BIRKBECK

(University of London)

MSc EXAMINATION FOR INTERNAL STUDENTS

MSc Computer Science

MSc Data Science

Department of Computer Science and Information Systems

Principles of Programming I

BUCI033S7

DATE OF EXAMINATION: Monday, 30th April 2018
DURATION OF PAPER: One Hour

Written — Mock Paper — Evening

WITH OUTLINE SOLUTIONS

RUBRIC:

1. Candidates should attempt ALL 9 questions on this paper.
2. You are advised to look through the entire examination paper before getting started, in order to plan your strategy.
3. Simplicity and clarity of expression in your answers is important.
4. All programming questions should be answered using the Python programming language.
5. Electronic calculators are **NOT** allowed.
6. Start each question on a new page.

Question:	1	2	3	4	5	6	7	8	9	Total
Marks:	10	2	4	5	6	3	7	6	7	50

Question 1 ..... Total: 10 marks

Given the following function:

(a) 4 marks

```
def letters(s):
    ss = []
    for ch in s:
        if ch.isalpha():
            ss.append(ch.lower())
    return ss
```

what is returned by the call `letters("1 and 2 and 3")` ?

Solution:

```
['a', 'n', 'd', 'a', 'n', 'd']
```

Skips the numbers and returns the characters as a list.

(b) Given the following function: 1 mark

```
def f(n):
    return [foo, points, letters, lets][n]
```

what is returned by the call `f(1)` ?

Solution:

```
First item in the list.

NameError: name 'foo' is not defined
```

(c) Given the function 5 marks

```
def fiddle(index, lst):
    index += 1
    lst[index] += 1
```

what will be printed by

```
n, nums = 3, range(1, 6)
fiddle(n, nums)
print(n, nums)
```

Solution:

```
3 range(1, 6)
```

Question 2 ..... Total: 2 marks

What is the purpose of the following code in a Python program?

```
if __name__ == '__main__':
   main()
```

Solution: To only run code when this file is the one that was invoked by the Python interpreter.

Question 3 ..... Total: 4 marks

Complete the code for the following function so it matches its documentation:

```
def doubleList(numberList):
    ''' For each of the numbers in the list numberList, print a line
    containing twice the original number.  For example,
    doubleList([3, 1, 5]) would print
    6
    2
    10
    '''
```

Solution:

```
def doubleList(numberList):
        ''' skip repeating docs... '''
        for n in numberList:
                print(2*n)
```

Question 4 ..... Total: 5 marks

Assume that you have written a class `ListUtils` to provide a number of operations on lists. This class is in the file `listutils.py`. In the following three questions you will write a complete class to test one of the functions in `ListUtils`.

(a) Write the necessary import statements and the class header for a class named `ListUtilsTest`. [2 marks]

Solution:

```
from otherfile import ListUtilsTest
```

(b) Write a test, using `pytest`, for the function `duplicate(lst)` which is supposed to return a shallow copy of the list given as a parameter. [3 marks]

Solution:

Question 5 ..... Total: 6 marks

The DRY (Don't Repeat Yourself) principle applies to both code and data.

(a) Why is it a good idea not to copy and paste code, or otherwise duplicate code? That is, what problems result from having duplicate code? [3 marks]

Solution: Any corrections or modifications have to be done in multiple places, and you might miss some.

(b) Why is it a good idea not to have the same information represented in more than one way? That is, what problems arise? 3 marks

Solution: Unless considerable care is taken, the representations may get out of sync.

Question 6 ..... Total: 3 marks

Suppose you have a `class Robot(object)`, and you want to create a class to represent a dog robot. What is the difference between saying `class Dogbot(object)` and saying `class Dogbot(Robot)` ?

Solution: Only one inherits from `Robot`, `class Dogbot(Robot)`.

Question 7 ..... Total: 7 marks

Compare and contrast *recursion* and *iteration*. When would you prefer one other the other and why? Provide appropriate examples to illustrate your answer.

Solution: In recursion, function call itself until the base condition is reached.

Iteration means repetition of process until the condition fails.

You might prefer iteration for efficiency, dependent on the language.

Recursion is easier to scale to concurrency, and maybe easier to represent the problem.

Any reasonable examples.

Question 8 ..... Total: 6 marks

Provide an English description of what the following function does:

```
import random
def points(n):
    directory = {}
    for i in range(0, n):
        pname = chr(ord('a') + i)
        x = 1000.0 * random.random()
        y = 1000.0 * random.random()
        directory[pname] = (x, y)
    return directory
```

Solution: Returns a dictionary with the characters, from `a` onwards, which maps to a pair of floating point numbers in the range 0 to 1000.

Question 9 ..... Total: 7 marks

Suppose you are defining a class `Circle`, and every object of this class must have three values: The `x` and the `y` coordinates of the circle's centre, and the radius of the circle.

(a) Write the constructor that you would put in this class. 2 marks

(b) Use the constructor you have written to create a circle named `unitCircle` with `radius` = 1 and centre at the origin (x = y = 0). 5 marks

Solution:

```
class Circle:
    def __init__(self, x, y, radius):
        self.x = x
        self.y = y
        self.radius = radius

circ = Circle(0,0,1)
```