# Introduction to Programming

## Lab Worksheet

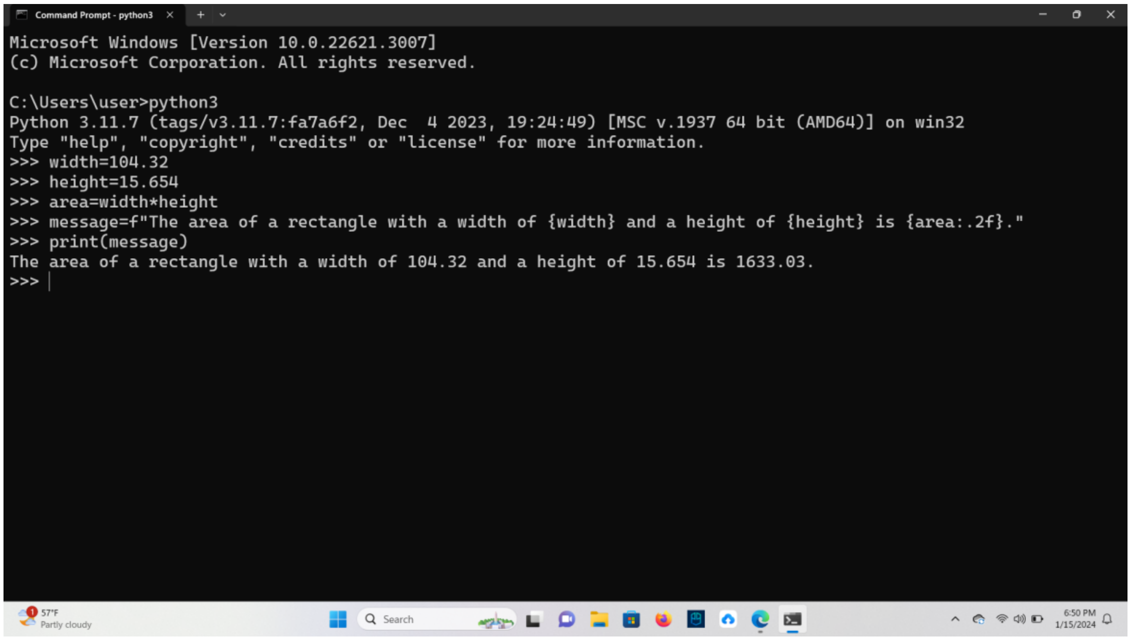
Week 8

# Basics of I/O

## Formatting with ‘f-strings’

**TASK**: Write some code that uses an f-string to calculate then display a message stating, “The area of a rectangle with a width of 104.32 and a height of 15.654 is ….”. Showing the correct answer at the end of the message.

**Ans**

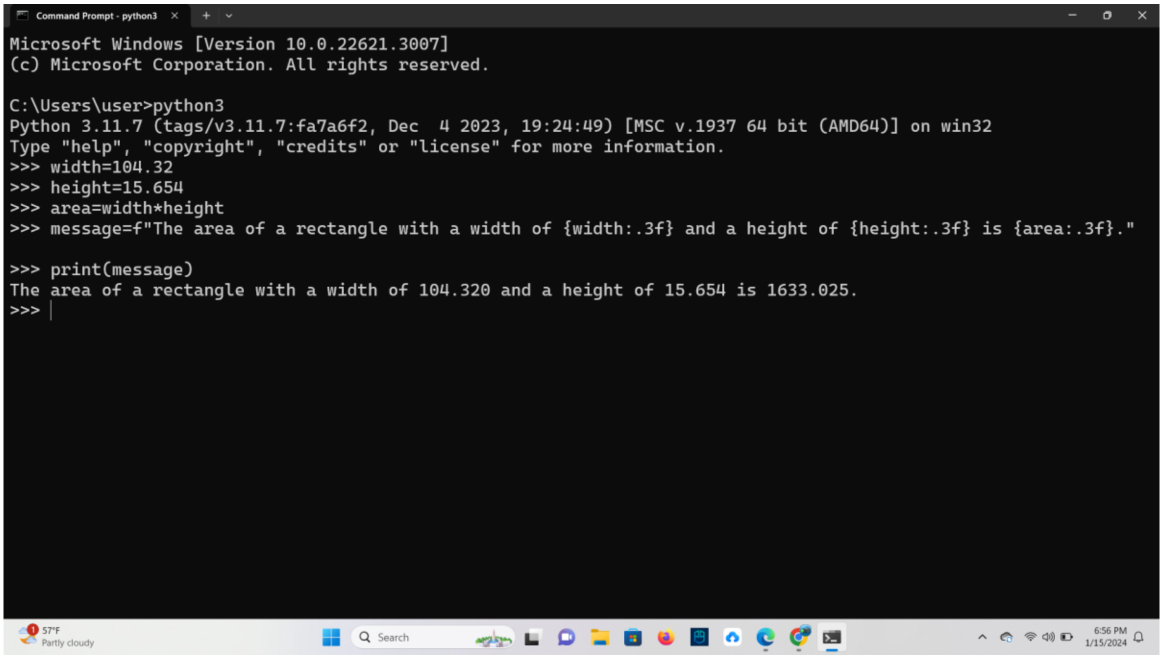


```
Microsoft Windows [Version 10.0.22621.3007]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>python3
Python 3.11.7 (tags/v3.11.7:fa7a6f2, Dec  4 2023, 19:24:49) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> width=104.32
>>> height=15.654
>>> area=width*height
>>> message=f"The area of a rectangle with a width of {width} and a height of {height} is {area:.2f}."
>>> print(message)
The area of a rectangle with a width of 104.32 and a height of 15.654 is 1633.03.
>>>
```

