SFU ID: ....

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 `for` statement.	**[T]**	**F**
2.	The variable in the `for` loop should be `i` instead of `x`.	**T**	**[F]**
3.	The variable named `acc` should actually be called `result` instead of `acc`, 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 `acc` by 0 it will erase whatever it contains and `acc` 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 `acc` should be initialized to 1.	**[T]**	**F**

EXERCISE 2: execute a program (10 minutes)

```
def myfunc(first, last):
  acc = 0
  for i in range(first, last+1):
    acc = acc + i
  for j in range(first, last+1):
    acc = acc * 2
  return acc

mynum1 = myfunc(1, 3) + myfunc(3, 4)
mynum2 = myfunc(1, 4)

if mynum1 < mynum2:
  print "First wins"
else:
  print "Second wins"
```

Listing 2: complicated function and program

At the end of the program:

- What is the value of `mynum1`? **76**
- What is the value of `mynum2`? **160**
- What does the program print out? **First wins**

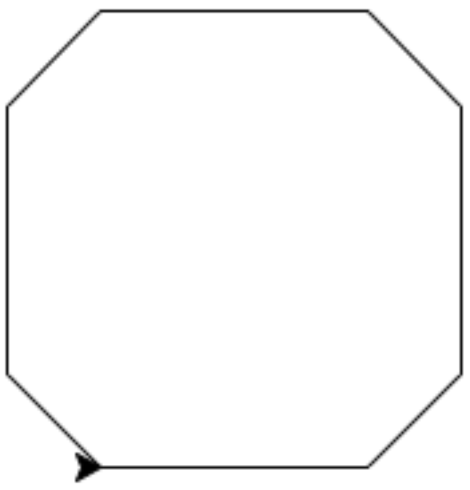
EXERCISE 3: output in a Python shell (10 minutes)

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")`	**3 (size of the string "var")**
`>>> len(var)`	**Error (variable "var" is undefined)**
`>>> x = "25"`	**Nothing (assignment returns no value)**
`>>> type(x)`	**str**
`>>> x = x*3`	**Nothing**
`>>> x`	**'252525'**
`>>> x.isdigit()`	**True**
`>>> y = int(x)`	**Nothing**
`>>> x + 10`	**Error (can't add a string and an integer)**
`>>> y + 10`	**252535**
`>>> s = "hello"`	**Nothing**
`>>> s.upper()`	**'HELLO'**
`>>> int("2") * "3"`	**'33'**
`>>> 10 % 2`	**0**
`>>> 11 % 2`	**1**
`>>> 1/3`	**0**
`>>> 1/3.`	**0.333333333...**
`>>> int("1")/float("3")`	**0.333333333...**

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 [last] executed (consider carefully the position of the turtle)? **Line 11**

2. Continue the drawing [...]

3. The program is very redundant, because it repeats a lot of very [...]

   Replace lines 6 to 21 by a loop since we are repeating the same lines of code 4 times in a row

```
for i in range(4):
  turtle.forward(100)
  turtle.left(45)
  turtle.forward(50)
  turtle.left(45)
```

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

```
def myfunction(string):
    total = 0
    cut = 0

    for char in string:
      if not char.isalpha():
        cut = 0
      else:
        if cut == 0:
          cut = 1
          total = total + 1

    return total
```

Listing 4

Try finding out what the following function calls do:

- `myfunction("hello---you get a haircut?")` returns **5**
- `myfunction("I...did...not...see...you")` returns **4**

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

`myfunction(string)` returns the number of blocks of contiguous non alphabetical characters (or it returns the number of words in a sentence, or the number of words minus one)

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.

```
def largest_divided_by_ten_in_range(first, last):
  last = -1
  for num in range(first, last+1):
    if num % 10 == 0:
      last = num
  return num
```