# Exception Handling Questions and Answers + MCQ (Multiple Choice Questions)

**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 Python code valid?**

```
try:
    # Do something
except:
    # Do something
finally:
    # 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: d
Explanation: Refer documentation.

**4. Is the following Python code valid?**

```
try:
    # Do something
except:
    # Do something
else:
    # 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: d
Explanation: Each type of exception can be specified directly. There is a need to put all exceptions are in side parenthesis like except (TypeError , SyntaxError, ValueError)

**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 will be the output of the following Python code?**

```
def foo():
    try:
        return 1
    finally:
        return 2
k = foo()
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 will be the output of the following Python code?**

```
def foo():
    try:
        print(1)
    finally:
        print(2)
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 will be the output of the following Python code?**

```
try:
    if '1' != 1:
        raise "someError"
    else:
        print("someError has not occurred")
except "someError":
    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.

**11. The following Python code will result in an error if the input value is entered as -5.**

```
assert False, 'Spanish'
```

a) True
b) False

Answer: a
Explanation: The code shown above results in an assertion error. The output of the code is:
Traceback (most recent call last):
File "<pyshell#0>", line 1, in <module>
assert False, 'Spanish'
AssertionError: Spanish
Hence, this statement is true.

**12. What will be the output of the following Python code?**

```
x = 10
y = 8
```

```
assert x > y00 , 'X too small'
```
a) Assertion Error
b) 10 8
c) No output
d) 108

Answer: c
Explanation: The code shown above results in an error if and only if x < y. However, in the above case, since x > y, there is no error. Since there is no print statement, hence there is no output.

## 13. What will be the output of the following Python code?

```
#generator
def f(x):
    yield x+1

g=f(8)
print(next(g))
```

a) 8
b) 9
c) 7
d) Error

Answer: b
Explanation: The code shown above returns the value of the expression x+1, since we have used to keyword yield. The value of x is 8. Hence the output of the code is 9.

## 14. What will be the output of the following Python code?

```
def f(x):
    yield x+1
    print("test")
    yield x+2

g=f(9)
```

a) Error
b) test
c)
test
10
12
d) No output

Answer: d
Explanation: The code shown above will not yield any output. This is because when we try to yield 9, and there is no next(g), the iteration stops. Hence there is no output.

## 15. What will be the output of the following Python code?

```
def f(x):
    yield x+1
    print("test")
    yield x+2

g=f(10)
print(next(g))
print(next(g))
```

**a)** No output
**b)**
11
test
12
**c)**
11
test
**d)** 11

Answer: b
Explanation: The code shown above results in the output:
11
test
12
This is because we have used next(g) twice. Had we not used next, there would be no output.

### 16. What will be the output of the following Python code?

```
def a():
    try:
        f(x, 4)
    finally:
        print('after f')
    print('after f?')
a()
```

a) No output
b) after f?
c) error
d) after f

Answer: c
Explanation: This code shown above will result in an error simply because ‘f’ is not defined. ‘try’ and ‘finally’ are keywords used in exception handling.

### 17. What will be the output of the following Python code?

```
def f(x):
    for i in range(5):
        yield i
g = f(8)
print(list(g))
```

a) [0, 1, 2, 3, 4]
b) [1, 2, 3, 4, 5, 6, 7, 8]
c) [1, 2, 3, 4, 5]
d) [0, 1, 2, 3, 4, 5, 6, 7]

Answer: a
Explanation: The output of the code shown above is a list containing whole numbers in the range (5). Hence the output of this code is: [0, 1, 2, 3, 4].

## 18. The error displayed in the following Python code is?

```
import itertools

l1=(1, 2, 3)

l2=[4, 5, 6]

l=itertools.chain(l1, l2)

print(next(l1))
```

a) 'list' object is not iterator
b) 'tuple' object is not iterator
c) 'list' object is iterator
d) 'tuple' object is iterator

Answer: b
Explanation: The error raised in the code shown above is that: 'tuple' object is not iterator. Had we given l2 as argument to next, the error would have been: 'list' object is not iterator.

## 19. Which of the following is not an exception handling keyword in Python?

a) try
b) except
c) accept
d) finally

Answer: c
Explanation: The keywords 'try', 'except' and 'finally' are exception handling keywords in python whereas the word 'accept' is not a keyword at all.

## 20. What will be the output of the following Python code?

