EXCEPTION HANDLING AND FILE HANDLING IN PYTHON

STUDY NOTES

- **Syntax Error:** Syntax errors are produced by Python when it is translating the source code into byte code. They usually indicate that there is something wrong with the syntax of the program. **Example:** Omitting the colon at the end of a def statement yields the somewhat redundant message **SyntaxError: invalid syntax.**

- **Exception:** An exception is an event, which occurs during the execution of a program that disrupts the normal flow of the program's instructions. In general, when a Python script encounters a situation that it cannot cope with, it raises an exception. An exception is a Python object that represents an error. When a Python script raises an exception, it must either handle the exception immediately otherwise it terminates and quits.

- **User-defined exception:** In Python, users can define custom exceptions by creating a new class. This exception class has to be derived, either directly or indirectly, from the built-in Exception class. Most of the built-in exceptions are also derived from this class.

- **Raising exception:** The raise keyword is used to raise an exception. The raise keyword raises an error and stops the control flow of the program.

- **The try-except statement:** If the Python program contains suspicious code that may throw the exception, we must place that code in the try block. The try block must be followed with the except statement, which contains a block of code that will be executed if there is some exception in the try block.

  Syntax

```
try:
    #block of code
except Exception1:
    #block of code
except Exception2:
    #block of code
#other code
```

- **Finally clause:** It defines a block of code to run when the try... except...else block is final. The finally block will be executed no matter if the try block raises an error or not. This can be useful to close objects and clean up resources.

- **Built in exceptions:** The table below shows built-in exceptions that are usually raised in Python:

Exceptions	Description
ArithmeticError	Raised when an error occurs in numeric calculations
AssertionError	Raised when an assert statement fails
AttributeError	Raised when attribute reference or assignment fails
Exception	Base class for all exceptions
EOFError	Raised when the input() method hits an "end of file" condition (EOF)
FloatingPointError	Raised when a floating point calculation fails

GeneratorExit	Raised when a generator is closed (with the close() method)
ImportError	Raised when an imported module does not exist
IndentationError	Raised when indendation is not correct
IndexError	Raised when an index of a sequence does not exist
KeyError	Raised when a key does not exist in a dictionary
KeyboardInterrupt	Raised when the user presses Ctrl+C, Ctrl+Z or Delete
LookupError	Raised when errors raised cant be found
MemoryError	Raised when a program runs out of memory
NameError	Raised when a variable does not exist
NotImplementedError	Raised when an abstract method requires an inherited class to override the method
OSError	Raised when a system related operation causes an error
OverflowError	Raised when the result of a numeric calculation is too large
ReferenceError	Raised when a weak reference object does not exist
RuntimeError	Raised when an error occurs that do not belong to any specific exceptions
StopIteration	Raised when the next() method of an iterator has no further values
SyntaxError	Raised when a syntax error occurs .
TabError	Raised when indentation consists of tabs or spaces
SystemError	Raised when a system error occurs
SystemExit	Raised when the sys.exit() function is called
TypeError	Raised when two different types are combined
UnboundLocalError	Raised when a local variable is referenced before assignment
UnicodeError	Raised when a unicode problem occurs
UnicodeEncodeError	Raised when a unicode encoding problem occurs
UnicodeDecodeError	Raised when a unicode decoding problem occurs
UnicodeTranslateError	Raised when a unicode translation problem occurs
ValueError	Raised when there is a wrong value in a specified data type
ZeroDivisionError	Raised when the second operator in a division is zero

- **File Handling:** File is a named location on disk to store related information. It is used to permanently store data in a non-volatile memory (e.g. hard disk).

There are three types of files that can be handled in Python, normal text files, binary files (written in binary language,0s and 1s) and CSV files.

