CHAPTER : 1

EXCEPTION HANDLING

STUDY NOTES

- In Python, an error can stop the execution of program, make it behave abnormally or allow it to execute but deliver unexpected output.
- **Types of errors in Python:** Syntax errors and Logical errors(Runtime)
- Exceptions are errors that are triggered automatically. They can be:
  - ❖ Forcefully triggered and
  - ❖ Handled with code.

Syntax Errors:

- Caused when the programmer does not follow the rules or proper syntax at the time of coding.
- Syntax errors are also known as parsing errors.
- The interpreter can only execute a program that is free of syntax errors.
- It is important to fix syntax errors and save the file to allow it to run.
- When a syntax error is encountered, Python interpreter provides its name and description.

Exceptions:

- It is possible for a syntactically correct code to have errors that may show up at the time of execution.
- Errors that interrupt normal execution of a program are known as Exception.
- An exception is a Python object that represents an error.
- Exceptions are raised during runtime.

Built-In Exceptions:

- Built-in exceptions are commonly occurring exceptions defined by compiler/ interpreter.
- To deal with the errors, Python defines an extensive collection of exceptions in its standard library.
- When an error occurs, the associated exception handler code is executed informing the programmer to take the necessary action to handle it.

Raising Exceptions:

- Python throws an exception every time an error is encountered. This is called raising an exception.
- All exception handlers are available for execution.
- Programmers can use assert statement to forcefully raise exceptions.
- The syntax of raise statement is: raise exception-name[(optional argument)]
- The syntax for assert statement is: assert Expression[,arguments]

Handling Exceptions:

- To avoid the program from crashing, the programmer must write the code to handle exceptions. This is known as Exception handling.
- When the code designed to handle a particular exception is executed, we say that the exception is caught.

QUESTION BANK

MULTIPLE CHOICE QUESTIONS

1. What types of errors are also known as parsing errors?
   (a) Logical error
   (b) Keyboard interrupt error
   (c) Syntax error
   (d) Runtime error

2. Which of the following is not an exception?
   (a) Divide by zero
   (b) Missing colon ':' after the function body
   (c) Trying to open a file that does not exist
   (d) Using a '+' instead of '*' in a mathematical formula implemented in code

3. Which of the following is true for Exceptions:
   (i) Exceptions disrupt normal flow of the program
   (ii) It is common for exceptions to show up before the program is executed.
   (iii) Runtime errors are known as Exceptions
   (iv) Exception is a Python object that represents an error
   (a) (i), (ii), (iv)
   (b) (i), (iii), (iv)
   (c) (i), (ii)
   (d) (iv), (ii), (iii)

4. An error is a :
   (a) Bug
   (b) Folly
   (c) Spam
   (d) Virus

5. A Python program encountered a syntax error when the programmer tried to execute it. A dialog box specifying the name of the error and a small description was displayed. The programmer had written his code in:
   (a) Shell mode
   (b) Script mode
   (c) Command prompt
   (d) Anaconda

6. The only exception that is raised when the program is syntactically incorrect:
   (a) ValueError
   (b) SyntaxError
   (c) IO error
   (d) None of these

7. Commonly occurring exceptions defined in the compiler/interpreter are called:
   (a) Error
   (b) Exception
   (c) Syntax Error
   (d) Built-in Errors

8. Which of the following statements is not true?
   (a) Python's standard library is an extensive collection of built-in exceptions that deals with commonly occurring errors.
   (b) Python's standard library does not provide solution for exceptions
   (c) On occurrence of exception the associated handler code is executed
   (d) The handler code displays the reason along with the raised exception name.