# Format Specifiers

**TASK**: Rewrite your earlier code that displayed the area of a rectangle but include a format specifier that limits the displayed result to three decimal places.

### Ans

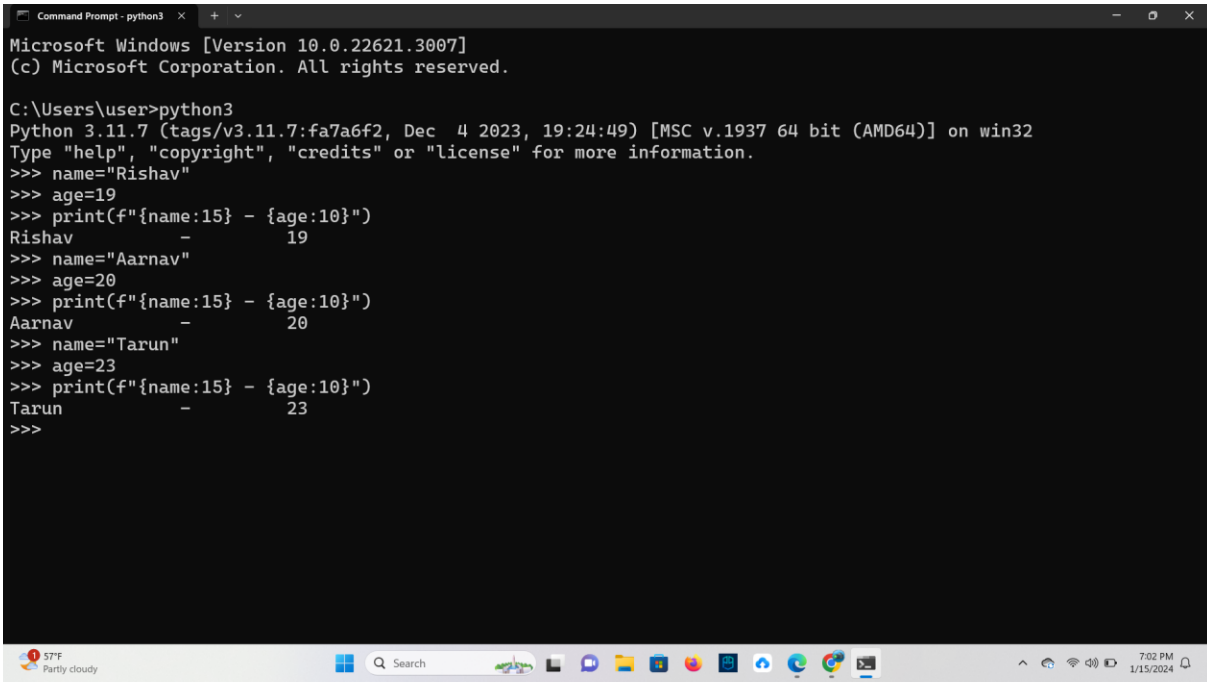


```
Microsoft Windows [Version 10.0.22621.3007]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>python3
Python 3.11.7 (tags/v3.11.7:fa7a6f2, Dec  4 2023, 19:24:49) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> width=104.32
>>> height=15.654
>>> area=width*height
>>> message=f"The area of a rectangle with a width of {width:.3f} and a height of {height:.3f} is {area:.3f}."

>>> print(message)
The area of a rectangle with a width of 104.320 and a height of 15.654 is 1633.025.
>>> |
```

