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

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 0000000000010.77

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

Answer:

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:

Print(f"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 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:

Binary or text data

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

Answer:

open() function

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:

Open for reading mode.

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 'myfile.txt' would be open for writing mode only.

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" after the given statement is executed would be reading and writing in binary mode (`"r+b"`).

What is the difference between the two following method calls?

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

Answer:

The `f.readline()` method reads a single line from the file, whereas the `f.readlines()` method reads all lines from the file and returns them as a list.

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

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

Answer:

The `content = f.read()` method call would read 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:

The given code would iterate through each line in the text file referenced by the variable `my_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 in Python is used to get the current position (in bytes) of the file cursor within an open file.

What does the following code do?

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

Answer:

Sets the file cursor position 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:

The 'with' statement is used for file handling to ensure automatic closure of the file, providing better readability and reducing the risk of resource leaks.

Exercises are complete

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