

1. When reading a file in Python, you must specify two items:
  1. a file name and file properties
  2. a file name and mode
  3. a file name and size
  4. a file name and data type

Title

When reading a file in Python, what information must be provided?

type

mc

Section

7.1 Reading and Writing Text Files

id

testbank-py-1-ch07-01

2. Before accessing a file, the program must:
  1. name the file
  2. read the file
  3. open the file
  4. close the file

Title

Before accessing a file, what must be done to allow access to the file?

type

mc

Section

7.1 Reading and Writing Text Files

id

testbank-py-1-ch07-02

3. After executing the following code snippet, what part is the file object?

```
infile = open("input.txt", "r")
```

1. infile
2. "input.txt"
3. "r"
4. input

Title

After executing the following code snippet, what part is the file object?

type

mc

Section

7.1 Reading and Writing Text Files

id

testbank-py-1-ch07-03

4. In the following code snippet, what does the "r" represent?

```
infile = open("input.txt", "r")
```

1. replace
2. recursive
3. random
4. read

Title

In the following code snippet, what does the "r" represent?

type

mc

Section

7.1 Reading and Writing Text Files

id

testbank-py-1-ch07-04

5. In the following code snippet, what does the "w" represent?

```
outfile = open("output.txt", "w")
```

1. wrapper
2. write
3. width
4. web

Title

In the following code snippet, what does the "w" represent?

type

mc

Section

7.1 Reading and Writing Text Files

id

testbank-py-1-ch07-05

6. In the following code snippet, what happens if the "output.txt" file does not exist?

```
outfile = open("output.txt", "w")
```

1. An error message is displayed
2. An empty file is created
3. Nothing, this statement is ignored
4. A new file is created with at least one record

Title

In the following code snippet, what happens if the "output.txt" file does not exist?

type

mc

Section

7.1 Reading and Writing Text Files

id  
testbank-py-1-ch07-06

7. In the following code snippet, what happens if the "output.txt" file already exists?

```
outfile = open("output.txt", "w")
```

1. Any new data is appended to the end
2. The existing file is emptied
3. Nothing, this statement is ignored
4. An error message is displayed

Title

In the following code snippet, what happens if the "output.txt" file already exists?

type

mc

Section

7.1 Reading and Writing Text Files

id

testbank-py-1-ch07-07

8. What method ensures that the output has been written to the disk file?

1. `commit()`

2. `write()`

3. `close()`

4. `complete()`

Title

What method ensures that the output has been written to the disk file?

type

mc

Section

7.1 Reading and Writing Text Files

id

testbank-py-1-ch07-08

9. Once a file has been opened, what method is used to read data from a file?

1. `readline()`

2. `readfile()`

3. `readln()`

4. `open()`

Title

Once a file has been opened, what method is used to read data from a file?

type

mc

Section

7.1 Reading and Writing Text Files

id

testbank-py-2-ch07-09

from

testbank-py-1-ch07-09

10. The `readline` method reads text until an end of line symbol is encountered, how is an end of line character represented?

1. `\r`

2. `\t`

3. `""`

4. `\n`

Title

The `readline` method reads text until an end of line symbol is encountered, how is an end of line character represented?

type

mc

Section

7.1 Reading and Writing Text Files

id

testbank-py-1-ch07-10

11. What is returned when the `readline` method reaches the end of the file?

1. `\r`

2. `\t`

3. ""

4. \n

Title

What is returned when the readline method reaches the end of the file?

type

mc

Section

7.1 Reading and Writing Text Files

id

testbank-py-1-ch07-11

12. When using the `readline` method, what data type is returned?

1. integer
2. float
3. character
4. string

Title

When using the readline method, what data type is returned?

type

mc

Section

7.1 Reading and Writing Text Files

id

testbank-py-1-ch07-12

13. What method(s) can be used to write to a file?

1. `write, print`

2. `out, print`

3. `write, out`

4. `post, write`

Title

What method(s) can be used to write to a file?

type  
mc  
Section  
7.1 Reading and Writing Text Files  
id  
testbank-py-1-ch07-13

14. What happens in the following code snippet?

```
infile = open("", "r")
```

1. a new file object is created
2. a run-time error occurs because the file does not exist
3. nothing, the statement is ignored
4. a default file name is used "inputfile"

Title  
What happens in the following code snippet?  
type  
mc  
Section  
7.1 Reading and Writing Text Files  
id  
testbank-py-1-ch07-14

15. Which of the following statements opens a text file for reading?

1. `infile = open("myfile.txt", "r")`
2. `infile = open("myfile.txt", "rw")`
3. `infile = open("myfile.txt", "read")`
4. `infile = open("myfile.txt", "reading")`