**TASK**: Try setting the name and age variables to different values and executing the above print () statement multiple times. Notice the alignment and column width enforced due to the print specifier.

print(f"{name:15} - {age:10}")

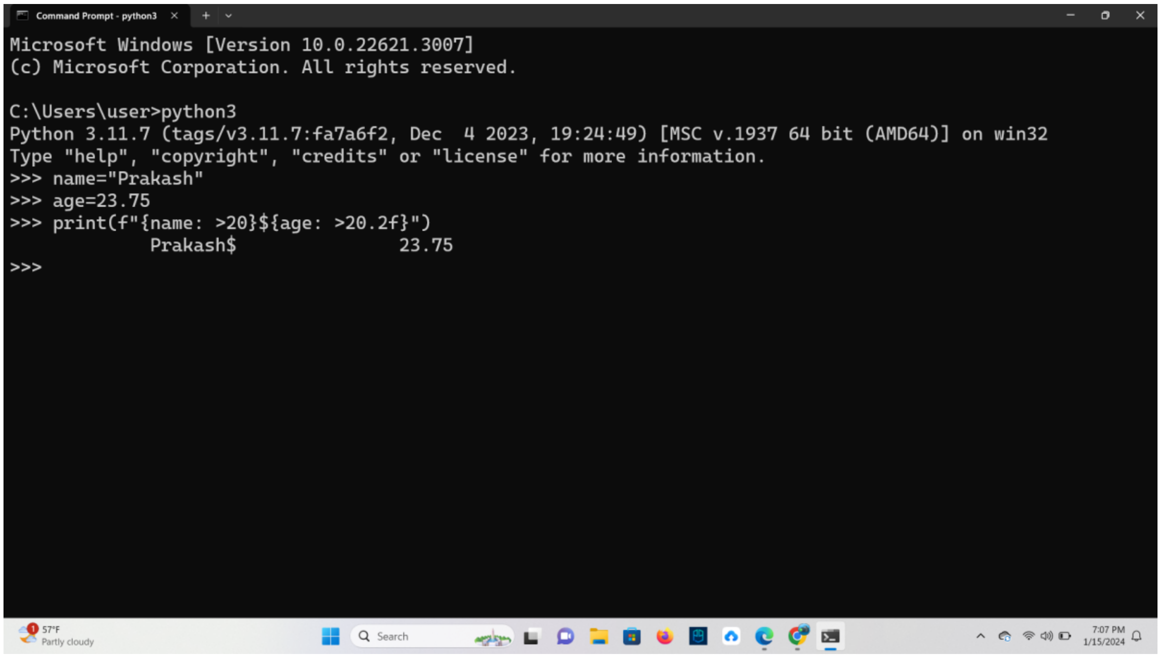
**Ans**



```
Microsoft Windows [Version 10.0.22621.3007]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>python3
Python 3.11.7 (tags/v3.11.7:fa7a6f2, Dec  4 2023, 19:24:49) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> name="Rishav"
>>> age=19
>>> print(f"{name:15} - {age:10}")
Rishav          -         19
>>> name="Aarnav"
>>> age=20
>>> print(f"{name:15} - {age:10}")
Aarnav          -         20
>>> name="Tarun"
>>> age=23
>>> print(f"{name:15} - {age:10}")
Tarun           -         23
>>>
```

**TASK**: Write a print () statement that displays the name and age values, with a column width of 20 for each, both right aligned, and with the age being shown to two decimal places. The fill character should be a dollar symbol $.