Access Modes	Description
r	Opens a file for reading only. The file pointer is placed at the beginning of the file. This is the default mode. Gives error if file does not exist.
rb	Opens a file for reading only in binary format. The file pointer is placed at the beginning of the file. This is the default mode.
r+	Opens a file for both reading and writing. The file pointer placed at the beginning of the file.

rb+	Opens a file for both reading and writing in binary format. The file pointer is placed at the beginning of the file.
w	Opens a file for writing only. Overwrites the file if the file exists. If the file does not exist, creates a new file for writing.
wb	Opens a file for writing only in binary format. Overwrites the file if the file exists. If the file does not exist, creates a new file for writing.
w+	Opens a file for both writing and reading. Overwrites the existing file if the file exists. If the file does not exist, creates a new file for reading and writing.
wb+	Opens a file for both writing and reading in binary format. Overwrites the existing file if the file exists. If the file does not exist, creates a new file for reading and writing.
a	Opens a file for appending. The file pointer is at the end of the file if the file exists. That is, the file is in the append mode. If the file does not exist, it creates a new file for writing.
ab	Opens a file for appending in binary format. The file pointer is at the end of the file if the file exists. That is, the file is in the append mode. If the file does not exist, it creates a new file for writing.
a+	Opens a file for both appending and reading. The file pointer is at the end of the file if the file exists. The file opens in the append mode. If the file does not exist, it creates a new file for reading and writing.
ab+	Opens a file for both appending and reading in binary format. The file pointer is at the end of the file if the file exists. The file opens in the append mode. If the file does not exist, it creates a new file for reading and writing.

- **Text File:** Text files are structured as a sequence of lines, where each line includes a sequence of characters. Each line is terminated with a special character, called the EOL or End of Line character. There are several types, but the most common is the comma {,} or newline character. It ends the current line and tells the interpreter a new one has begun.

- In Python, a file operation takes place in the following order.
    1. Open a file
    2. Read or write (perform operation)
    3. Close the file

- **Open a file:** To open the file, use the built-in open() function. The open() function returns a file object, it is used to read or modify the file accordingly

    Syntax

    ```
    file_object = open("filename", "mode")
    ```

    where

    file_object is the variable to add the file object.

    Filename is the name of file

    Mode(optional) tells which way the file will be used.

- Mode specifies whether we want to read 'r', write 'w' or append 'a' to the file. We also specify if we want to open the file in text mode or binary mode. The default is reading in text mode. In this mode, we get strings when reading from the file.

    For example:

    ```
    f=open("abc.txt","w")
    ```

    where

```
f is the object of file
"abc.txt" is the name of file
"w" is the mode
```

- **Closing a file:** The close() function closes the file and frees the memory space acquired by that file.

  You should always close your files, in some cases, due to buffering, changes made to a file may not show until you close the file.

```
f=open("abc.txt","w")

...  .  .  ..  .  .

f.close()
```

- **With statement:** The with statement will automatically close the file after the nested block of code.

```
with open("filename") as file:

do something with data

with open('abc.txt', 'w') as f:

      f.write('Hi there!')
```

- **Writing in file:** There are two ways to write in a file.
  1. **write():** Inserts the string str1 in a single line in the text file.
     ```
     File_object.write(str1)
     ```
  2. **writelines():** For a list of string elements, each string is inserted in the text file. Used to insert multiple strings at a single time.
     ```
     L = [str1, str2, str3]
     File_object.writelines(L)
     ```

- **Reading from text file:** Using read() function: Read n no of bytes and returns it in form of a string. If n is not specified, reads the entire file.

  File_object.read([n])

- **Using readline():** Reads a line of the file and returns in form of a string.

  If n is given, reads at most n bytes. However, does not reads more than one line, even if n exceeds the length of the line.

  File_object.readline([n])

- **Using readlines():** Reads all the lines and return them as each line a string element in a list.

  File_object.readlines()

- **Binary Files:** A binary file is a file stored in binary format. A binary file is computer-readable but not human-readable. All executable programs are stored in binary files, as are most numeric data files. In contrast, text files are stored in a form (usually ASCII) that is human-readable.