Title  
Which statement opens a text file for reading?  
type  
mc  
Section  
7.1 Reading and Writing Text Files  
id  
testbank-py-1-ch07-15

16. What is the name of the file object variable in the following code segment?

```
a = open("b.txt", "r")
```

1. a
2. b
3. b.txt
4. r

Title

Identify the file object variable in a code segment

type

mc

Section

7.1 Reading and Writing Text Files

id

testbank-py-1-ch07-16

17. What happens when the following code segment executes if `test.txt` does not exist?

```
infile = open("test.txt", "r")
```

1. The file `test.txt` is created as a new empty file
2. The program raises an exception
3. All attempts to read from the file return an empty string
4. All attempts to read from the file return random values

Title

What happens when opening a file that doesn't exist for reading?

type

mc

Section

7.1 Reading and Writing Text Files

id

testbank-py-1-ch07-17

18. What happens when the following code segment executes if `test.txt` does not exist?

```
outfile = open("test.txt", "w")
```

1. The file `test.txt` is created as a new empty file
2. The program ends with an exception
3. All attempts to write to the file succeed but do not save any data
4. The program continues executing until the first write statement

Title

What happens when opening a file that doesn't exist for writing?

type

mc

Section

7.1 Reading and Writing Text Files

id

testbank-py-1-ch07-18

19. Which statement is used to close the file object opened with the following statement?

```
infile = open("test.txt", "r")
```

1. `close(infile)`
2. `close("test.txt")`
3. `infile.close()`
4. `infile.close("test.txt")`

Title

Which statement correctly closes a file?

type

mc

Section

7.1 Reading and Writing Text Files

id

testbank-py-1-ch07-19

20. What is wrong with the following code snippet that is supposed to print the contents of the file twice?

```
infile = open("input.txt", "r")
for sentence in infile :
    print(sentence)
    for sentence in infile :
        print(sentence)
```

1. Python cannot iterate over the file twice without closing and reopening the file
2. A run-time error occurs because the file does not exist
3. Nothing, the code prints the contents two times
4. The program cannot use the variable `sentence` twice

Title

What is wrong with a code snippet that is supposed to print the contents of the file twice?

type

mc

Section

7.2 Text Input and Output

id

testbank-py-1-ch07-20

21. What is output by the following code segment when `input.txt` contains the following words: `apple`, `pear`, and `banana` stored one per line?

```
infile = open("input.txt", "r")
for word in infile :
    print(word)
    1. apple pear banana
    2. apple
    pear
    banana
```

```
3. apple
   pear
   banana
4. apple, pear, banana
```

Title

What is output by a code segment that reads data from a file?

type

mc

Section

7.2 Text Input and Output

id

testbank-py-2-ch07-21

from

testbank-py-1-ch07-21

22. In the code snippet below, if the file contains the following words: apple, pear, and banana stored one per line, what would be the output?

```
infile = open("input.txt", "r")
for word in infile :
    word = word.rstrip()
    print(word)
1. apple pear banana
2. apple
   pear
   banana
3. apple

   pear

   banana
4. apple, pear, banana
```

Title

In the code snippet below, if the file contains the following words: apple, pear, and banana stored one per line, what would be the output?

type

mc

Section

7.2 Text Input and Output

id

testbank-py-1-ch07-22

23. In the code snippet below, if the file contains the following words: Monday! Tuesday. Wednesday? stored one per line, what would be the output?

```
infile = open("input.txt", "r")
for word in infile :
    word = word.rstrip("!.\\n")
    print(word)
1. Monday
   Tuesday
   Wednesday
2. Monday
```

Tuesday

Wednesday

3. Monday!

Tuesday.

Wednesday

4. Monday

Tuesday

Wednesday?

Title

In the code snippet below, if the file contains the following words: Monday! Tuesday. Wednesday? stored one per line, what would be the output?

type

mc

Section

7.2 Text Input and Output

id

testbank-py-1-ch07-23

24. In the code snippet below, if the file contains the following words: Monday! Tuesday. Wednesday? stored one per line, what would be the output?

```
infile = open("input.txt", "r")
```

```
for word in infile :
```

```
    word = word.lstrip("!.")
```

```
    print(word)
```

```
    1. Monday
```

```
    Tuesday
```

```
    Wednesday
```

```
    2. Monday
```

```
Tuesday
```

```
Wednesday
```

```
3. Monday!
```

```
Tuesday.
```

```
Wednesday?
```

```
4.
```

```
Monday
```

```
Tuesday
```

```
Wednesday?
```

Title

What is output by a code segment that reads data from a file?

type

mc

Section

7.2 Text Input and Output

id

testbank-py-2-ch07-24

from

testbank-py-1-ch07-24

25. Which of the following methods strips specified punctuation from the front or end of each string (s)?

1. `s.lstrip("!.?;:")`

2. `s.rstrip("!.?;:")`

3. `s.strip("!.?;:")`

4. `s.strip()`

Title

Which of the following methods strips specified punctuation from the front or end of each string (s)?

type

mc

Section

7.2 Text Input and Output

id

testbank-py-1-ch07-25

26. Which of the following methods provides a way to split the contents of the string `sentence` into a list of individual words?

