

COL100: Lab 8

Exceptions

March 5, 2017

Part 1: Questions

1. List five common examples of exceptions. Write a Python program to detect the following exceptions: Memory exhaustion, array index out of bounds, arithmetic overflow, division by zero, invalid method parameters.
2. What is the difference between assert and exceptions?
3. If no exceptions are thrown in a try block, where does control proceed to when the try block completes execution?
ANS: The catch blocks for that try statement are skipped, and the program resumes execution after the last catch block. If there is a finally block, it's executed first; then the program resumes execution after the finally block.

Part 2: Practice Questions (MCQs)

1. How many except statements can a try-except block have?
 - (a) zero
 - (b) one
 - (c) more than one
 - (d) more than zero

Answer: d

Explanation: There has to be at least one except statement.

2. When will the else part of try-except-else be executed?
 - (a) always
 - (b) when an exception occurs
 - (c) when no exception occurs
 - (d) when an exception occurs in to except block

Answer: c

Explanation: The else part is executed when no exception occurs.

3. Is the following code valid?

```

1 try:
2     # Do something
3 except:
4     # Do something
5 finally:
6     # Do something

```

- (a) no, there is no such thing as finally
- (b) no, finally cannot be used with except
- (c) no, finally must come before except
- (d) yes

Answer: b

Explanation: Refer documentation.

4. Is the following code valid?

```

1 try:
2     # Do something
3 except:
4     # Do something
5 else:
6     # Do something

```

- (a) no, there is no such thing as else
- (b) no, else cannot be used with except
- (c) no, else must come before except
- (d) yes

Answer: d

Explanation: Refer documentation.

5. Can one block of except statements handle multiple exception?

- (a) yes, like except TypeError, SyntaxError [...]
- (b) yes, like except [TypeError, SyntaxError]
- (c) no
- (d) none of the mentioned

Answer: a

Explanation: Each type of exception can be specified directly. There is no need to put it in a list.

6. When is the finally block executed?

- (a) when there is no exception
- (b) when there is an exception
- (c) only if some condition that has been specified is satisfied
- (d) always

Answer: d

Explanation: The finally block is always executed.

7. What is the output of the following code?

```
1 def foo():
2     try:
3         return 1
4     finally:
5         return 2
6 k = foo()
7 print(k)
```

- (a) 1
- (b) 2
- (c) 3
- (d) error, there is more than one return statement in a single try-finally block

Answer: b

Explanation: The finally block is executed even there is a return statement in the try block.

8. What is the output of the following code?

```
1 def foo():
2     try:
3         print(1)
4     finally:
5         print(2)
6 foo()
```

- (a) 1 2
- (b) 1
- (c) 2
- (d) none of the mentioned

Answer: a

Explanation: No error occurs in the try block so 1 is printed. Then the finally block is executed and 2 is printed.

9. What is the output of the following?

```
1 try:
2     if '1' != 1:
3         raise "someError"
4     else:
5         print("someError has not occurred")
6 except "someError":
7     print("someError has occurred")
```

- (a) someError has occurred
- (b) someError has not occurred
- (c) invalid code
- (d) none of the mentioned

Answer: c

Explanation: A new exception class must inherit from a BaseException. There is no such inheritance here.

10. What happens when `'1' == 1` is executed?

- (a) we get a True
- (b) we get a False
- (c) an `TypeError` occurs
- (d) a `ValueError` occurs

Answer: b

Explanation: It simply evaluates to False and does not raise any exception. (ASCII value of '1' is compared to 1)