- **Differentiate between Binary Files and Text Files: Text files:** In this type of file, each line of text is terminated with a special character called EOL (End of Line), which is the new line character ('\n') in Python, by default. A text file stores data in the form of alphabets, digits and other special symbols by storing their ASCII values and are in a human readable format.

- **Binary files:** In this type of file, there is no terminator for a line and the data is stored after converting it into machine understandable binary language. A binary file stores the data in the same way as stored in the memory. The .exe files, mp3 file, image files, word documents are some of the examples of binary files. We can't read a binary file using a text editor.

- **Pickle:** Python Pickle is used to serialize and deserialize a Python object structure. It's the process of converting a Python object into a byte stream to store it in a file/database. Any object on Python can be pickled so that it can be saved on disk.

- **Writing in Binary File:** For writing in binary file, open the file in 'wb'(or 'w+b') mode. The 'w' means that you'll be writing to the file, and 'b' refers to binary mode.
  ```
  f=open(filename,"wb")
  ```
- **Using dump():** dump() is used for writing in file. It takes two parameters, the object you want to pickle and the file to which the object has to be saved.
  ```
  dump('object to write', 'file object')
  ```
- **Reading from file:** For reading a binary file, open the file in 'rb' mode, 'r' means you'll be reading from file.
  ```
  f=open(filename,"rb")
  ```
- **using load():** load() is used for reading from file. It takes one 'file object' parameter, reading data from and then assigned the contents of file to new variable.

QUESTION BANK

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

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

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
   (c) No, finally must come before except
   (b) No, finally cannot be used with except
   (d) Yes

4. Can one block of except statements handle multiple exception?
   (a) Yes, like except TypeError, SyntaxError [,...]
   (c) No
   (b) Yes, like except [TypeError, SyntaxError]
   (d) None of the mentioned

5. 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

6. Which of the following blocks will be executed whether an exception is thrown or not?
   (a) except　　　(b) else　　　　(c) finally　　　(d) assert

7. Which block lets you test a block of code for errors?
   (a) try　　　　(b) except　　　(c) finally　　　(d) None of these

8. What will be output for the following code?
   ```
   try:
   print(x)
   except:
    print(""An exception occurred"")
   ```
   (a) x
   (c) Error
   (b) An exception occurred
   (d) None of these

9. Which of the following elements are used in Exception handling in Python?
   (a) try   (b) except   (c) finally   (d) All of these

10. Syntax Errors are also known as:
   (a) Logical Errors   (b) Parsing Errors   (c) Code Errors   (d) Programming Errors

11. "wb" mode in file handling also can be written as:
   (a) w*b   (b) w/b   (c) w+b   (d) None of these

12. _____ function is used to write in pickle file.
   (a) write   (b) dump   (c) store   (d) pick

13. Mode used for reading from pickle file is _____.
   (a) rb   (b) r   (c) br   (d) read

14. _____ function is used to read from pickle file in Python.
   (a) read   (b) readdata   (c) pickread   (d) load

15. When file is opened as "with open" then file is _____ automatically.
   (a) read   (b) write   (c) close   (d) open

16. Which of the following options is correct?
   (a) If file opens in r mode, file is opened for reading only
   (b) If file opens in r mode, file is opened for reading and writing both
   (c) If file opens in r mode, file is opened for writing only
   (d) If file opens in r mode, file is opened for reading and appending

17. Which of the following options is correct?
   (i) If file is opens in r mode, the file pointer is placed at the beginning of the file
   (ii) If file is opens in r mode, the file pointer is placed at the end of the file
   (iii) If file is opens in r+ mode, the file pointer is placed at the beginning of the file
   (iv) If file is opens in r +mode, the file pointer is placed at the end of the file
   (a) only (i)   (b) (i) and (iii) both   (c) (ii) and iii both   (d) (ii) and iv

18. Which of the following option is correct?
   (a) The default mode in text file is r
   (b) The default mode in text file is r+
   (c) The default mode in text file is w
   (d) The default mode in text file is rw