1. `sentence.split()`

2. `sentence.strip()`

3. `sentence.splice()`

4. `sentence.separate()`

Title

Which statement splits a string into a list of individual words?

type

mc

Section

7.2 Text Input and Output

id

testbank-py-2-ch07-26

from

testbank-py-1-ch07-26

27. What will be stored in `substrings` after the following code snippet has run?

```
states = "Michigan,Maine,Minnesota,Montana,Mississippi"
```

```
substrings = states.split(",")
```

1. "Michigan,Maine,Minnesota,Montana,Mississippi"
2. ["Michigan", "Maine", "Minnesota", "Montana", "Mississippi"]
3. ["Michigan,Maine,Minnesota,Montana,Mississippi"]
4. "Michigan"

Title

What is returned when the split method is called?

type

mc

Section

7.2 Text Input and Output

id

testbank-py-2-ch07-27

from

testbank-py-1-ch07-27

28. What is stored in `substrings` after the following code snippet has run?

```
=states = "Michigan,Maine,Minnesota,,Montana,Mississippi"
substrings = states.split(",")
```

1. "Michigan,Maine,Minnesota,Montana,Mississippi"
2. ["Michigan"]
3. ["Michigan,Maine,Minnesota,Montana,Mississippi"]
4. ["Michigan", "Maine", "Minnesota", "", "Montana", "Mississippi"]

Title

What is returned when the split method is called?

type

mc

Section

7.2 Text Input and Output

id

testbank-py-2-ch07-28

from

testbank-py-1-ch07-28

29. Which statement below can be used to read data from a file one character at a time?

1. `inputFile.get(1)`

2. `inputFile.read(1)`

3. `inputFile.split(1)`

```
4. inputFile.open(1)
```

Title

Which statement below can be used to read data from a file one character at a time?

type

mc

Section

7.2 Text Input and Output

id

testbank-py-1-ch07-29

30. Given a text file `quote.txt` that contains this sentence:

```
Home computers are being called upon to perform many new
functions, including the consumption of homework formerly eaten by
the dog. ~Doug Larson
```

What will be printed by this code snippet?

```
letterCounts = [0] * 26
inputFile = open("quote.txt", "r")
char = inputFile.read(1)
while char != "":
    char = char.upper()
    if char >= "A" and char <= "Z":
        code = ord(char) - ord("A")
        letterCounts[code] = letterCounts[code] + 1
    char = inputFile.read(1)
inputFile.close()
print(letterCounts[1])
```

1. 0
2. 2
3. 1
4. 5

Title

What is printed by a code snippet that reads a text file using the read method?

type

mc

Section

7.2 Text Input and Output

id

testbank-py-2-ch07-30

from

testbank-py-1-ch07-30

31. Given a text file `quote.txt` that contains this sentence:

```
Home computers are being called upon to perform many new
functions, including the consumption of homework formerly eaten by
```

the dog. ~Doug Larson

What is the result of this code snippet:

```
inputFile = open("quote.txt", "r")
char = inputFile.read(1)
while char != "" :
    print(char)
    char = inputFile.read(1)
inputFile.close()
1. Home computers are being called upon to perform many new
functions, including the consumption of homework formerly eaten by
the dog. ~Doug Larson
```

2. each character of the quote is printed on a separate line

3. H

4. o

Title

What is the result of the code snippet that reads a text file using the read method?

type

mc

Section

7.2 Text Input and Output

id

testbank-py-1-ch07-31

32. Suppose the input file contains a person's last name and age on the same line separated by a space. Which statements extract this information correctly?

```
1. record = inputFile.readline()
data = record.split()
name = data[0].rstrip()
age = int(data[1])
```

2.

```
record = inputFile.readline()
data = record.split()
age = int(data[0])
name = data[1].rstrip()
```

3.

```
record = inputFile.read(1)
data = record.split()
name = data[0].rstrip()
age = int(data[1])
```

4.

```
record = inputFile.readline()
name = record[0].rstrip()
age = int(record[1])
```

Title

Which statement reads data from a file correctly?

type

mc

Section

7.2 Text Input and Output

id

testbank-py-2-ch07-32

from

testbank-py-1-ch07-32

33. Assume that outfile is a file object that has been opened for writing. Which of the following code segments stores

```
Hello  
World
```

in the file?

