Semester 1 Examinations 2019/2020

Module	CA116 — Computer Programming I
Programmes	BSc in Computer Applications BSc in Data Science Study Abroad (ECSAX) Study Abroad (ECSAO)
Year of Study	1
Examiner	Dr Stephen Blott (ext. 5984 Dr Hitesh Tewari (external examiner for CA) Prof Mathieu d'Aquin (external examiner for DS)
Instructions	Answer *all four* questions. All questions carry equal marks.

The use of programmable or text storing calculators is expressly forbidden.
Please note that where a candidate answers more than the required number of questions, the examiner will mark all questions attempted and then select the highest scoring ones.

Requirements for this paper (please mark (X) as appropriate):

Log tables		*Thermodynamic tables*	
Graph paper		*Actuarial tables*	
Dictionaries		*MCQ only (do not publish)*	
Statistical tables		*Attached answer sheet*	

Additional Instructions

> Instructions:
>
> - Upload all answers to *Einstein*: here.
> - You may upload your answers as many times as you like. If you upload answers to *the same question multiple times*, then *only the last upload will be considered*.
> - For Python programming tasks, upload and test your solutions on *Einstein* as usual:
>   - If *Einstein* does not report your solution as correct, then attempt marks will be awarded subsequently as part of the marking process, as merited.
> - For other questions:

- You are told the name to use for your upload; this will be a **text file**, with a ".txt" extension.
- Create such a file using a **text editor** (such as *gedit* or *sublime*; do not use a word processor), and upload the resulting file to *Einstein*, as usual.
- *Einstein* simply accepts and stores such uploads; it does not correct your work. All such uploads will be assessed manually later as part of the marking process.

Only the work which you upload to *Einstein* will be considered. Any files you leave on the local desktop **will not** be collected. Any written work you do in exam scripts **will not** be assessed.

The Python shebang (for reference)

```
#!/usr/bin/env python
```

Question 1 25 Marks

The following five fragments of Python code each contain errors.

Correct the errors.

Where you are instructed to "*assume an existing variable ...*", *Einstein* will provide a value for that variable when you upload your solution.

1.1 5 Marks

Assume an **existing** list variable a.

Print out each element of the list, one element per line.

Name your Python fragment `exj-1.1.py`.

Initial fragment (contains errors)

```
#!/usr/bin/env python

i = 0
while i <= len(a):
   print a[i]
```

1.2 5 Marks

Assume **existing** integer variables `home_goals`, `home_points` `away_goals` and `away_points`.

These are the goals and points scored respectively by the home and away teams in a gaelic football match.

Print out the result of the match. The result is either "home win", "away win" or "draw".

Name your Python fragment `exj-1.2.py`.

Initial fragment (contains errors)

```
#!/usr/bin/env python

home_score = home_goals * (7 + home_points)
away_score = away_goals * (7 + away_points)

if home_score < away_score
   print "away win"
elif away_score < home_score
   print "home win"
else
   print "draw"
```

Tip

In gaelic football, one goal is worth seven points, so 1 goal and 2 points (for a total of 9) is better than 0 goals and 6 points (for a total of just 6).

1.3

5 Marks

The command-line arguments consist of exactly one argument, a file name.

Copy the contents of that file to standard output.

Name your Python script `exj-1.3.py`.

Initial script (contains errors)

```
#!/usr/bin/env python

import sys

file_name = sys.argv[1]

with open(file_name) as f:
   sys.write(f.read)
```

1.4

5 Marks

Assume an **existing** dictionary variable d.

Print out each key-value pair in the dictionary, one pair per line, separated by `-->`.

Name your Python fragment `exj-1.4.py`.

Initial fragment (contains errors)

```
#!/usr/bin/env python

for key in sorted(d):
   write key, "-->", d{key}
```

1.5

5 Marks

Assume an **existing** string variable s.

s consists of a sequence of whitespace-separated tokens.

Print the number of tokens contained in s.

Name your Python fragment `exj-1.5.py`.

Initial fragment (contains errors)

```
#!/usr/bin/env python

print len(s.strip(" "))
```

Question 2 25 Marks

2.1 Largest position 15 Marks

In a module (file) named `exj_21.py`, write a Python function named `largest_position` which accepts exactly one argument, a non-empty list, and returns the position of the largest element in that list.

5 of the available 15 marks will be awarded for having included suitable test cases in your upload.

Example

```
import exj_21

print exj_21.largest_position([7, 12, 20, 14, 20])    # 2
```

Note In the case of multiple candidate positions, choose the lowest-numbered position. See the case of 20 in the example, which first occurs at position 2.

2.2 Reverse 10 Marks

Assume an **existing** list a.

Write a Python fragment named `exj-2.2.py` which reverses the elements of a *in place*.

Your script should produce no output.

Note *Einstein* will provide values for a when you upload your solution.

Question 3 25 Marks

Standard input consists of a sequence of x, y coordinates each in the range 0 (inclusive) to 20 (exclusive), one per line. See the example, below.

Write a Python script named `exj-3.py` which plots those points on a two-dimensional graph. See the example.

Example standard input

```
0 0
2 10
19 16
19 17
19 18
19 19
```

Example standard output

```
 --------------------
|                   *|
|                   *|
|                   *|
|                   *|
|                    |
|                    |
|                    |
|                    |
|                    |
|  *                 |
|                    |
|                    |
|                    |
|                    |
|                    |
|                    |
|                    |
|                    |
|                    |
|*                   |
 --------------------
```

> **Details**
>
> The input points may be in any order.
>
> The first and last lines of the output consist of 21 characters: each a single space followed by 20 hyphens.
>
> All of the other lines consist 22 characters: a `|`, 20 spaces or asterisks (depending upon the input), and another `|`.

Note We have encountered this problem in labs. There, it was named "plot points".

Question 4 25 Marks

4.1 Index error 5 Marks

Write a Python script named `exj-4.1.py` which, when run, generates an index error (list index out of range).

4.2 Key error 5 Marks

Write a Python script named `exj-4.2.py` which, when run, generates a key error.

4.3 Sorting 15 Marks

For the remaining three question parts, write your answers in a single text file named `exj-4.3.txt` and upload them to *Einstein*.

4.3.1

Consider the following list: 5 Marks

```
a = [12, 6, 5, 8, 10, 3, 8, 2]
```

Give the state of the list after the first three iterations of the main/outer loop of the selection-sort algorithm.

4.3.2

Considering the same initial list again (see question 4.3.1, above), give the state of the list after the first three iterations of the main/outer loop of the insertion-sort algorithm. 5 Marks

4.3.3

Using big-oh notation, state the complexity of binary search. 5 Marks

Last updated 2019-12-17 12:07:10 GMT