9. What exception will be thrown in the following case?

```
def add()
      a = 10
      b = 10
      return a+b
add()
```

   (a) SyntaxError
   (b) valueError
   (c) IOError
   (d) None of these

10. What exception will occur in following code?

```
>>> import math
>>>math.sqrt(-4)
```

   (a) NumError
   (b) ValueError
   (c) TypeError
   (d) SignError

11. Using Python when you are not able to open a specified file what exception is raised?
   (a) SyntaxError
   (b) ValueError
   (c) Filenotfound error
   (d) IOerror

12. Accidentally hitting the "Delete" or "Escape" key while a program is executing will raise:
    (a) KeyboardError
    (b) InterruptError
    (c) KeyboardInterruptError
    (d) InterruptKeyboardError

13. When the end of file condition is reached without reading any data by input(), it will raise:
    (a) IndexError (b) EOFError (c) NameError (d) IndentationError

14. Throwing an exception in Python is known as:
    (a) Raising an exception
    (b) Handling an exception
    (c) Creating an exception
    (d) All of these

15. User-defined exceptions are:
    (a) Available in Python directory
    (b) Defined by the user
    (c) Defined by the programmer
    (d) Occurs when user does something wrong

16. Which of the following does not interrupt the execution of a program?
    (a) Syntax Error (b) Logical Errors (c) Exceptions (d) Runtime Error

17. One can deal with an exception when it is raised by the help of:
    (a) Python Library (b) Exception handlers (c) Testing (d) Debugging

18. Which of the following is not true for exceptions?
    (a) If an exception is raised no statement in the current block of code is executed.
    (b) Raising an exception interrupting the normal flow execution of program.
    (c) Control jumps to exception handler code.
    (d) The exception handling code is written to purposely change the flow of execution of a program.

19. On giving the "raise syntaxError("HI")" command on Python shell, no exception is raised because the right way to write is :
    (a) raise syntaxError("HI")
    (b) raise SyntaxError("HI")
    (c) raise SyntaxErrorException("HI")
    (d) raise syntaxErrorException("HI")

20. An expression in the program code can be tested with:
    (a) Strings (b) Assert (c) Stack traceback (d) Exception

21. Which of the following statement is not true for assert?
    (a) Checks for valid input
    (b) Used for evaluation of expression
    (c) If expression is false AssertionErrorException is raised
    (d) It imposes an exception on the code

22. What would be the output of the following code?

```
def evenNumber(number):
    assert(number%2==0),"not even"
    print("Number is even")
evenNumber(7)
```

    (a) No output
    (b) Not even
    (c) Number is even
    (d) Assertion Error: not even

23. A programmer writes a code to avoid abrupt crashing of a system. What is he working on?
    (a) Testing procedures
    (b) Assert statements
    (c) Exception handling
    (d) All of these

24. What is not true for exception handling?
    (a) It is used in all programming languages
    (b) Helps in capturing runtime errors
    (c) Is only done for built-in exceptions
    (d) Separates the main logic of the code from the error detection and correction code

25. When an error occurs an exception object is created by the:
(a) Code (b) Interpreter (c) Shell (d) Standard library

26. A complex mechanism consisting of hardware and software that comes into action as soon as the program, written in any programming language is put for execution is known as:
(a) Python shell (b) Python editor
(c) Runtime systems (d) Development Environment

27. An error object does not have information of:
(a) Error type (b) Error Name (c) Error Location (d) Error time

28. Throwing an exception means:
(a) Creating an error
(b) Creating an exception and handling it at runtime system
(c) Creating exception handle (d) Creating user defined exception

29. In exception handling, what keeps track of exact position where error has occurred?
(a) Registers (b) Primary Memory (c) Interpreter (d) Cache

30. What happens immediately after an error is encountered?
(a) Exception object is created (b) Exception is raised
(c) Runtime system searches for the exception handler in the current method.
(d) None of the above

31. Put in correct order:
(i) Create exception object (ii) Error is encountered
(iii) Exception is raised
(iv) Runtime system searches for exception handler in the current method.
(a) 1,2,3,4 (b) 3,2,1,4 (c) 2,1,3,4 (d) 4,1,2,3

32. How does the runtime system look for methods in callStack when the exception occurs?
(a) Reverse sequence (b) Forward sequence
(c) Random sequence (d) As per instructions given in the handler

33. When the code designed to handle an exception is executed, it is said that the exception is:
(a) Thrown (b) Raised (c) Caught (d) a. and b.

