

Model Answers for Module BUCI007H4

Summer

Birkbeck
(University of London)

BSc/FD EXAMINATION

Department of Computer Science and Information Systems

Introduction to Programming (BUCI007H4)

CREDIT VALUE: 15 credits

Summer 2019

Date of examination: XXXDAY ? MAY 2019

Duration of paper: 14:30 – 16:30

The programming language is PYTHON.

There are **ten** questions in this paper.

Answer all **ten** questions.

Each question carries **10** marks in total.

Calculators and other electronic devices are not permitted.

The examination is closed book.

No supplementary material is provided.

This paper is not prior disclosed.

The quote marks for strings are in this style "string" or 'string'.

1. (a) Which of the following would result in an error in Python and which wouldn't?
- A) `print("Once upon a time...", she said.)`
 - B) `print("He said, 'Yes!' ")`
 - C) `print('3\')`
 - D) `print('That's okay.')`
 - E) `print("string"*3)`
- (b) Which of the following are accepted by the compiler as names of variables?
- A) `3times`
 - B) `in`
 - C) `__dollar__` (Two underscores, then dollar, followed by two underscores)
 - D) `_` (One underscore)
 - E) `true`

Answer:

- (a) A), C), D), E) error; B) correct
- (b) A), B) not accepted; C) D) E) accepted

1 mark each.

2. Consider the following code.

(10 marks)

```
a = 2
b = 7
c = -3
d = 4
e = 4
a = b - a * a
b = b // 3 + 2 * c
c = round(c ** 3 / 7, 2)
d = 10 % d ** 3 ** 2
e = abs(-14 + 34 / e)
```

What are the values of a, b, c, d and e after the above code is executed?

Answer:

- i) a = 3
- ii) b = -4
- iii) c = -3.86
- iv) d = 10
- v) e = 5.5

2 marks each

3. (a) Explain what compile-time error, run-time exception and run-time error (but not an exception) are. Give an example for each category. (6 marks)
- (b) Write a function `compare()` that takes in two arguments. It returns 1 if the result of multiplying the first argument by 10 and then subtracting 38 from the result is strictly less than half of the second argument; and returns 0 otherwise. (4 marks)

Answer:

- (a) *i)* Compile-time error: an error in the syntax (grammar of Python) is detected by the compiler, e.g. `print(Hello World!)`.
- ii)* Run-time exception: the program is compiled but the run time process stops when the error is encountered, e.g. `print(1/0)`.
- iii)* Run-time error (but not an exception): the program runs but does not produce the desired result, e.g. `print("Helo World!")`.

1 mark for each explanation, 1 mark for each example.

- (b) `def compare(x, y):`
 `if 10 * x - 38 < y / 2:`
 `return 1`
 `else:`
 `return 0`

1 mark for correctly defining two arguments, 1 mark for the correct form of writing a function, 1 mark for the correct form of if/else structure, 1 mark for the correct comparison.

4. (a) Write Python code to obtain the `sqrt` and `cos` functions from the `math` module. (1 mark)

(b) Write the following mathematical expression in Python (2 marks)

$$d = \sqrt{P^5 - (1 + \cos(F) \div 100)}$$

(c) Write Python code to prompt the user with the string:

"Please enter the price of a loaf of bread:"

The input will represent the price of a loaf of bread and be assigned to the variable `priceLoafBread`. Note that the value of `priceLoafBread` should be of type float. If the number entered is strictly negative then set `priceLoafBread` to be 0 and include an error message. (2 marks)

(d) What is the value of `priceLoafBread` in your code if the user enters the following? (2 marks)

- i) Please enter the price of a loaf of bread: 5
- ii) Please enter the price of a loaf of bread: -4

(e) What is printed when the following code is run? (3 marks)

- i) `print("MALET"[-2] * 4 + 3 * "street"[2])`
- ii) `print(len("\come\n on!\n\n"))`
- iii) `bbk = ["Birkbeck", "University", "London", "England"]`
`print(bbk[-3][5])`

Answer:

- (a) `from math import sqrt, cos`
- (b) `d = sqrt(P**5 - (1+cos(F)/100))`
- (c) `priceLoafBread = float(input("Please enter the price of a loaf of bread:"))`
`if priceLoafBread < 0 :`
`priceLoafBread = 0`
`print("Error! The price cannot be negative.")`
- (d) i) 5.0
ii) 0
- (e) i) "EEEErrr"
ii) 12
iii) "r"

5. (a) `percentage_w` is a variable for the percentage of women in a population. Write down a reasonable `print` statement so that the output is of the following form:

The percentage of women: `~~percentage_w%`.

Note that `~` represents one space. There should be at least two spaces between "The percentage of women:" and `percentage_w%`. Round the value of `percentage_w%` to two decimal places.

For instance, if `percentage_w = 54.8765`, then the output will be

The percentage of women: `~~54.88%`.

(6 marks)

- (b) The following is displayed by one or more calls to the `print` function.

tom

dick

harry

Write down two different ways to make one or more function calls to `print`, in order to produce this output (4 marks)

Answer:

- (a)

```
percentage_w = 54.8765
```

```
print("The percentage of women: %7.2f%%" % percentage_w)
```

Correct "7.2f" 2 marks, correct %% 1 mark, correct application symbol 1 mark, correct in general 2 marks.