19. Which of the following options is correct?
   (a) Opening file in rb mode means opened a file for reading only in binary format.
   (b) Opening file in rb mode, the file pointer is placed at the beginning of the file.
   (c) rb mode is the default mode in binary file.
   (d) All are correct

20. Which of the following options is correct?
   (a) Opening file in r+ mode means opening a file for reading only.
   (b) When file is opened in r+ mode, the file pointer placed at the beginning of the file
   (c) r+ is the default mode
   (d) All of the above

21. Which of the following options can be used to read complete text file as String?
   (a) myfile = open('Myfile.txt'); myfile.read()   (b) myfile = open('Myfile.txt','r'); myfile.read(n)
   (c) myfile = open('Myfile.txt'); myfile.readline()   (d) myfile = open('Myfile.txt'); myfile.readlines()

22. Which of the following options can be used to read all the lines and return them as each line a string element in a list?
   (a) myfile = open('Myfile.txt'); myfile.read()   (b) myfile = open('Myfile.txt','r'); myfile.read(n)
   (c) myfile = open('Myfile.txt'); myfile.readline()   (d) myfile = open('Myfile.txt'); myfile.readlines()

23. Which of the following options can be used to read at most n number of bytes, can read more than one line?
   (a) myfile = open('Myfile.txt'); myfile.read()   (b) myfile = open('Myfile.txt','r'); myfile.read(n)
   (c) myfile = open('Myfile.txt'); myfile.readline(n) (d) myfile = open('Myfile.txt'); myfile.readlines()

24. Which of the following options can be used to read the two lines of a text file in form of string from Myfile.txt? Assuming the file having two lines only.
   (a) myfile = open('Myfile.txt'); myfile.read()   (b) myfile = open('Myfile.txt','r'); myfile.read(2)
   (c) myfile = open('Myfile.txt'); myfile.readline(2) (d) myfile = open('Myfile.txt'); myfile.readlines()

25. Assume that the position of the file pointer is at the beginning of 3rd line in a text file. Which of the following option can be used to read all the remaining lines?
   (i) myfile.read()   (ii) myfile.read(n)   (iii) myfile.readline()   (iv) myfile.readlines()
   (a) (i) and (iii)   (b) (i) and (iv)   (c) (ii) and (iv)   (d) (ii) and (iii)

26. Assume that the position of the file pointer is at the beginning of 3rd line in a text file. Which of the following option can be used to read from the beginning of the file?
   (a) myfile.read()   (b) myfile.readline()   (c) myfile.readlines()   (d) None of these

27. Which of the following is the correct option to close the file?
   (a) f.close   (b) f.close()   (c) f(close)   (d) close(f)

28. _____ function closes the file and frees the memory space acquired by that file.
   (a) close()   (b) close   (c) closed()   (d) closes()

29. Identify the correct option out of the following options to open the binary file in read mode.
   (a) myfile = open('student','rb')   (b) myfile = open('student','r+b')
   (c) myfile = open('student','br')   (d) all of the above

30. Identify the correct option out of the following options to open the binary file for reading and writing both.
   (i) myfile = open('student','rb+')   (ii) myfile = open('student','wb+')
   (iii) myfile = open('student','rwb')   (iv) myfile = open('student',rb+wb+)
   (a) only (i)   (b) both (i) and (iv)   (c) both (iii) and (iv)   (d) both (i) and (ii)

31. Which option is correct when a file is opened in 'a' mode?
   (a) Opens a file for appending.
   (b) The file pointer is at the end of the file if the file exists.
   (c) If the file does not exist, it creates a new file for writing.
   (d) All of the above

32. Which of the following statement is incorrect in the context of binary files?
   (a) A binary file stores the data in the same way as stored in the memory.
   (b) We can read a binary file using a text editor.
   (c) The .exe files, mp3 file, image files, word documents are some of the examples of binary files.
   (d) There is no terminator for a line.