**Ans**



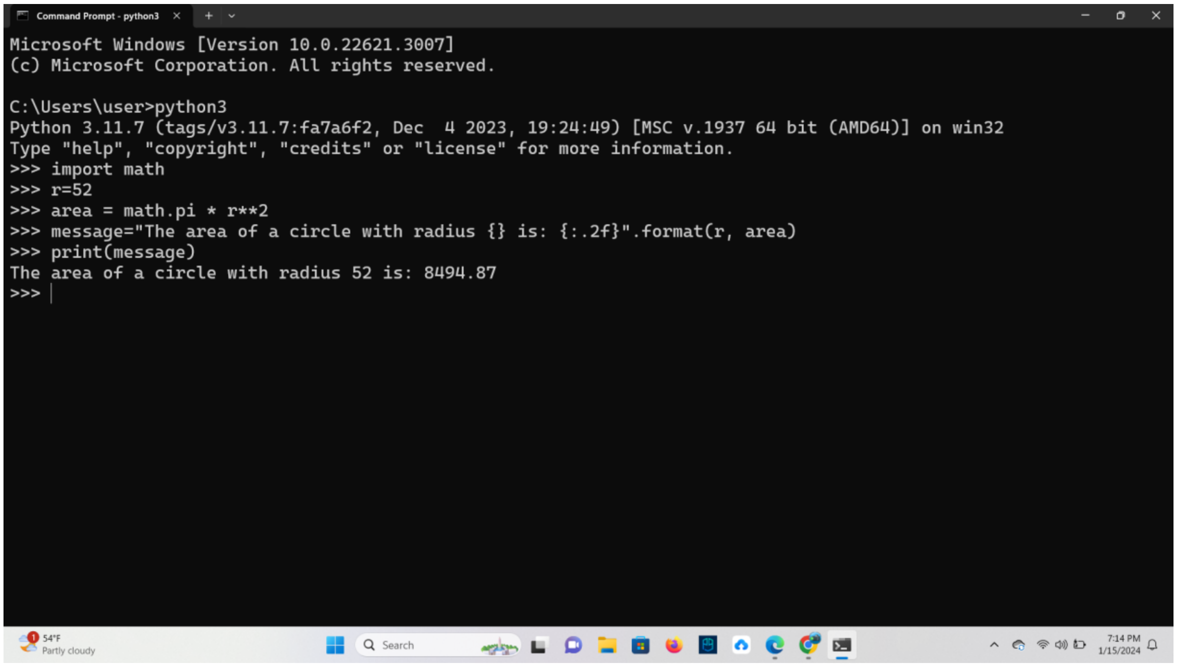
```
Microsoft Windows [Version 10.0.22621.3007]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>python3
Python 3.11.7 (tags/v3.11.7:fa7a6f2, Dec  4 2023, 19:24:49) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> name="Prakash"
>>> age=23.75
>>> print(f"{name: >20}${age: >20.2f}")
             Prakash$               23.75
>>>
```

# Formatting with str. format ()

**TASK**: Write some code that uses the str. format () method to display the area of a circle with a radius specified by the variable r. Use a keyword replacement field called area to identify the calculated area and refer to this when passing the value to the str. format () method. The output should look like the following, in the case where r is set to 52.

A circle with radius 52 has an area of 8494.8665353068

**Ans**

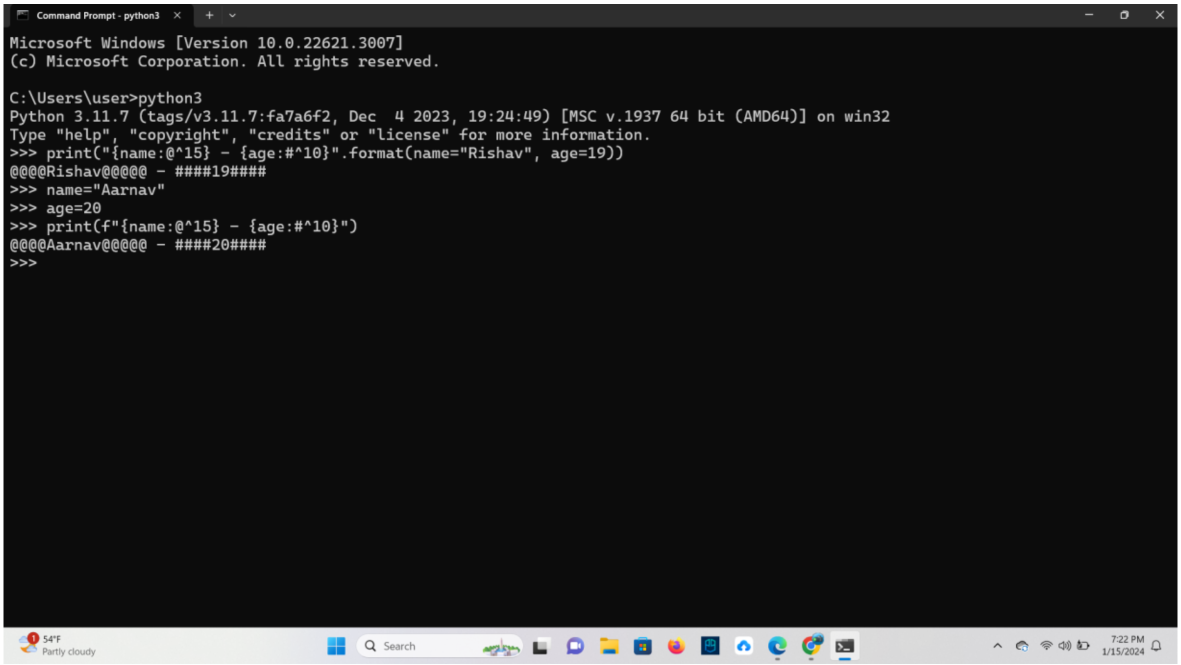


```
Microsoft Windows [Version 10.0.22621.3007]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>python3
Python 3.11.7 (tags/v3.11.7:fa7a6f2, Dec  4 2023, 19:24:49) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import math
>>> r=52
>>> area = math.pi * r**2
>>> message="The area of a circle with radius {} is: {:.2f}".format(r, area)
>>> print(message)
The area of a circle with radius 52 is: 8494.87
>>> 
```

