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

WITH OUTLINE SOLUTIONS

RUBRIC:

1. Candidates should attempt ALL 6 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	Total
Marks:	4	4	12	18	5	7	50

outline solutions

 ©Birkbeck 2018

Question 1 ..... Total: 4 marks

(a) Write a Python program that prompts the user for two numbers, reads them in, and prints out the product. [2 marks]

Solution:

```
x = int(input("Enter a number: "))        # or some such prompt
y = int(input("Enter another number: ")) # or some such prompt
print("The product is ", x*y)              # or some such label
```

(b) Given a string `s`, write a short expression for a string that includes s repeated five times. [2 marks]

Solution:

```
 s*5    # or:   s+s+s+s+s
```

Question 2 ..... 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 3 ..... Total: 12 marks

For each of the following, write the fewest possible Python statements to accomplish the required task. (Half credit if your answer is correct, but longer than necessary.)

(a) Use a for loop to print the contents of list variable list, in order, one value per line.

Solution:

```
for i in list:
    print(i)
```

Outline solutions

or

```
for i in range(0, len(list)):
    print(list[i])
```

(b) Use a while loop to print the contents of list variable list, in order, one value per line.

Solution:

```
i = 0
while i < len(list):
    print(list[i])
    i += 1
```

(c) Use a for loop to print the contents of list variable list, in reverse order, one value per line.

Solution:

```
for i in range(1, len(list) + 1):
    print(list[-i])
```

(d) Print one of the words negative, zero, or positive, according to whether variable `x` is less than zero, zero, or greater than zero, respectively.

Solution:

```
if x < 0: print("negative")
elif x == 0: print("zero")
else: print("positive")
```

(e) Create a file named `foo.txt`, and write all the values in list variable words to it, one value per line.

Solution:

```
f = open('foo.txt', 'w')
for word in words:
    f.write(word + '\n')
f.close()
```

(f) Write a function named `isEven` that, given a single integer parameter, returns `True` if the parameter is an even number, `False` otherwise.

Solution:

```
def isEven(n):
    return n % 2 == 0
```

or

```
isEven = lambda n: n % 2 == 0
```

Outline solutions

(g) Write a unit test method that says calling `evenRand()` should return an even number.

Solution:

```
def testEvenRand():
    assert evenRand() % 2 == 0
```

(h) Create, and save in a variable, a 10x10 array (list of lists), all of whose values are `None`.

Solution:

```
a = [[[None] * 10] * 10]
```

or

```
a = [ ]
for i in range(0, 10):
    a.append(10 * [None])
```

or

```
a = [0] * 10
for i in range(0, 10):
    a.append(10 * [None])
```

Question 4 ..... Total: 18 marks

Provide a short definition, and example, for each of the following:

(a) recursion — 3 marks

Solution: The act of a function [method] calling itself.

(b) side effect — 3 marks

Solution: A change that occurs in addition to the obvious effect. For functions [methods], this refers to any effect the function has beyond returning a value.

(c) dynamic typing — 3 marks

Solution: Where the type of the variable is determined at runtime and can change.

(d) magic number — 3 marks

Solution: A number appearing in code with no obvious interpretation.

(e) encapsulation — 3 marks

Solution: information hiding; not using the internal variables and methods of a class [or module] from outside that class

Outline Solutions

(f) refactoring [3 marks]

Solution: Rewriting code without changing its functionality, usually to make it more readable or more generally useful.

Question 5 ..... Total: 5 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

print(points(30))
```

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 6 ..... 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)
```

Online solutions