34. Exceptions are caught using:
(a) try catch block (b) try except block (c) if else (d) conditional statement

35. The suspicious lines of code where there is doubt is placed inside:
(a) try block (b) except block (c) finally (d) def

36. The code to handle the exception is written in:
(a) try (b) except (c) (a) and (b) (d) finally

37. While executing the program, if an exception is encountered in try block then execution of the code inside the try block is stopped and:
(a) code of except block is executed and control is moved to the beginning of try block
(b) control goes to except block and program stops.
(c) Control goes to except block and lines after try...except are executed.
(d) Except bock is ignored.

38. A code is suspected to have more than one exception. Two except blocks were written but the exception encountered was a different one. What can be done in this case?
(a) Add another except block without specifying exception
(b) Use only try block (c) Use only else block (d) Use finally

39. If there is a code that you want to execute irrespective of whether the execution has occurred or not is put in:
(a) else block (b) finally block (c) except block (d) after except block

40. Code in a try block is executed and not handled. It has a finally block. What will happen?

(a) Code of finally will be executed
(b) Finally, block is executed and exception is re-raised
(c) Control goes to first except block
(d) (a) and (b)

41. Difference between the finally and except block is that:

(a) finally does not terminate exception
(b) finally is not used with try and except is used with try
(c) (a) and (b)
(d) finally has the code that should be executed only after an exception is encountered.

42. The else part of try-except-else executes:

(a) Irrespective of whether an exception occurs or not
(b) When an exception is encountered
(c) When no exception is not encountered
(d) When code of except block is executed

43. What is the correct syntax for try-except finally?

(a)
```
try:
    Code
finally:
Code
```

(b)
```
finally:
    Code
try:
    Code
```

(c)
```
try:
    Code
except:
    Code
finally:
    Code
```

(d) Both (a) and (c)

44. The finally block is always executed...

(a) between try and except
(b) after leaving the try statement
(c) before try
(d) At the end of the program

45. If finally is used with a try block without using an except statement:

(a) Exception will occur
(b) Exception will not occur
(c) Exception will not be handled
(d) None of these

46. What will be the output of the following code?

```
def re():
   try:
      print("1")
      return 7
   finally:
      print("2")
      return 6
print(re())
```

(a) 1 2 6
(b) 1 7 2
(c) 1 2 7
(d) 6 2 1

47. The code of _____ block is always executed regardless of whether an exception occurred in the try block or not.

(a) try
(b) except
(c) else
(d) finally

48. _____ and _____ are used to raise exceptions.

(a) Raise and try
(b) Raise and assert
(c) Try and except
(d) Else and finally

49. Which is not a built in Exception?

(a) ImportError
(b) IOError
(c) NameError
(d) None of these

50. In Python syntax errors are :

(a) Handled as exception
(b) Not handled
(c) Are different type of errors
(d) None of these

INPUT TEXT BASED MCQs

Read the following passage and answer the following questions (51 to 53).

The try statement in Python can also have an optional finally clause. The statements inside the finally block are always executed regardless of whether an exception has occurred in the try block or not. It is a common practice to use finally clause while working with files to ensure that the file object is closed. If used, finally should always be placed at the end of try clause, after all except blocks and the else block.

Based on the information given above, look at the following code:

```
print("Exception Handling")
x = int(input("Enter a number :"))
y = int(input("Enter second number :"))
try:
      print(x/y)
#except ZeroDivisionError:
  #     print("Divide by zero error")
finally:
      Print("Done")
```

51. What will be the output of the code given above if the user enters x= 2 and y = 0?

(a) "Done" is printed first and then error is displayed (b) Error is displayed and "Done" is printed

(c) Error occurs and program stops (d) Nothing happens

52. What will be the output if the except block is uncommented and the user provide x = 6 and y = 2?

(a) 2.0 (b) Exception handling 3.0 Done

(c) Exception handling 2.0 (d) None of these

53. What will be the output if the except block is uncommented, the code inside finally block is print("Done") and the user provide x = 6 and y =2?