1. outfile.print("Hello\n", "World\n")
2. outfile.write("Hello\n", "World\n")
3. outfile.print("Hello\nWorld\n")
4. outfile.write("Hello\nWorld\n")

Title

Which statement saves "Hello World" in a file?

type

mc

Section

7.2 Text Input and Output

id

testbank-py-1-ch07-33

34. The following code segment is supposed to read all of the lines from test.txt and save them in copy.txt.

```
infile = open("test.txt", "r")  
outfile = open("copy.txt", "w")  
  
line = infile.readline()  
  
_____  
    outfile.write(line)  
    line = infile.readline()  
  
infile.close()  
outfile.close()
```

Which line of code should be placed in the blank to achieve this goal?

1. while line == "" :

2. `while line != "" :`
3. `while line == "\n" :`
4. `while line != "\n" :`

Title

Complete the code for copying a file

type

mc

Section

7.2 Text Input and Output

id

testbank-py-1-ch07-34

35. The following code segment is supposed to read and display the contents of a file. What problem does it have in its current form?

```
infile = open("test.txt", "r")

line = infile.readline()
while line != "" :
    print(line)
    line = infile.readline()

infile.close()
```

1. The program displays all of the lines, except for the first one
2. The program displays all of the lines, except for the last one
3. The program displays the contents of the file, but it is double spaced
4. There is no problem -- the program works as desired

Title

What is wrong with the program for displaying a text file?

type

mc

Section

7.2 Text Input and Output

id

testbank-py-1-ch07-35

36. The following program opens `test.txt` and displays its contents. If nothing is placed in the blank, then the contents of the file is displayed double spaced. What should be placed on the blank so that the contents of the file is displayed without the extra blank lines?

```
infile = open("test.txt", "r")

line = infile.readline()
while line != "" :
    _____
    print(line)
    line = infile.readline()

infile.close()
```

```
1. infile = infile.lstrip()
2. infile = infile.rstrip()
3. line = line.lstrip()
4. line = line.rstrip()
```

Title

What code should be added so that extra blank lines are removed?

type

mc

Section

7.2 Text Input and Output

id

testbank-py-1-ch07-36

37. Which of the following code segments will display all of the lines in the file object named `infile`, assuming that it has successfully been opened for reading?

```
1. for infile in line :
    print(line)
```

```
2. for line in infile :
    print(line)
```

3.

```
while infile in line :
    print(line)
```

```
4. while line in infile :
    print(line)
```

Title

Which code segment displays all of the lines in a file?

type

mc

Section

7.2 Text Input and Output

id

testbank-py-2-ch07-37

from

testbank-py-1-ch07-37

38. Consider the following code segment:

```
line = "hello world!"
parts = line.split()
print(parts)
```

What is displayed when this code segment runs?

1. `hello,world!`
2. `hello world!`
3. `"hello", "world!"`
4. `["hello", "world!"]`

Title

Trace code involving the split method

type

mc

Section

7.2 Text Input and Output

id

testbank-py-1-ch07-38

39. Assume that a line has just been read from a file and stored in a variable named `line`. The line contains several words, each of which is separated by one or more spaces. Which of the following statements will store a list of all the words in `wordList`?

1. `wordList = line.split()`
2. `wordList = line.splitlines()`
3. `wordList = line.strip()`
4. `wordList = line.words()`

Title

Which statement produces a list of words from a line in a file?

type

mc

Section

7.2 Text Input and Output

id

testbank-py-1-ch07-39

40. What type of value is returned by the expression `ord("A")`?

1. Integer
2. Floating Point Number
3. String
4. List

Title

What type of value is returned by the `ord` function?

type

mc

Section

7.2 Text Input and Output

id

testbank-py-1-ch07-40

41. How is a program executed or started from the command line?

1. Selecting "Run" in the development environment
2. Clicking an icon
3. Typing the name of the program at the prompt in a terminal window
4. Python programs cannot be started from the command line

Title

How is a program executed or started from the command line?

type

mc

Section

7.3 Command Line Arguments

id

testbank-py-2-ch07-41

from

testbank-py-1-ch07-41

42. Given the following command line, where are the arguments stored?

```
python program.py -v input.dat
```

1. user defined list
2. argv list
3. input.dat
4. program.py

Title

Given the following command line, where are the arguments stored?

type

mc

Section

7.3 Command Line Arguments

id

testbank-py-1-ch07-42

43. If a program is invoked with `python program.py -r input.dat output.dat`, what are the elements of `argv`?

1. `argv[0]: "-r input.dat output.dat"`

2. `argv[1]: "-r"`  
`argv[2]: "input.dat"`  
`argv[3]: "output.dat"`

- 3.

```
argv[0]: "program.py"  
argv[1]: "-r"  
argv[2]: "input.dat"  
argv[3]: "output.dat"
```