33. Which type of file stores the data in the same way as stored in the memory?
   (a) Text   (b) Binary   (c) CSV   (d) All of these

34. Observe the following code
   ```
   File = open("Mydata","a")
   _____     #Blank1
   File.close()
   ```
   Fill the Blank 1 with statement to write "ABC" in the file "Mydata"
   (a) File.write("ABC")   (b) File.writelines("ABC")   (c) File.writedata("ABC")   (d) File.writes("ABC")

35. EOL stands for _____.
   (a) End of Line   (b) End of List   (c) End of Lists   (d) None of these

36. Which of the following statements is correct to store list L in binary file object f.p is the object of pickle library?
    (a) p.dump(L,f)   (b) L.dump(p,f)   (c) f.dump(L,p)   (d) p.dump(f,L)

37. _____ module is used for serializing and de-serializing a Python object structure.
    (a) Random   (b) Math   (c) Serial   (d) Pickle

38. Which statement is correct to open binary file for writing?
    (a) f.open("file",wb)
    (b) f=open("file",w+b)
    (c) open(f="file",bw)
    (d) f[open("file",wb)]

39. Which statement is correct to unpickle data into list where a is file object, b pickle object and c is the name of list?
    (a) c=b.load(a)   (b) a=b.load(c)   (c) b=a.load(b)   (d) a=c.load(b)

40. Which of the following statements is correct?
    (a) import pickle as i
    (b) import pickles as p
    (c) import Pickle as P
    (d) import Pickles as I

41. Which of the following options is incorrect?
    (a) Reading and Writing both can be done when file is opened in r+ mode.
    (b) When file is opened in r+ mode the file pointer get placed at the beginning of the file.
    (c) Only Reading in binary file can be done when file is opened in rb+ mode.
    (d) When file in opened in rb+ mode the file pointer placed at the beginning of the file.

42. Which of the following options is correct?
    (a) Opens a file in w mode for writing only
    (b) If we try to open a text file in w mode that exists, overwrites the file
    (c) If we try to open a text file in w mode that does not exist, the file gets Created.
    (d) All of the above

43. Which of the following options is correct?
    (a) Opens a file for writing only in binary file in wb mode
    (b) If we try to open a binary file in wb mode that exists, overwrites the file
    (c) If we try to open a binary file in wb mode that does not exist, the file gets Created.
    (d) All of the above

44. Which of the following options is not correct?
    (a) If we try to open text file in r mode that does not exist, an error occurs.
    (b) If we try to open file in w mode that does not exist, the file gets created.
    (c) If we try to open file in r+ mode that does not exist, no error occurs.
    (d) If we try to open file in w+ mode that does not exist, the file gets created.

45. Which of the following options is not correct?
    (a) If we try to open text file in r mode that does not exist, an error occurs.
    (b) If we try to open file in wb mode that does not exist, the file gets created.
    (c) If we try to open file in rb+ mode that does not exist, no error occurs.
    (d) If we try to open file in wb+ mode that does not exist, the file gets created.

46. The file pointer placed at the beginning of the file when the file is opened in:
    (a) r   (b) r+   (c) rb   (d) All of these

47. The file pointer placed at the beginning of the file when the file is opened in:
    (a) r mode   (b) w mode   (c) a mode   (d) All of these

48. The file pointer placed at the end of the file when the file is opened in:
    (a) r mode   (b) w mode   (c) a mode   (d) All of these

49. The file pointer placed at the beginning of the file when the file is opened in:
    (a) a mode   (b) ab mode   (c) a+ mode   (d) all of these

50. Rahul is trying to write a Dictionary Dic={1:'Aman',2:'Karan',3:'Diya'}on a binary file test.bin. Consider the following code written by him.

```
import pickle as p
Dic={1:'Aman',2:'Karan',3:'Diya'}
with open('test','wb') as f:
    _____        #statement 1
```

Identify the missing code in Statement 1.
    (a) p=dump(f)   (b) p.dump(Dic,f)   (c) p,dump(f,Dic)   (d) f.dump(Dic,p)