- (b)

```
print('tom\ndick\nharry')
```

or

```
print("tom")
```

```
print("dick")
```

```
print("harry")
```

2 marks each

6. (a) Give short English language descriptions of the relational operators \geq and $<$. (2 marks)
- (b) Single characters are ordered such that the upper case letters come before lower case letters, numbers come before letters and the space character comes before all non-empty characters. Explain how this ordering for single characters can be extended to non-empty strings that contain letters or numbers or the space character. (4 marks)
- (c) Evaluate the following expressions
- i) "prefix" $<$ "postfix"
 - ii) "3pq" $<$ "3B47"
 - iii) "car nation" $<$ "carnation"
 - iv) "a43" $<$ "a49"
- (4 marks)

Answer:

- (a) \geq is greater than or equal (one mark). $<$ is strictly less than (one mark).
- (b) Let s_1 and s_2 be two non-empty strings. Assume $s_1 \neq s_2$ and $\text{length}(s_1) \leq \text{length}(s_2)$. If s_1 is a prefix of s_2 , then s_1 precedes s_2 . If s_1 is not a prefix of s_2 , let i be the first index, $i \geq 0$ for which s_1 and s_2 differ. If $s_1[i] < s_2[i]$, then s_1 comes before s_2 otherwise s_1 comes after s_2 . Four marks for any reasonable account.
- (c) False, False, True, True. One mark each.

7. (a) The function call `grade(m)` returns "A" if the mark `m` is in the range 70 to 100, "B" if `m` is in the range 60 to 69, "C" if `m` is in the range 50 to 59 and "D" if `m` is in the range 0 to 49. Implement the function `grade` using a nested set of if statements. Use the `if ... elif ... else...` construction. (6 marks)
- (b) Implement a second function `grade1` that requests an integer `m` from the keyboard with the prompt "Please input an integer in the range 0 to 100". If an integer in the correct range is typed in then `grade1` returns the same string value as the function `grade`. If an integer outside the correct range is typed in then `grade1` prints the error message "Integer out of range" and returns the empty string. The function `grade1` has no arguments. Suggestion: use a function call to `grade` from `grade1`, if appropriate. (4 marks)

Answer:

```
(a) def grade(m) :  
    if m ≥ 70 :  
        return "A"  
    elif m ≥ 60 :  
        return "B"  
    elif m ≥ 50 :  
        return "C"  
    else :  
        return "D"
```

Six marks. One mark deducted for each error.

```
(b) def grade1() :  
    m = int(input("Please input an integer in the range 0 to 100"))  
    if m > 100 or m < 0 :  
        print("Integer out of range")  
        return ""  
    else :  
        return grade(m)
```

Four marks. One mark deducted for each error.

8. (a) Describe the action of the following code in a correct Python program.

```
i = 0
while i ≤ 0 :
    i = int(input("Input a strictly positive integer"))
print(i)
```

(4 marks)

- (b) Revise the code in part (a) of this question such that the revised code allows no more than three attempts to input a correct value of i. The revised code prints the correct value of i if there is one, otherwise it prints "No further tries allowed". (6 marks)

Answer:

- (a) The variable i is initialised with the value 0. The while loop is entered. The prompt "Input a strictly positive integer" is printed and a value of i is input. If this value is strictly greater than 0 then the while loop is exited and the value of i is printed. Otherwise, a new value of i is read in. The while loop runs repeatedly until a value of i strictly greater than 0 is entered. Four marks.

- (b)
- ```
i = 0
c = 0
while i ≤ 0 and c < 3:
 i = int(input("Input a strictly positive integer"))
 c = c+1
if i > 0 :
 print(i)
else:
 print("No further tries allowed")
```

Six marks. One mark deducted for each error.

9. (a) State one advantage of using functions in programming. (2 marks)
- (b) What is a local variable in a function? (2 marks)
- (c) The function calculation is defined as follows.

```
def calculation(x, y) :
 v = x+y*y
 if v < 10 :
 return v
 else :
 return x
```

Identify the header and the body of the function calculation. (2 marks)

- (d) What is printed when the following code is executed after executing the code defining the function calculation in part (c) of this question. Justify your answer.

```
v = 10
print(calculation(1, 2))
```

(4 marks)

Answer:

- (a) To simplify the task of programming, to hide complexity, to ensure that repeated calculations are carried out by the same lines of code. Any reasonable answer accepted (two marks).
- (b) A local variable is one that is defined in the body of the function (two marks).
- (c) The header is the first line def calculation(x, y) (one mark). The body consists of the remaining lines (one mark).
- (d) 5 (two marks). The variable v defined and initialised outside the function call is distinct from the local variable v defined within the function call (two marks).

10. (a) The list `values` is defined by `values = [32, 54, 67, 1]`. Show how an index can be used to access the entry 54 in `values`. (1 mark)
- (b) The variable `values` is as given in part (a) of this question. Which of the following statements can be executed correctly and which cannot be executed without error.
- i)* `print(values[3])`
  - ii)* `print(values)`
  - iii)* `v = values[4]`
  - iv)* `v = values[values[-1]]`
- (4 marks)
- (c) What is the value of `n` after the following code is executed in a correct Python program. Justify your answer.
- ```
friends = ["Harry", "Emily", "Emily"]
n = friends.index("Emily")
```
- (2 marks)
- (d) Write code to count the number of times the string "Emily" appears in a list `ls` of strings. It is not necessary to include the code in a function. (3 marks)

Answer:

- (a) The index `i = 1` or the index `-3` is used to extract the value 54, i.e. `values[1]` or `values[-3]`. Either answer accepted (one mark).
- (b) correct, correct, error, correct (one mark each).
- (c) `n = 1` (one mark). The first occurrence of "Emily" is at the position with index 1. The occurrence of "Emily" at the position with index 2 is ignored (one mark).
- (d)
- ```
n = 0
for x in ls :
 if x == "Emily" :
 n = n+1
```

Three marks. Any reasonable version accepted. One mark deducted for each error.