(a) 2.0 (b) Exception handling 2.0 Done

(c) Exception handling 2.0 (d) None of these

ANSWERS

Multiple Choice Questions

1. (c)	**2.** (b)	**3.** (b)	**4.** (a)	**5.** (b)	**6.** (b)	**7.** (d)	**8.** (b)	**9.** (a)	**10.** (b)
11. (d)	**12.** (a)	**13.** (b)	**14.** (a)	**15.** (c)	**16.** (b)	**17.** (b)	**18.** (d)	**19.** (c)	**20.** (b)
21. (d)	**22.** (d)	**23.** (c)	**24.** (c)	**25.** (c)	**26.** (c)	**27.** (d)	**28.** (b)	**29.** (c)	**30.** (a)
31. (c)	**32.** (b)	**33.** (c)	**34.** (d)	**35.** (a)	**36.** (b)	**37.** (c)	**38.** (a)	**39.** (b)	**40.** (b)
41. (a)	**42.** (c)	**43.** (d)	**44.** (b)	**45.** (c)	**46.** (a)	**47.** (d)	**48.** (b)	**49.** (d)	**50.** (a)

Input Text Based MCQs

51. (c) **52.** (c) **53.** (b)

HINTS/EXPLANATION

2. Syntax Error is a compilation error.

3. Option (b) is incorrect. When an error occurs during the execution of a program, an exception is said to have been raised.

5. The code was written in script mode therefore the error was displayed in a dialog box. If the code was written in the shell mode then the description of the code would have displayed in the shell itself.

7. Commonly occurring exceptions defined in the compiler/interpreter are called built-in exceptions.

8. (b) is wrong because Python's standard library does provide solution for exception.

13. When the end of file condition is reached without reading any data by input(), it will raise EOFError.

15. User-defined exceptions are defined by the programmer.

17. One can deal with an exception when it is raised with the help of an Exception handler.

18. The last statement "The exception handling code is written to purposely change the flow of execution of a program." Is wrong because exception handling is done to save the program from behaving abnormally.

19. The correct way to raise a syntax error exception is raise SyntaxErrorException("HI").

22. 7 is not an even number. The remainder for 7/2 is not zero hence AssertionError is raised.

23. When a programmer writes a code to avoid abrupt crashing of a system, it mears he is working on exception handling.

24. Exception handling can be done for built-in exceptions as well as user-defined exceptions.

26. A complex mechanism consisting of hardware and software that comes into action as soon as the program, written in any programming language is put for execution is known as Runtime systems.

28. Throwing an exception means creating an exception and handling it at runtime system.

29. In exception handling, the interpreter keeps track of exact position where error has occurred.

31. The correct order is:
    1. Error is encountered 2. Create exception object 3. Exception is raised
    4. Runtime system searches for exception handler in the current method.

32. When an exception occurs the runtime system look for methods in call Stack in reverse sequence.

34. Exceptions are caught using try except block.

37. While executing the program, if an exception is encountered in try block then execution of the code inside the try block is stopped and control goes to except block and lines after try...except are executed.

40. Finally, block is executed and exception is re-raised

42. Finally can be used with try-except-else executes when no exception is not encountered.

43. Two ways to use finally with try are as follows:

(a)
```
try:
Code
finally:
Code
```

(b)
```
try:
Code
except:
Code
finally:
Code
```

45. If finally is used with a try block without using an except statement, the exception will not be handled.

46. Since, the finally block returns a value the value 7 is not returned.

49. ImportError, IOError and NameError are all built-in exceptions.