**TASK**: Convert the f-string based statement below into an equivalent that uses the str. format () method to generate the same output.

print (f"{name: @^15} - {age: #^10}")

**Ans**



```
Microsoft Windows [Version 10.0.22621.3007]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>python3
Python 3.11.7 (tags/v3.11.7:fa7a6f2, Dec  4 2023, 19:24:49) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print("{name:@^15} - {age:#^10}".format(name="Rishav", age=19))
@@@@Rishav@@@@@ - ####19####
>>> name="Aarnav"
>>> age=20
>>> print(f"{name:@^15} - {age:#^10}")
@@@@Aarnav@@@@@ - ####20####
>>>
```

## Alternative Formatting Approaches

The Python language originally used a formatting method loosely based on the techniques used in the ‘C’ programming language.

This is sometimes referred to as %-formatting. Although superseded by f-strings and str. Format () it is still advisable to be aware of this approach, even though it probably should not be used in the new code. It uses a ‘%’ operator, between a string and associated values:

```
print ("Name is %s, and age is %.2f" % (name, age))
```

The ‘%’ values within the string itself represent the replacement fields and format specifier information, but the format of these is different from what we have seen available within f-strings and the str. format () method.

It also is possible to perform a certain amount of formatting without using any of the mechanisms described. A more manual approach can be taken simply by making calls to some of the string methods that are available.

Specifically, methods such as rjust (), ljust () and center () can be used to align values within a specified column width. The following two statements result in the exactly the same output.

```
print (name .rjust (15), " - ", str(age). center (10))
print(f"{name:>15} - {age: ^10}")
```

Using a manual approach is occasionally useful, but under most circumstances it is easier to use either the f-strings or the str. format () method to perform formatting. The resulting code is easier to understand, and probably easier to create!

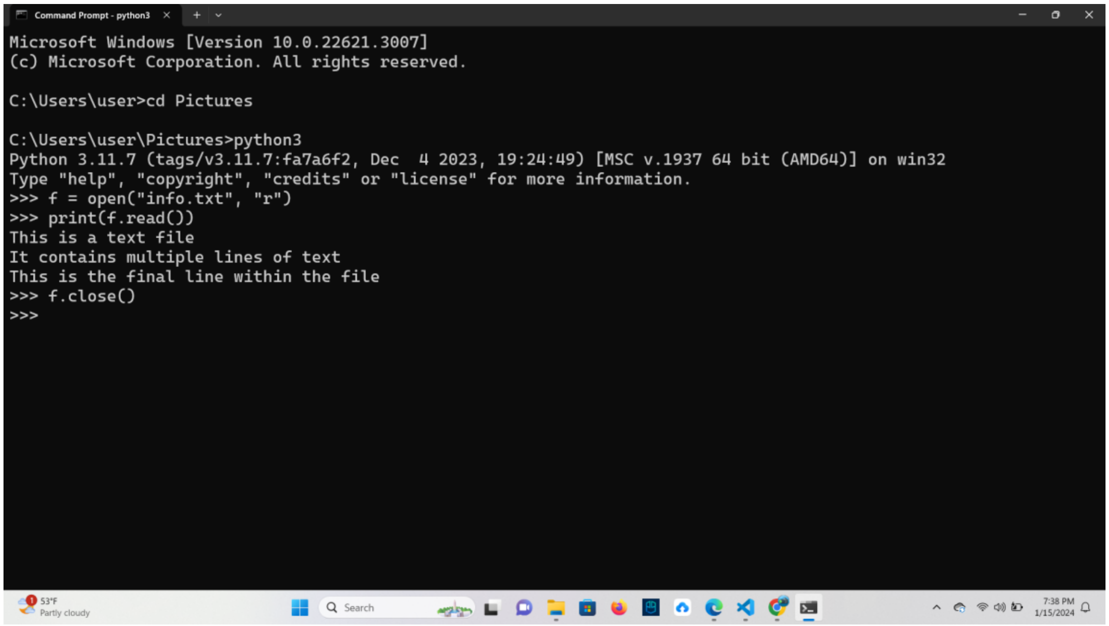
# File Handling

## Opening, Reading and Closing

**TASK**: Use a text editor to create a file called info.txt and enter the text shown below. Once the file has been created and saved, write a small program that:

1. Opens the file, 2. reads and prints the contents, 3. closes the file.

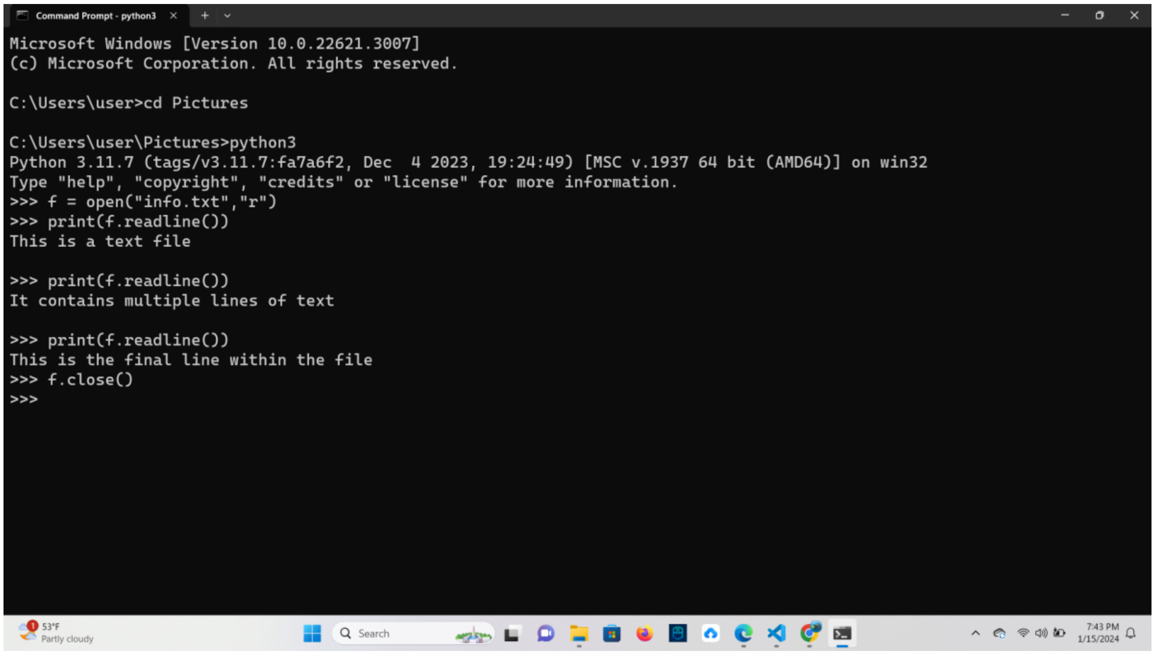
**Ans**



```
Microsoft Windows [Version 10.0.22621.3007]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>cd Pictures

C:\Users\user\Pictures>python3
Python 3.11.7 (tags/v3.11.7:fa7a6f2, Dec  4 2023, 19:24:49) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> f = open("info.txt", "r")
>>> print(f.read())
This is a text file
It contains multiple lines of text
This is the final line within the file
>>> f.close()
>>>
```

**TASK**: write a small program that opens the info.txt file, then reads and displays each of the three lines of text using calls to the readline () method. Remember to close the file once the content has been read.