4.

```
argv[0]: "-r"  
argv[1]: "input.dat"  
argv[2]: "output.dat"
```

Title

What elements will be stored in argv for a given program invocation?

type

mc

Section

7.3 Command Line Arguments

id

testbank-py-2-ch07-43

from

testbank-py-1-ch07-43

44. Assume that your program is started with the following command:

```
python myProgram.py -z 100
```

What will be displayed by the following statement?

```
print(sys.argv[0])
```

1. python
2. myProgram.py
3. -z
4. 100

Title

Trace the code involving command line arguments

type

mc

Section

7.3 Command Line Arguments

id

testbank-py-1-ch07-44

45. Assume that your program is started with the following command:

```
python myProgram.py the quick brown fox
```

What will be displayed by the following statement?

```
print(sys.argv[5])
```

1. brown
2. fox

3. A blank line
4. The program will raise an exception

Title

Trace the code involving several command line arguments

type

mc

Section

7.3 Command Line Arguments

id

testbank-py-1-ch07-45

46. Which statement must appear in a program before the command line arguments can be accessed?

1. `from commandline import argv`
2. `from argv import commandline`
3. `from sys import argv`
4. `from argv import sys`

Title

Which statement must appear in a program before the command line arguments can be accessed?

type

mc

Section

7.3 Command Line Arguments

id

testbank-py-1-ch07-46

47. What modes are used to read and write a binary file?

1. `"r", "w"`

2. `"-r", "-w"`

3. `"rb", "wb"`

4. `"-rb", "-wb"`

Title

What modes are used to read and write a binary file?

type

mc

Section

7.4 Binary Files and Random Access (Optional)

id

testbank-py-1-ch07-47

48. If a Byte consists of 8 Bits, what is the min and max values of one Byte?

1. 0 - 255

2. 1 - 256

3. 0 - 256

4. 1 - 255

Title

If a Byte consists of 8 Bits, what is the min and max values of one Byte?

type

mc

Section

7.4 Binary Files and Random Access (Optional)

id

testbank-py-1-ch07-48

49. What is the result of the variable `position` after the following code snippet is executed?

```
inFile.seek(0)
inFile.seek(8, SEEK_CUR)
inFile.seek(-3, SEEK_CUR)
position = inFile.tell()
```

1. 0

2. 8

3. 5

4. -3

Title

What is the result of the variable position after the following code snippet is executed?

type

mc

Section

7.4 Binary Files and Random Access (Optional)

id

testbank-py-1-ch07-49

50. Consider the following function call for computing a linear regression:

```
result = scipy.stats.linregress(data1, data2)
```

What is stored in result when this function call completes?

1. A Boolean value indicating whether or not the regression was completed successfully.
2. A floating point number between -1 and 1 that indicates the amount of correlation between the data sets.
3. A list containing the mean and median values of both data sets.
4. A tuple containing the slope, intercept and correlation coefficient for the data sets.

Title

What is returned when computing a linear regression?

type

mc

Section

7.6 Application: Handling Input Errors

id

testbank-py-2-ch07-50

51. Which type of chart can be used to represent the relationship between three dimensional data?

1. A bar chart
2. A bubble chart
3. A histogram
4. A pie chart

Title

Which type of chart can be used to represent the relationship between three dimensional data?

type

mc

Section

7.6 Application: Handling Input Errors

id  
testbank-py-2-ch07-51

52. Which of the following statements opens a binary file for reading?

1. `inFile = open("test.dat", "r")`
2. `inFile = open("test.dat", "rb")`
3. `inFile = open("test.dat", "binary")`
4. `inFile = open("test.dat", "readbinary")`

Title  
Which statement opens a binary file for reading?  
type  
mc  
Section  
7.4 Binary Files and Random Access (Optional)  
id  
testbank-py-1-ch07-52

53. Which of the following statements moves the file marker 4 bytes earlier in a binary file with its file object stored in `inFile`?

1. `inFile.seek(-4, SEEK_CUR)`
2. `inFile.seek(4, SEEK_CUR)`
3. `inFile.reverse(-4, SEEK_CUR)`
4. `inFile.reverse(4, SEEK_CUR)`

Title  
Which statement moves the file marker 4 bytes earlier in a file?  
type  
mc  
Section  
7.4 Binary Files and Random Access (Optional)  
id  
testbank-py-1-ch07-53

54. Which of the following statements stores the current position of the file marker for `inFile` into `x`?

1. `x = inFile.currentPos()`
2. `x = inFile.position()`
3. `x = inFile.read()`
4. `x = inFile.tell()`

Title  
Which statement stores the current position of the file marker in a variable?  
type  
mc  
Section  
7.4 Binary Files and Random Access (Optional)  
id  
testbank-py-1-ch07-54

55. It is good programming practice to plan for possible exceptions and provide code to handle the exception. Which exception must be handled to prevent a divide by zero logic error?

