# I/O and File Handling

## Exercises

### Week 8

Which of the following represents a Python *f-string*?

1. "Hello {}, you have logged in".format(name)
2. "Hello {name}, you have logged in"
3. f"Hello {name}, you have logged in"
4. "Hello %s, you have logged in" % name

*Answer:*

c) f”Hello {name}, you have logged in”

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Given the following definition of value, what would each of the following statements display?

value = 10.768572

print(f"Value is {value}")

*Answer:*

Value is 10.768572

print(f"Value is {value \* 10}")

*Answer:*

Value is 107.68572

print(f"Value is {value:.2f}")

*Answer:*

Value is 10.77

print(f"Value is {value:16.2f}")

*Answer:*

Value is 10.77

print(f"Value is {value:0>16.2f}")

*Answer:*

Value is 00000000010.77

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Within an *f-string* **format specifier** what does the ‘^’ alignment character signify?

*Answer:*

Within an f –string format specifier, the caret ‘^” alignment character signifies the center alignment.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Write a statement which uses the str.format() to generate the same output as the following *f-string* statement -

print(f"pi to 5 decimal places is {math.pi:.5f}")

*Answer:*

import math

print("pi to 5 decimal places is {:.5f}".format(math.pi))

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

What would the following statement display?

print("Length = {1} Width = {0}".format(10,20))

*Answer:*

Length = 20 Width = 10

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

What *exactly* would the following statement display?

print("Hello".rjust(10))

*Answer:*

Hello

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

On which older programming language is the *%-formatting* style loosely based?

*Answer:*

The %-formatting style in Python is loosely based on the printf-style

formatting used in C and C++.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Write a Python program that uses a loop and the str.rjust() method to generate the following output.

##########

#########

########

#######

######

#####

####

###

##

#

*Hint:* The program will start as follows

for n in range(10,0,-1):

line = "#" \* n

# rest of code....

*Answer:*

for n in range(10, 0, -1):

line = "#" \* n

print(line.rjust(10))

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

What is the basic element that *all* computer files contain?

*Answer:*

Bytes

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

What *function* must be called before the contents of a file can be accessed?

*Answer:*

open()

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

What *method* must be called on a file object once processing is complete?

*Answer:*

close()

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Following execution of the given statement, would the file ‘myfile.txt’ be open for *reading* or for *writing*?

f = open("myfile.txt")

*Answer:*

Writing

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Following execution of the given statement, would the file yourfile.txt be open for *reading* or for *writing*?

f2 = open("yourfile.txt", "w")

*Answer:*

Open for writing

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Following execution of the given statement, what would be the *mode of operation* applied to file gfxlib.so ?

f3 = open("gfxlib.so", "r+b")

*Answer:*

This allows both reading and writing.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

What is the difference between the two following method calls?

f.readline()

f.readlines()

*Answer:*

f.readline () reads a single line from the file and returns the line as a string whereas f.readlines () reads all the lines from the file and returns them as list of strings.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

How much of the file content would be read with the following method call?

content = f.read()

*Answer:*

It reads entire file at once.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

If the variable ‘my\_file’ referred to a text file, what would the following code do?

for next in my\_file:

print(next)

*Answer:*

If my\_file is a file object opened in read mode, this code will iterate over each line in the text file, and for each iteration it will print the content of that line.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

What is the issue with the following code? And how could it be fixed?

f = open("details.txt", "w")

total = 100

f.write(total)

f.close()

*Answer:*

The write method expects an argument in the string format but total here is an integer. To fix this, total can be converted to a string data-type and then be passed as an argument to the write method.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

What is the purpose of the file tell() method?

*Answer:*

Tell() method is used to find the current position of the cursor.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

What does the following code do?

f.seek(0)

*Answer:*

It is used to bring the cursor back to the top as specified by the argument provided to it.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Why is file handling often done using a ‘with’ statement as shown below?

with open("data.txt") as f:

lines = f.readlines()

*Answer:*

The with statement ensures that the file is automatically closed after use, including the risk of the resource leaks.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**