` that takes two positive integers, `start` and `end`, and returns the sum of all integers comprised between `start` and `end` inclusive. The method should return -1 if:

- `end < start`.
- `start < 0`.
- `end < 0`.

For example:

```
>>> sum_integers(5, 15)
110
>>> sum_integers(0, 4)
10
>>> sum_integers(6, 4)
-1
>>> sum_integers(-2, 4)
-1
```

4 (25 marks) Using Python strings and built-in list.

All code for this question must be written in the file `question_4.py`.

Write a function `scramble(word)` that takes a string parameter named `word`, and returns a string where the first two letters and the last two letters are the same as `word`, and the middle letters of the returned string are the remaining letters from `word` in a random order. A word of length smaller or equal to five letters is not changed by the function. It should be noted that calling the function several time with the same word may return a different value each time, as shown in the examples below. We assume that the string provided as parameter contains only letters from the English alphabet and no other symbols.

For example:

```
>>> scramble('hearing')
'heirang'
>>> scramble('hearing')
'heairng'
```

Hint: the package `random` and the function `shuffle` from this package might be useful.

End of examination paper