```
g = (i for i in range(5))

type(g)
```

a) class <'loop'>
b) class <'iteration'>
c) class <'range'>
d) class <'generator'>

Answer: d
Explanation: Another way of creating a generator is to use parenthesis. Hence the output of the code shown above is: class<'generator'>.

**21. What happens if the file is not found in the following Python code?**

```
a = False
while not a:
    try:
        f_n = input("Enter file name")
        i f = open(f_n, 'r')
    except:
        print("Input file not found")
```

a) No error
b) Assertion error
c) Input output error
d) Name error

Answer: a
Explanation: In the code shown above, if the input file in not found, then the statement: "Input file not found" is printed on the screen. The user is then prompted to reenter the file name. Error is not thrown.

**22. What will be the output of the following Python code?**

```
lst = [1, 2, 3]

lst[3]
```

a) NameError
b) ValueError
c) IndexError
d) TypeError

Answer: c
Explanation: The snippet of code shown above throws an index error. This is because the index of the list given in the code, that is, 3 is out of range. The maximum index of this list is 2.

**23. What will be the output of the following Python code?**

```
t[5]
```

a) IndexError
b) NameError
c) TypeError
d) ValeError

Answer: b
Explanation: The expression shown above results in a name error. This is because the name '**t**' is not defined.

**24. What will be the output of the following Python code, if the time module has already been imported?**

```
4 + '3'
```

a) NameError
b) IndexError
c) ValueError
d) TypeError

Answer: d
Explanation: The line of code shown above will result in a type error. This is because the operand '+' is not supported when we combine the data types 'int' and 'str'. Sine this is exactly what we have done in the code shown above, a type error is thrown.

**25. What will be the output of the following Python code?**

```
int('65.43')
```

a) ImportError
b) ValueError
c) TypeError
d) NameError

Answer: b
Explanation: The snippet of code shown above results in a value error. This is because there is an invalid literal for int() with base 10: '65.43'.

**26. Compare the following two Python codes shown below and state the output if the input entered in each case is -6?**

```
#CODE 1
import math
num = int(input("Enter a number of whose factorial you want to find"))
print(math.factorial(num))

#CODE 2
num = int(input("Enter a number of whose factorial you want to find"))
print(math.factorial(num))
```

a) ValueError, NameError
b) AttributeError, ValueError
c) NameError, TypeError
d) TypeError, ValueError

Answer: a
Explanation: The first code results in a ValueError. This is because when we enter the input as -6, we are trying to find the factorial of a negative number, which is not possible. The second code results in a NameError. This is because we have not imported the math module. Hence the name 'math' is undefined.

**27. What will be the output of the following Python code?**

```
def getMonth(m):
    if m < 1 or m > 12:
        raise ValueError("Invalid")
    print(m)
```

```
getMonth(6)
```

a) ValueError
b) Invalid
c) 6
d) ValueError(“Invalid”)

Answer: c
Explanation: In the code shown above, since the value passed as an argument to the function is between 1 and 12 (both included), hence the output is the value itself, that is 6. If the value had been above 12 and less than 1, a ValueError would have been thrown.

**28. What will be the output of the following Python code if the input entered is 6?**

```
valid = False
while not valid:
    try:
        n = int(input("Enter a number"))
        while n%2 == 0:
            print("Bye")
        valid = True
    except ValueError:
        print("Invalid")
```

a) Bye (printed once)
b) No output
c) Invalid (printed once)
d) Bye (printed infinite number of times)

Answer: d
Explanation: The code shown above results in the word “Bye” being printed infinite number of times. This is because an even number has been given as input. If an odd number had been given as input, then there would have been no output.

**29. Identify the type of error in the following Python codes?**

```
Print(“Good Morning”)
print(“Good night)
```

a) Syntax, Syntax
b) Semantic, Syntax
c) Semantic, Semantic
d) Syntax, Semantic

Answer: b
Explanation: The first code shows an error detected during execution. This might occur occasionally. The second line of code represents a syntax error. When there is deviation from the rules of a language, a syntax error is thrown.

**30. Which of the following statements is true?**

a) The standard exceptions are automatically imported into Python programs

b) All raised standard exceptions must be handled in Python
c) When there is a deviation from the rules of a programming language, a semantic error is thrown
d) If any exception is thrown in try block, else block is executed

Answer: a
**Explanation**: When any exception is thrown in try block, except block is executed. If exception is not thrown in try block, else block is executed. When there is a deviation from the rules of a programming language, a syntax error is thrown. The only true statement above is: The standard exceptions are automatically imported into Python programs.

**31. Which of the following is not a standard exception in Python?**
a) NameError
b) IOError
c) AssignmentError
d) ValueError

Answer: c
Explanation: NameError, IOError and ValueError are standard exceptions in Python whereas Assignment Error is not a standard exception in Python.

**32. Syntax errors are also known as parsing errors.**
a) True
b) False

Answer: a
Explanation: Syntax errors are known as parsing errors. Syntax errors are raised when there is a deviation from the rules of a language. Hence the statement is true.

**33. An exception is _____**
a) an object
b) a special function
c) a standard module
d) a module

Answer: a
Explanation: An exception is an object that is raised by a function signaling that an unexpected situation has occurred, that the function itself cannot handle.

**34. _____ exceptions are raised as a result of an error in opening a particular file.**
a) ValueError
b) TypeError
c) ImportError
d) IOError

Answer: d
Explanation: IOError exceptions are raised as a result of an error in opening or closing a particular file.

**35. Which of the following blocks will be executed whether an exception is thrown or not?**
a) except
b) else

c) finally
d) assert

Answer: c
Explanation: The statements in the finally block will always be executed, whether an exception is thrown or not. This clause is used to close the resources used in a code.