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:

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 = 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 000000000010.77

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

Answer:

The ^ alignment character centers the value within the specified width.

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:

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:

C programming

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:

data

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:

reading

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

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

Answer:

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:

The file gfxlib.so would be opened in read and write binary mode

---

What is the difference between the two following method calls?

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

Answer:

f.readline() reads the next line from the file
f.readlines() reads all the lines in the file

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

```
content = f.read()
```

Answer:

f.read() reads the entire content of the file.

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

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

Answer:

This code would iterate through each line in the my_file text file and print each 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:

f = open("details.txt", "w")
total = 100
f.write(str(total))
f.close()

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

Answer:

The tell() method returns the current position of the file pointer and indicates where the next read or write will occur in the file.

What does the following code do?

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

Answer:

seek(0) moves the file pointer back to the beginning of the file.

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 once the block of code is executed, even if an error occurs.

Exercises are complete

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