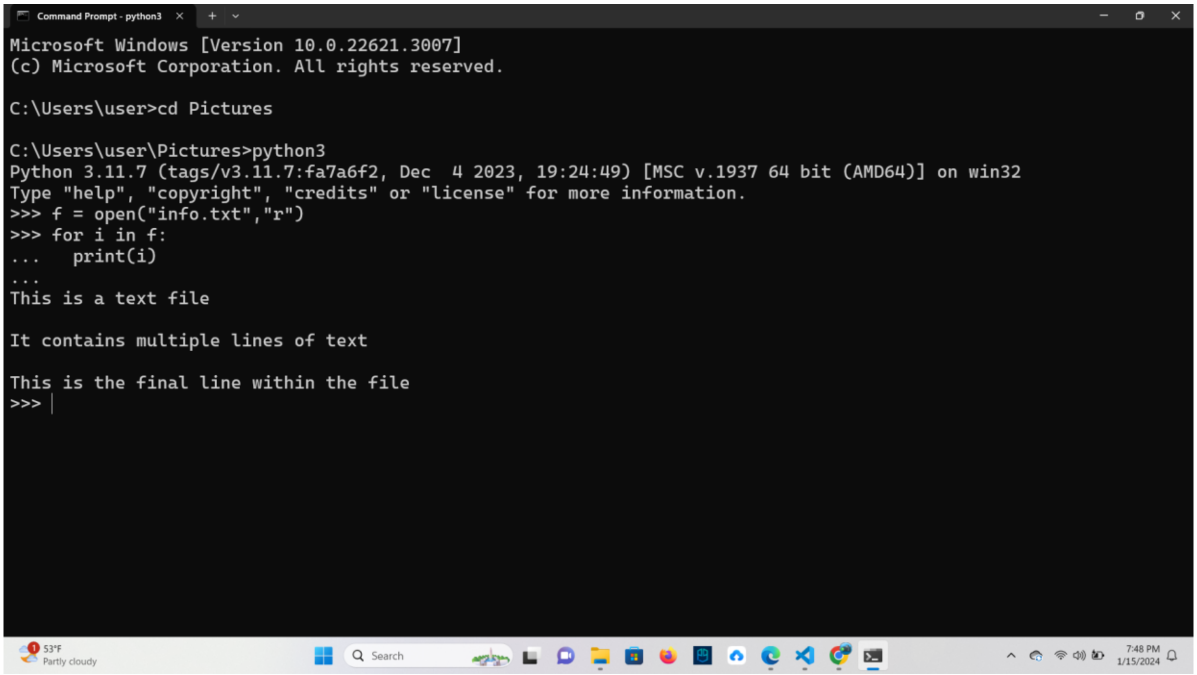
**Ans**



```
Microsoft Windows [Version 10.0.22621.3007]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>cd Pictures

C:\Users\user\Pictures>python3
Python 3.11.7 (tags/v3.11.7:fa7a6f2, Dec  4 2023, 19:24:49) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> f = open("info.txt","r")
>>> print(f.readline())
This is a text file

>>> print(f.readline())
It contains multiple lines of text

>>> print(f.readline())
This is the final line within the file
>>> f.close()
>>>
```

**TASK**: write a small program that opens the info.txt file, then reads and displays each line of text using a for...in loop. Remember to close the file once the content has been read.

**Ans**



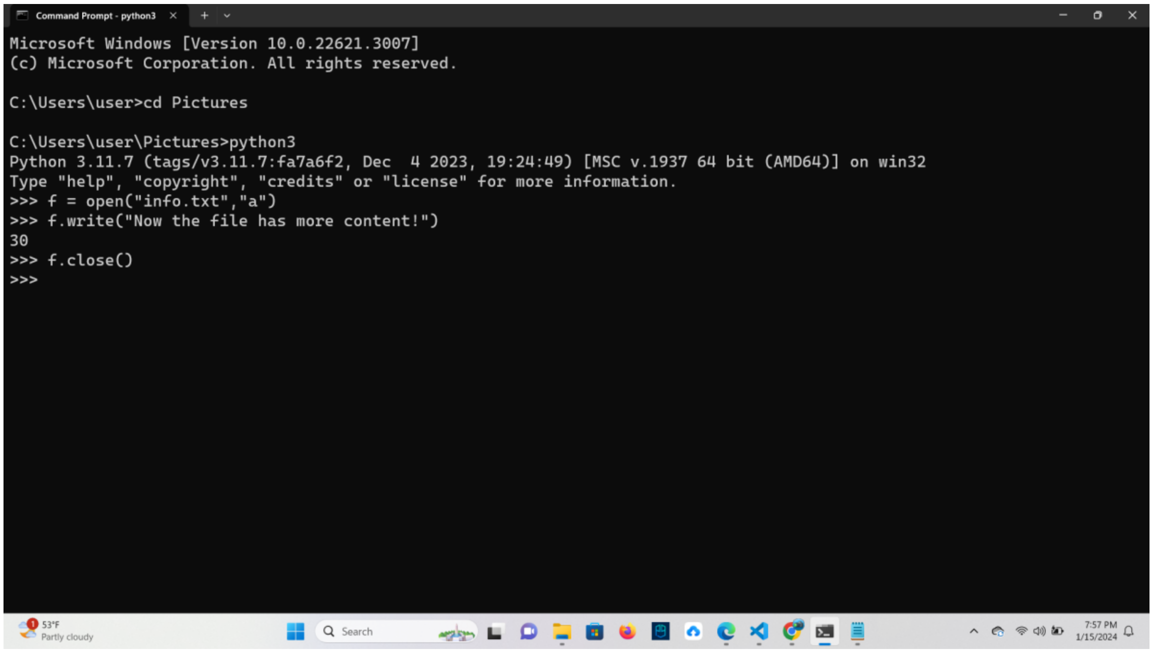
```
Microsoft Windows [Version 10.0.22621.3007]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>cd Pictures

C:\Users\user\Pictures>python3
Python 3.11.7 (tags/v3.11.7:fa7a6f2, Dec  4 2023, 19:24:49) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> f = open("info.txt","r")
>>> for i in f:
...   print(i)
...
This is a text file

It contains multiple lines of text

This is the final line within the file
>>>
```