1. ValueError
2. TypeError
3. ArithmeticError
4. ZeroDivisionError

Title

Which exception must be handled to prevent a divide by zero logic error?

type

mc

Section

7.5 Exception Handling

id

testbank-py-2-ch07-55

from

testbank-py-1-ch07-55

56. Consider the following code segment:

```
try :  
    inputFile = open("lyrics.txt", "r")  
    line = inputFile.readline()  
    print(line)  
  
    _____  
    print("Error")
```

What should be placed in the blank so that the program will print `Error` instead of crashing if an exception occurs while opening or reading from the file?

1. `except RuntimeError :`
2. `except EnvironmentError :`
3. `except IOError :`
4. `except IndexError :`

Title

What should be added to a code segment to prevent it from crashing when an exception occurs?

type

mc  
Section  
7.5 Exception Handling  
id  
testbank-py-2-ch07-56  
from  
testbank-py-1-ch07-56

57. Consider the following code segment:

```
try :  
    inputFile = open("lyrics.txt", "r")  
    line = inputFile.readline()  
    words = line.split()  
    print(words[len(words)])  
  
    _____  
    print("Error.")
```

The statement `print(words[len(words)])` will raise an exception. What should be placed in the blank so that this exception will be caught and the error message will be displayed?

1. `except RuntimeError :`
2. `except EnvironmentError :`
3. `except IOError :`
4. `except IndexError :`

Title  
What should be added to a code segment to prevent it from crashing when an exception occurs?  
type  
mc  
Section  
7.5 Exception Handling  
id  
testbank-py-2-ch07-57  
from  
testbank-py-1-ch07-57

58. Python's error handling process includes the `finally` clause. In the following code snippet, when is the `finally` clause executed?

```
inputFile = open("lyrics.txt", "r")
try :
    line = inputFile.readline()
    words = line.split()
    print(words)
finally :
    inputfile.close()
```

1. Only when there is an error opening the file.
2. Only when there is an error reading the file.
3. The `finally` clause is always executed in this example.
4. The `finally` clause is never executed in this example.

Title

When is the `finally` clause executed?

type

mc

Section

7.5 Exception Handling

id

testbank-py-2-ch07-58

from

testbank-py-1-ch07-58

59. Consider the following code segment:

```
print("W", end="")
try :
    inFile = open("test.txt", "r")
    line = inFile.readline()
    value = int(line)
    print("X", end="")

except IOError :
    print("Y", end="")

except ValueError :
    print("Z", end="")
```

What output is generated when this program runs if `test.txt` is opened successfully and its first line contains the number 5?

1. W
2. WX
3. WXY
4. WXYZ

Title

Trace code involving files and exceptions

type

mc  
Section  
7.5 Exception Handling  
id  
testbank-py-1-ch07-59

60. Consider the following code segment:

```
print("W", end="")
try :
    inFile = open("test.txt", "r")
    line = inFile.readline()
    value = int(line)
    print("X", end="")

except IOError :
    print("Y", end="")

except ValueError :
    print("Z", end="")
```

What output is generated when this program runs if `test.txt` is not opened successfully?

1. WX
2. WY
3. WZ
4. WXY

Title  
Trace code involving files and exceptions  
type  
mc  
Section  
7.5 Exception Handling  
id  
testbank-py-1-ch07-60

61. Consider the following code segment:

```
print("W", end="")
try :
    inFile = open("test.txt", "r")
    line = inFile.readline()
    value = int(line)
    print("X", end="")

except IOError :
    print("Y", end="")

except ValueError :
    print("Z", end="")
```

What output is generated when this program runs if `test.txt` is opened successfully and its first line contains the `hello world`?

1. `WX`
2. `WY`
3. `WZ`
4. `WXZ`

Title

Trace code involving files and exceptions

type

mc

Section

7.5 Exception Handling

id

testbank-py-1-ch07-61

62. What exception is raised by the following code segment?

```
data = ["A", "B", "C", "D"]
print(data[4])
```

1. `IndexError`
2. `IOError`
3. `ValueError`
4. No exception is raised by the code segment

Title

What exception is raised by this code segment?

type

mc

Section

7.5 Exception Handling

id

testbank-py-1-ch07-62

63. What code should be added to the end of the following code segment to ensure that `inFile` is always closed, even if an exception is thrown in the code represented by `...`?

```
inFile = open("test.txt", "r")
try :
    line = inFile.readline()
    . . .
    1. inFile.close()

    2. always :
       inFile.close()
```

3.

```
ensure :  
    inFile.close()
```

4.

```
finally :  
    inFile.close()
```

Title

What code ensures that a file is always closed, even if an exception occurs?

type

mc

Section

7.5 Exception Handling

id

testbank-py-1-ch07-63

64. Consider the following code segment:

```
line = input()  
try:  
    ...  
    if line == "":  
        _____  
    ...  
except RuntimeError:  
    print("A blank line was encountered.")
```

This code segment is supposed to print out A blank line was encountered when a blank line is entered by the user. What code should be placed in the blank so that it will accomplish this goal?

