



CSI2100-01 Spring 2020

Lab 13_final

Due to the Covid19 precautions, this lab (Lab 13_final) is taken as a replacement for the final exam. The score that you will achieve for this lab will constitute your final exam score.

Because of its exam character, Lab 13_final differs from our regular labs in the following ways:

- To give equal opportunity to all course participants, no individual counseling is provided for Lab 13_final.
- In case of questions, please ask on the Q&A board, for everybody to participate and benefit.

This lab consists of questions to check your understanding of concepts discussed in the course. Please submit the answers to these questions as a single textfile (similar to what you did with previous labs, for example, Lab 3). The required file name of your text file is stated in Section 2.

You may include code snippets with your answers, if it benefits your explanation. Unless specifically asked, code snippets are not mandatory.

1 Marking Criteria and Due Date

- **Due date: Saturday, June 27, 2020, 23:59.**

This is a *firm* due date. Please submit your answers by this due date on YSCEC.

- This is an individual assignment. Submissions are checked for plagiarism. Once detected, measures will be taken for **all** students involved in the plagiarism incident (including the “source” of the plagiarized code).

2 Deliverables

- For this lab, you only need to submit a single text file that contains your answers.
- Please use the name README_lab13_final_<student_id>.txt for this file.

3 Questions

Question 1 (15 pts):

- (a) What is the purpose of an *escape sequence*?
- (b) Please consider the below Python 3 program. Assuming that the file operations succeed (no exception raised), what will be the size of the written file? **Justify your answer.**

```
1 f = open('myfile.txt', 'w')
2 f.write('\N{HANGUL SYLLABLE TAEG}')
3 f.close()
```

- (c) UTF-8 character codes are between 1 and 4 bytes long. Why do they vary in their length?

Question 2 (10 pts):

Please consider the below Python 3 program.

```
1 x = [1, 2, 3, 4, 5]
2
3 for i in range(len(x)):
4     if x[i] % 2 == 0:
5         del x[i]
```

- (a) This program may not work as expected. What is the reason for this problem?
- (b) How can this program be fixed? In your answer, you are not allowed to make any assumption on the contents of the list from Line 1 except that the list contains integers only.

Question 3 (10 pts):

In Python, what is the difference between a *function call* and a *method call*?

What is the purpose of the `self` parameter of Python methods?

Question 4 (10 pts):

Please consider the below program.

```
1 def check(x):
2     if x < 0:
3         raise ValueError('Too small!')
4     elif x == 0:
5         raise ValueError('Zero not allowed!')
6     elif x > 150:
7         raise OverflowError('Too large!')
8
9
10 def getInput():
11     try:
12         val = int(input('Enter an integer: '))
13         check(val)
14     except ValueError as msg:
15         val = 1
16     except OverflowError as msg:
17         raise msg
18     return val
19
20
21 try:
22     val = getInput()
23 except ValueError:
24     pass
25 except OverflowError:
26     pass
```

- (a) Assume that the user will input 5 (followed by “enter”) in Line 12. For each of the exceptions raised (if any), state the line number where the exception is raised, and where the exception is handled. Write “none” if no exception occurs.
- (b) Assume that the user will input -1 (followed by “enter”) in Line 12. For each of the exceptions raised (if any), state the line number where the exception is raised, and where the exception is handled. Write “none” if no exception occurs.
- (c) Assume that the user will input 1000 (followed by “enter”) in Line 12. For each of the exceptions raised (if any), state the line number where the exception is raised, and where the exception is handled. Write “none” if no exception occurs.
- (d) Assume that the user will input foo (followed by “enter”) in Line 12. For each of the exceptions raised (if any), state the line number where the exception is raised, and where the exception is handled. Write “none” if no exception occurs.

Question 5 (10 pts):

What is the purpose of the below code? To answer this question, please write down the number of each of the below statements that you regard to be true.

```
from math import factorial as simpFact
```

- (S1) Make every function from the math module available in a client program, but the function factorial will be renamed to simpFact.
- (S2) Import only the factorial function from the math module into client, which must be fully qualified when called.
- (S3) Import only the factorial function from the math module and integrate it into the namespace of the client as simpFact instead of factorial.
- (S4) This is not proper Python code.

Question 6 (20 pts):

In the lecture, we discussed the three main concepts of object orientation: encapsulation, inheritance and polymorphism.

For each concept, please (1) explain its purpose, and (2) provide a code snippet that shows how this concept can be applied in practice.

Question 7 (15 pts):

In the lecture, we discussed Python's `global` keyword. Assume that in the next version of Python this `global` keyword will be eliminated without any replacement mechanism.

- (a) How would you have to modify your programs to work without the `global` keyword?
- (b) Are there any applications where you would expect to encounter problems without the `global` keyword? **Justify your answer.**

Question 8 (10 pts):

In our discussion of *recursion*, we observed in the PythonTutor that Python uses a new *frame* for each recursive function call. Why is this the case? **Justify your answer.**