51. Rahul is trying to read data from a binary file test.bin and store it into List L. Consider the following code written by him.

```
import pickle as p
with open('test','rb') as f:
    _____        #statement 1
Identify the missing code in Statement 1.
```
    (a) f=p.load(L)   (b) p=p.load(L)   (c) L=p.load(f)   (d) f=L.load(p)

52. Manoj is trying to open binary file to write data but facing some problems. Select the correct option to open the file "STUDENT.DAT" for writing only in binary format.
    (a) F= open("STUDENT.DAT",'wb')
    (b) F= open("STUDENT.DAT",'w')
    (c) F= open("STUDENT.DAT",'wb+')
    (d) F= open("STUDENT.DAT",'w+')

53. Which of the following statement(s) are correct regarding the file access modes?
    (a) 'r+' opens a file for both reading and writing. File object points to its beginning.
    (b) 'w+' opens a file for both writing and reading. Adds at the end of the existing file if it exists and creates a new one if it does not exist.
    (c) 'wb' opens a file for reading and writing in binary format. Overwrites the file if it exists and creates a new one if it does not exist.
    (d) 'a' opens a file for appending. The file pointer is at the start of the file if the file exists.

54. Suppose content of 'Myfile.txt' is:
    Twinkle twinkle little star
    How I wonder what you are
    Up above the world so high
    Like a diamond in the sky
    What will be the output of the following code?

```
f = open("Myfile.txt")
data=f.readline()
data=f.readline()
print(data)
f.close()
```

    (a) Twinkle twinkle little star   (b) How I wonder what you are
    (c) Twinkle twinkle little star   (d) Twinkle twinkle little star
        How I wonder what you are   How I wonder what you are
        Up above the world so high
        Like a diamond in the sky

55. Suppose content of 'Myfile.txt' is:

Twinkle twinkle little star
How I wonder what you are
Up above the world so high
Like a diamond in the sky
What will be the output of the following code?

```
f = open("Myfile.txt")
data=f.readline()
data=f.read()
print(data)
f.close()
```

(a) Twinkle twinkle little star

(b) How I wonder what you are

(c) Twinkle twinkle little star
How I wonder what you are
Up above the world so high
Like a diamond in the sky

(d) How I wonder what you are
Up above the world so high
Like a diamond in the sky

56. Suppose content of 'abc.txt' is:

India is Great
I love
my India
What will be the output of the following code?

```
f=open('abc.txt','r')
data=f.readline(20)
print(data)
f.close()
```

(a) India is Great
    I lov

(b) India is Great
    I love

(c) India is Great

(d) India is Great
    I love
    my I

57. Suppose content of 'abc.txt' is:

India is Great
I love
my India
What will be the output of the following code?

```
f=open('abc.txt','r')
data=f.read(20)
print(data)
f.close()
```

(a) India is Great
    I lov

(b) India is Great
    I love

(c) India is Great

(d) India is Great
    I love

58. Which of the following options can be used to read the 2nd line of a text file?

(a) file=f.readlines()
    print(file[1])

(b) file=f.readline()
    print(file[1])

(c) file=f.read()
    print(file[1])

(d) file=f.readlines()
    print(file[2])

59. Which of the following is not a file access mode?

(a) rb

(b) rb+

(c) wb

(d) rw

60. In f=open("abc.txt","w"), f refers to?

(a) File name

(b) File object

(c) File access mode

(d) File pointer

61. In f=open("abc.txt","w"), 'w' refers to:
    (a) File name   (b) File object
    (c) File access mode   (d) File pointer

62. In f=open("abc.txt","w"), "abc.txt" refers to:
    (a) file name   (b) file object
    (c) file access mode   (d) function

63. Which statement is correct to open the file using with statement?
    (a) with open('abc.txt', 'w') to f: (b) open('abc.txt', 'w') with f:
    (c) f=with open('abc.txt', 'w')   (d) with open('abc.txt', 'w') as f:

64. When the method display() is executed it read lines from a text file DIARY.TXT, and display those lines, which are starting with an alphabet 'P'.

```
def display():
    file=open('DIARY.TXT','r')
    l=file.readlines()
    for line in l:
        if _____ :    # statement 1
            print (line)
    file.close()
```