1. except RuntimeError
2. raise RuntimeError
3. RuntimeError(raise)
4. RuntimeError(except)

Title

Which statement raises an exception?

type

mc

Section

7.6 Application: Handling Input Errors

id

testbank-py-2-ch07-64

65. When your program contains logic to read one or more files, which of the following statements is NOT true about the error handling logic needed:
1. The file might not exist
  2. The file name might be too long
  3. The file might contain invalid data
  4. All files must be opened and closed prior to program termination

Title

When your program contains logic to read one or more files, which of the following statements is NOT true about the error handling logic?

type

mc

Section

7.6 Application: Handling Input Errors

id

testbank-py-1-ch07-65

66. Consider the following code segment:

```
done = False
while not done :
    try :
        filename = input("Enter the file name: ")
        inFile = open(filename, "r")

    except IOError :
        print("Error: File not found.")
```

It is supposed to keep on prompting the user for file names until the user provides the name of a file that can be opened successfully. What line of code should be placed in the blank to achieve this goal?

1. done = False
2. done = True
3. done = inFile
4. done = not inFile

Title

Complete the code segment for reading a valid file name from the user

type

mc

Section

7.6 Application: Handling Input Errors

id

testbank-py-1-ch07-66

67. Which of the following statements is NOT valid for reading from a file:

1. inputFile.readline()
2. inputFile.read()

```
3. inputFile.readline(5)
```

```
4. inputFile.read(5)
```

Title

Which of the following statements is NOT valid for reading from a file:

type

mc

Section

7.2 Text Input and Output

id

testbank-py-1-ch07-67

68. What is wrong with the following code snippet:

```
file = open("lyrics.txt", "w")
line = file.readline()
words = line.split()
print(words)
file.close()
```

1. The file has only been opened for writing, not reading.
2. The file name is invalid.
3. The program cannot run without a try/catch statement.
4. The split method has not been used correctly.

Title

What is wrong with a code segment that reads from a file?

type

mc

Section

7.2 Text Input and Output

id

testbank-py-2-ch07-68

from

testbank-py-1-ch07-68

69. How can you read from a file starting at a designated position in it?

1. Open the file for sequential access.
2. Move the *file marker* prior to a read or write operation.
3. You can't. Python only allows you to start reading at the beginning of a file.
4. Use the `readline(position)` method.

Title

How can you read a file starting at a designated position in the file?

type  
mc  
Section  
7.4 Binary Files and Random Access (Optional)  
id  
testbank-py-2-ch07-69  
from  
testbank-py-1-ch07-69

70. Consider a program that wants to read a file from the absolute path

c:\users\user1\states.dat. What statement allows you to read this file?

1. file = open("c:\\\\users\\\\user1\\\\states.dat", "r")

2. file = open("c:\users\user1\states.dat", "r")

3. file = open("../states.dat", "r")

4. file = open("c:\\users\\user1\\states.dat", "r")

Title  
What statement allows you to ready a file from its absolute path?

type

mc

Section

7.1 Reading and Writing Text Files

id

testbank-py-2-ch07-70

from

testbank-py-1-ch07-70

71. Which statement(s) writes the sentence "Today is the first day of the rest of your life" to a file with one word on each line?

1. outfile.write("Today is the first day of the rest of your life")

2. outfile.write("Today \nis \nthe \nfirst \nday \nof \nthe \nrest \nof \nyour \nlife")

3. outfile.write(Today is the first day of the rest of your life)

```
4. outfile.writeline("Today \nis \nthe \nfirst \nday \nof \nthe
\nrest \nof \nyour \nlife")
```

Title

Which statement(s) writes the sentence "Today is the first day of the rest of your life" to a file with one word on each line?

type

mc

Section

7.1 Reading and Writing Text Files

id

testbank-py-1-ch07-71

72. Suppose an input file contains a grocery list. Each line contains the item name followed by its cost. The item name and cost are separated by a comma. Which statements extract this information correctly?

```
record = inputFile.read(1)
data = record.split(',')
groceryItem = data[0].rstrip()
cost = int(data[1])
```

```
record = inputFile.readline()
data = record.split()
groceryItem = int(data[0])
cost = data[1].rstrip()
```

```
record = inputFile.readline()
data = record.split(",")
groceryItem = data[0].rstrip()
cost = float(data[1])
```

```
record = inputFile.readline()
groceryItem = record[0].rstrip()
cost = int(record[1])
```

