

CMPT 120—QUIZ 1 (Feb. 5)

This quiz contains six exercises, each taking roughly 10 minutes. You may pick five out of the six. The grading of each exercise is not fixed. However there are bonus points for each exercise that you complete entirely successfully with no mistakes (so in total, you may get up to six bonus points on this quiz).

DO NOT FORGET TO PUT YOUR SFU ID AT THE TOP LEFT OF EVERY PAGE.

EXERCISE 1: point out the mistakes (10 minutes)

The program in Listing 1 is supposed to print out the result of 10! (which is the product of 1, 2, 3, ... up to 10 included), but there are certain mistakes. You must try to identify them with the following statements.

```

1. acc = 0
2. for x in range(0, 10)
3.     acc = acc * x
4.     print acc

```

Listing 1: printing the factorial of 10 = 1 * 2 * 3 * ... * 10

Circle T if you believe the statement to be correct and F if you believe the statement to be incorrect. *Before you start responding, you are encouraged to try executing the algorithm on your own, to see if it works.*

- | | | |
|---|----------|----------|
| 1. A colon is missing from line 2, containing the <code>for</code> statement. | T | F |
| 2. The variable in the <code>for</code> loop should be <code>i</code> instead of <code>x</code> . | T | F |
| 3. The variable named <code>acc</code> should actually be called <code>result</code> instead of <code>acc</code> , because it will contain the result of the program's computation. | T | F |
| 4. The range is incorrect because it includes 0: when we multiply <code>acc</code> by 0 it will erase whatever it contains and <code>acc</code> will always be equal to 0. The range should start in 1. | T | F |
| 5. The range is incorrect: it should end in 11 instead of 10. | T | F |
| 6. The print statement in line 4 is incorrectly indented. | T | F |
| 7. The print statement in line 4 should be a return statement instead because we are defining a function in this program. | T | F |
| 8. The variable <code>acc</code> should be initialized to 1. | T | F |

EXERCISE 2: execute a program (10 minutes)

```
1. def myfunc(first, last):
2.     acc = 0
3.     for i in range(first, last+1):
4.         acc = acc + i
5.     for j in range(first, last+1):
6.         acc = acc * 2
7.     return acc
8.
9. mynum1 = myfunc(1, 3) + myfunc(3, 4)
10. mynum2 = myfunc(1, 4)
11.
12. if mynum1 < mynum2:
13.     print "First wins"
14. else:
15.     print "Second wins"
```

Listing 2: complicated function and program

At the end of the program:

- What is the value of mynum1?
- What is the value of mynum2?
- What does the program print out?

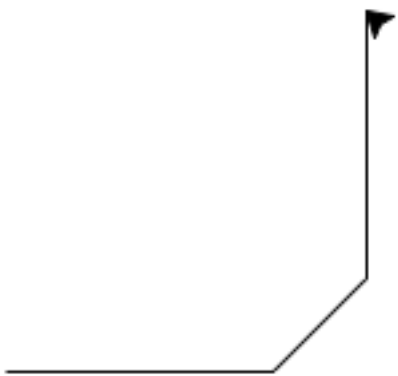
EXERCISE 3: output in a Python shell (10 minutes)

Assume that the following instructions are typed in a newly opened Python shell, and in the order presented here, one statement or expression after the other. For each statement or expression show the output *if there is any*; and if there is an error is caused, explain briefly why.

Statement	Output and/or explanation
>>> len("var")	
>>> len(var)	
>>> x = "25"	
>>> type(x)	
>>> x = x*3	
>>> x	
>>> x.isdigit()	
>>> y = int(x)	
>>> x + 10	
>>> y + 10	
>>> s = "hello"	
>>> s.upper()	
>>> int("2") * "3"	
>>> 10 % 2	
>>> 11 % 2	
>>> 1/3	
>>> 1/3.	
>>> int("1")/float("3")	

EXERCISE 4: turtle code (10 minutes)

The following program has been written in Python using the Python module, but the execution of the program has accidentally been stopped in the middle.



```

1. import turtle
2.
3. turtle.reset()
4. turtle.speed("fastest")
5.
6. turtle.forward(100)
7. turtle.left(45)
8. turtle.forward(50)
9. turtle.left(45)
10. turtle.forward(100)
11. turtle.left(45)
12. turtle.forward(50)
13. turtle.left(45)
14. turtle.forward(100)
15. turtle.left(45)
16. turtle.forward(50)
17. turtle.left(45)
18. turtle.forward(100)
19. turtle.left(45)
20. turtle.forward(50)
21. turtle.left(45)

```

Listing 3: turtle code

1. The program did not finish. Which was the line that was executed (consider carefully the position of the turtle)?
2. Continue the drawing as the program would have done it, if possible keeping to scale!
3. The program is very redundant, because it repeats a lot of very similar code. Would you have a suggestion on how to improve the program, and make it shorter (while doing the same thing)? You can write code, or you can simply explain what construct you would use.

.....

.....

.....

.....

.....

EXERCISE 5: what does this program do? (10 minutes)

```

1. def myfunction(string):
2.     total = 0
3.     cut = 0
4.
5.     for char in string:
6.         if char.isalpha():
7.             cut = 0
8.         else:
9.             if cut == 0:
10.                cut = 1
11.                total = total + 1
12.
13.     return total

```

Listing 4

Try finding out what the following function calls do:

- `myfunction("hello---you get a haircut?")` returns
- `myfunction("I...did...not...see...you")` returns

From these examples, what do you think the function does?

.....

EXERCISE 6: write your own program (10 minutes)

Write a function `largest_divided_by_ten_in_range(first, last)`, that takes two parameters `first` and `last`, and returns the largest number that can be divided by 10 without a remainder (that is, the largest number with the units equal to 0).

For instance:

- `largest_divided_by_ten_in_range(4, 15)` returns 10.
- `largest_divided_by_ten_in_range(56, 1509)` returns 1500.
- `largest_divided_by_ten_in_range(-34, -9)` returns -10.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....