Select the appropriate statement to fill the statement 1.
    (a) line[1]= ='P'   (b) line[0] ='P'
    (c) line(0)= ='P'   (d) line[0]= ='P'

65. When the function ABLINES() is executed, it read contents from a text file LINES.TXT, to display those lines, which are either starting with an alphabet 'A' or starting with alphabet 'B'.

```
def ABLINES():
    file=open('LINES.TXT','r')
    lines = file.readlines()
    for w in lines:
        if _____ :   #statement 1
            print (w)
    file.close()
```

Select the appropriate statement to fill the statement 1
    (i) w[0]= ="A" or w[0]= ="B"   (ii) w[0] in ["A","B"]
    (iii) w[0]= ="A" and w[0]= ="B"   (iv) w[0]= ="A","B"

    (a) (i) and (ii)   (b) (ii) and (iv)   (c) (i) and (iv)   (d) (ii) and (iii)

INPUT TEXT BASED MCQs

Read the following passage and answer the following questions (66 to 69).

It is a common misconception that a character can be expressed using a single character. That is not true. Some characters that take up more than a character when expressed are '\n', '\r', '\t', etc. So, it is allowed to use single quotes around this character. Also, in Python you can express a string by enclosing it within single, double or triple quotes also. An important character that makes sense to discuss is the EOF character. It is present at the end of the file as a delimiter. When you open a file in Python, a file object is created and all operations of the file are done via the file object after that. To read the contents of the file character by character, read() routine may be used. To read a line at a time, readlines() routine may be used. While you should close the file object using close(), if you use the with method, the file object opens for use and after its use in the code block, closes also automatically.

66. How many characters does the string '\t' contain?
    (a) One   (b) Two   (c) Three   (d) Four

67. You may insert multiple strings at the input point of the file using:

(a) Read() (b) readline() (c) writelines() (d) write()

68. In Python, what does the readlines() routine return?

(a) It returns a list of lines (b) It returns the EOF character

(c) It returns the number of lines (d) None of these

69. What does the read() routine take as its input argument?

(a) Number of characters to read at a time

(b) The file object

(c) The string variable name to read the characters into

(d) Nothing

ANSWERS

Multiple Choice Questions

1. (d)	2. (c)	3. (d)	4. (a)	5. (d)	6. (c)	7. (a)	8. (b)	9. (d)	10. (b)
11. (c)	12. (b)	13. (a)	14. (a)	15. (c)	16. (a)	17. (b)	18. (a)	19. (d)	20. (b)
21. (a)	22. (d)	23. (b)	24. (a)	25. (b)	26. (d)	27. (b)	28. (a)	29. (d)	30. (d)
31. (d)	32. (b)	33. (b)	34. (a)	35. (a)	36. (a)	37. (d)	38. (b)	39. (a)	40. (a)
41. (b)	42. (d)	43. (d)	44. (c)	45. (c)	46. (d)	47. (a)	48. (c)	49. (d)	50. (b)
51. (c)	52. (a)	53. (a)	54. (b)	55. (d)	56. (c)	57. (a)	58. (a)	59. (d)	60. (b)
61. (c)	62. (a)	63. (d)	64. (d)	65. (a)					

Input Text Based MCQs

66. (a)	67. (c)	68. (a)	69. (a)

HINTS/EXPLANATION

1. There has to be at least one except statement.

2. The else part is executed when no exception occurs.

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

5. The finally block is always executed.

7. The try block lets you test a block of code for errors.

8. An exception occurred be the output for the followinng code because the try block will generate an error, because x is not defined.

55. The first line is read using readline(). The read() reads the rest of the lines.

57. The read() reads the first 20 characters from the beginning of the file. Newline or '\n' counts as one character.

63. "abc.txt" refers to the file to be opened here.

65. The condition of equality is line[0] == 'P'.

68. EOF for End of File and EOL for End of Line.