Title

Which statements extract information from a file correctly?

type

mc

Section

## 7.2 Text Input and Output

id

testbank-py-2-ch07-72

from

testbank-py-1-ch07-72

73. Which of the following file operations is NOT valid for reading a binary file?

1. `fileName = open("input.dat", "r")`

2. `fileName.close()`

3. `fileName.write("Good Bye")`

4. `fileName = open("input.dat", "rw")`

Title

Which of the following file operations is NOT valid?

type

mc

Section

7.2 Text Input and Output

id

testbank-py-1-ch07-73

74. Which of the following is NOT a valid exception in Python?

1. `OverflowError`

2. `TryError`

3. `IOError`

4. `IndexError`

Title

Which of the following is NOT a valid exception in Python?

type

mc

Section

7.5 Handling Exceptions

id

testbank-py-2-ch07-74

from

testbank-py-1-ch07-74

75. What happens if you try to open a file for reading that doesn't exist?

1. A new file is created and opened.
2. A run-time error occurs because the file does not exist.
3. The statement is ignored.
4. An environment error occurs.

Title

What happens if you try to open a file for reading that doesn't exist?

type

mc

Section

7.1 Reading and Writing Text Files

id

testbank-py-2-ch07-75

from

testbank-py-1-ch07-75

76. What portable file format is commonly used to export data from a spreadsheet so that it can be read and processed by a Python program?

1. Comma-Separated Values (CSV)
2. Graphics Interchange Format (GIF)
3. Hypertext Markup Language (HTML)
4. Portable Spreadsheet Format (PSF)

Title

What portable file format is commonly used to export data from a spreadsheet so that it can be read and processed by a Python program?

type

mc

Section

7.2 Text Input and Output

id

testbank-py-2-ch07-76

77. Consider the following code segment:

```
from csv import writer
outfile = open("newdata.csv", "w")
csvWriter = writer(outfile)
```

What statement should be added to the end of this code segment so that it will write a row of values to the file?

1. csvWriter.writerow(1, 2, 3, 4)
2. csvWriter.writerow([1, 2, 3, 4])
3. outf.write(1, 2, 3, 4)
4. outf.write([1, 2, 3, 4])

Title

Which statement is used to write a row of data to a CSV file?

type

mc  
Section  
7.2 Text Input and Output  
id  
testbank-py-2-ch07-77

78. Which statement imports all of the functions in Python's regular expression library?

1. `from re import *`
2. `from regex import *`
3. `from reg_exp import *`
4. `from regular_expression import *`

Title  
Which statement imports Python's regular expression library?  
type  
mc  
Section  
7.2 Text Input and Output  
id  
testbank-py-2-ch07-78

79. Which character encoding standard uses sequences of between 1 and 4 bytes to represent a huge number of different characters?

1. ASCII
2. CSV
3. Extended ASCII
4. UTF-8

Title  
Which character encoding standard uses sequences of between 1 and 4 bytes to represent a huge number of different characters?  
type  
mc  
Section  
7.2 Text Input and Output  
id  
testbank-py-2-ch07-79

80. Which of the following statements should be used to open a file that might contain special characters (such as accents, Greek letters or musical symbols) for reading?

1. `inf = open("input.txt")`
2. `inf = open("input.txt", "r")`
3. `inf = open("input.txt", "r", "utf-8")`
4. `inf = open("input.txt", "r", encoding="utf-8")`

Title  
Which statement will open a file containing special characters for reading?  
type  
mc  
Section

## 7.2 Text Input and Output

id

testbank-py-2-ch07-80

81. Which of the following is **not** a function that resides in Python's `os` module?

1. `chdir`
2. `open`
3. `remove`
4. `rename`

Title

Which of the following is not a function that resides in Python's `os` module?

type

mc

Section

7.3 Command Line Arguments

id

testbank-py-2-ch07-81

82. Consider the following code segment:

```
import os

filename = input("Enter the name of a file: ")
if _____ :
    print("That file exists!")
else:
    print("That file does not exist.")
```

This code segment is supposed to print an appropriate message indicating whether or not the file specified by the user exists. What code should be placed in the blank so that the code segment performs its intended task?

1. `filename != ""`
2. `os.path.exists()`
3. `os.path.exists() == filename`
4. `os.path.exists(filename)`

Title

Which statement will check if a file exists?

type

mc

Section

7.3 Command Line Arguments

id

testbank-py-2-ch07-82

83. Which function is **not** part of the Python `statistics` module?

1. `mean`
2. `median`
3. `stdev`

#### 4. average

Title

Which function is not part of the Python statistics module?

type

mc

Section

7.6 Application: Handling Input Errors

id

testbank-py-2-ch07-83