I/O and File Handling

Exercises

Week 8

Prior to attempting these exercises ensure you have read the lecture notes and/or viewed the video, and followed the practical. You may wish to use the Python interpreter in interactive mode to help work out the solutions to some of the questions.

Download and store this document within your own filespace, so the contents can be edited. You will be able to refer to it during the test in Week 6.

Enter your answers directly into the highlighted boxes.

For more information about the module delivery, assessment and feedback please refer to the module within the MyBeckett portal.

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

```
a) "Hello {}, you have logged in".format(name)
b) "Hello {name}, you have logged in"
c) f"Hello {name}, you have logged in"
d) "Hello %s, you have logged in" % name
```

Answer:

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:

Given the following definition of `value`,each of the following statements would display : 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:

In an f-string format specifier, the ^ alignment character signifies 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:

formatted_string = "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:

The given str.format() statement will display:
Length = 20 Width = 10

_ What *exactly* would the following statement display?

```
print("Hello".rjust(10))
```

Answer:

The following statement will display
Hello

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

The %-formatting style is loosely based on the formatting features found in the C programming language.

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:

The basic element that all computer files contain is **data.**

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

The **open()** function must be called before the contents of a file can be accessed.

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

The **close()** method must be called on a file object once processing is complete

---

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

```
f = open("myfile.txt")
```

Answer:

The file "myfile.txt" would be open for reading after the given statement is executed.

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

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

Answer:

The file "yourfile.txt" would be open for writing after the given statement is executed.

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:

The mode of operation applied to the file "gfxlib.so" would be read and write in binary mode ("r+b") after the given statement is executed.

_ What is the difference between the two following method calls?

```
f.readline()
f.readlines()
```

Answer:

`f.readline()` reads a single line from the file,

`f.readlines()` reads all lines from the file and returns them as a list of strings.

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

```
f.read()
```

Answer:

The content of the file will be read into the variable content when the method call content = f.read() is used.

_ If the variable ‘`my_file`’ referred to a text file, what would the following code do?

```
for next in my_file:
     print(next)
```

Answer:

The code publishes each line to the console as iterates through the text file's lines that are referenced by the variable my_file.

_ 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 problem lies in the fact that the variable total is an integer, yet the write() method expects a string. To fix it, use str(total) to convert total to a string before putting it to the file.

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

Answer:

The file cursor's current byte position within the file can be found using the file handling tell() method.

_ What does the following code do?

```
f.seek(0)
```

Answer:

The file cursor is moved to the beginning of the file (offset 0) by the code f.seek(0). It makes it possible for later read or write operations to begin at the beginning of the file by setting the file pointer to the beginning.

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

```
with open("data.txt") as f:
     lines = f.readlines()
```

Answer:

When handling files, using a with statement makes that the file opens and closes automatically, simplifies the code, and improves error handling.

Exercises are complete

Save this logbook with your answers. Then ask your tutor to check your responses to each question.