## Modes of Operation

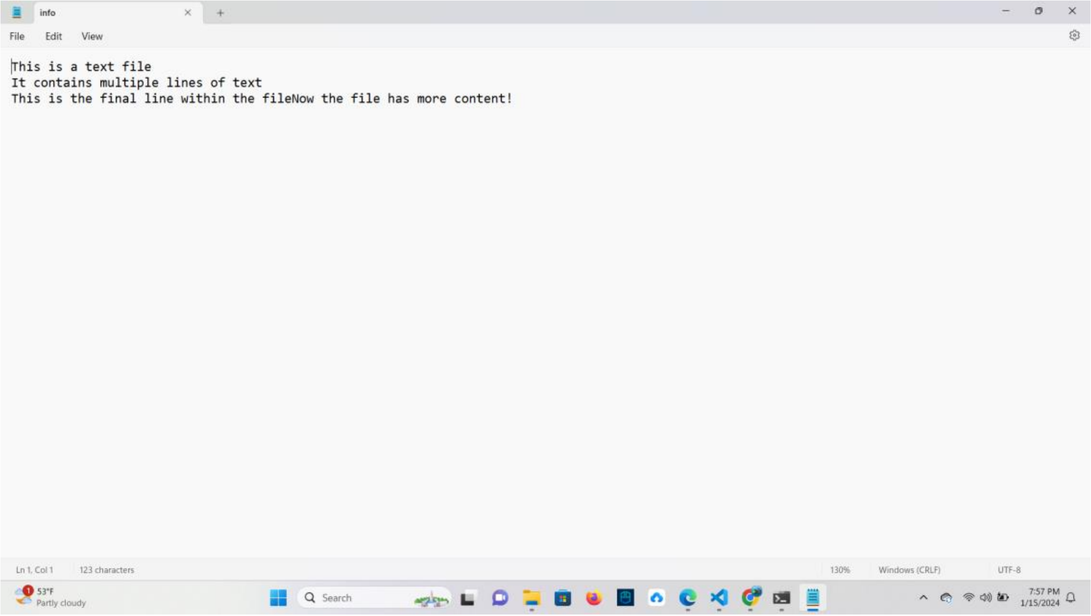
## Writing to a file

**TASK**: Write a small program that opens the "info.txt" file in append (a) mode. Use the write () method to add an extra line of text saying, "this is an extra line". Remember to close the file once the content has been read. Open the file with a text editor and examine the contents**.**

### Ans



```
Microsoft Windows [Version 10.0.22621.3007]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>cd Pictures

C:\Users\user\Pictures>python3
Python 3.11.7 (tags/v3.11.7:fa7a6f2, Dec  4 2023, 19:24:49) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> f = open("info.txt","a")
>>> f.write("Now the file has more content!")
30
>>> f.close()
>>>
```



```
This is a text file
It contains multiple lines of text
This is the final line within the fileNow the file has more content!
```

## Random Access

When a file is being processed the file object maintains a value that refers to the current position within the file. We can read the position using the tell () method and change it using the seek () method.

Using seek () allows a program to perform random access, i.e. have the ability to change read and write locations on an open file. For example, the following call would move the position back to the beginning of the file, allowing content to be re-read.

f. seek (0)

The seek () method takes a second parameter (known as 'whence') that can be used to make the provided offset relative to something other than the start of the file. This takes one of the following values - start (0), current (1), or end (2) position of the file. For example, the following call would move the current position to the end of the file.

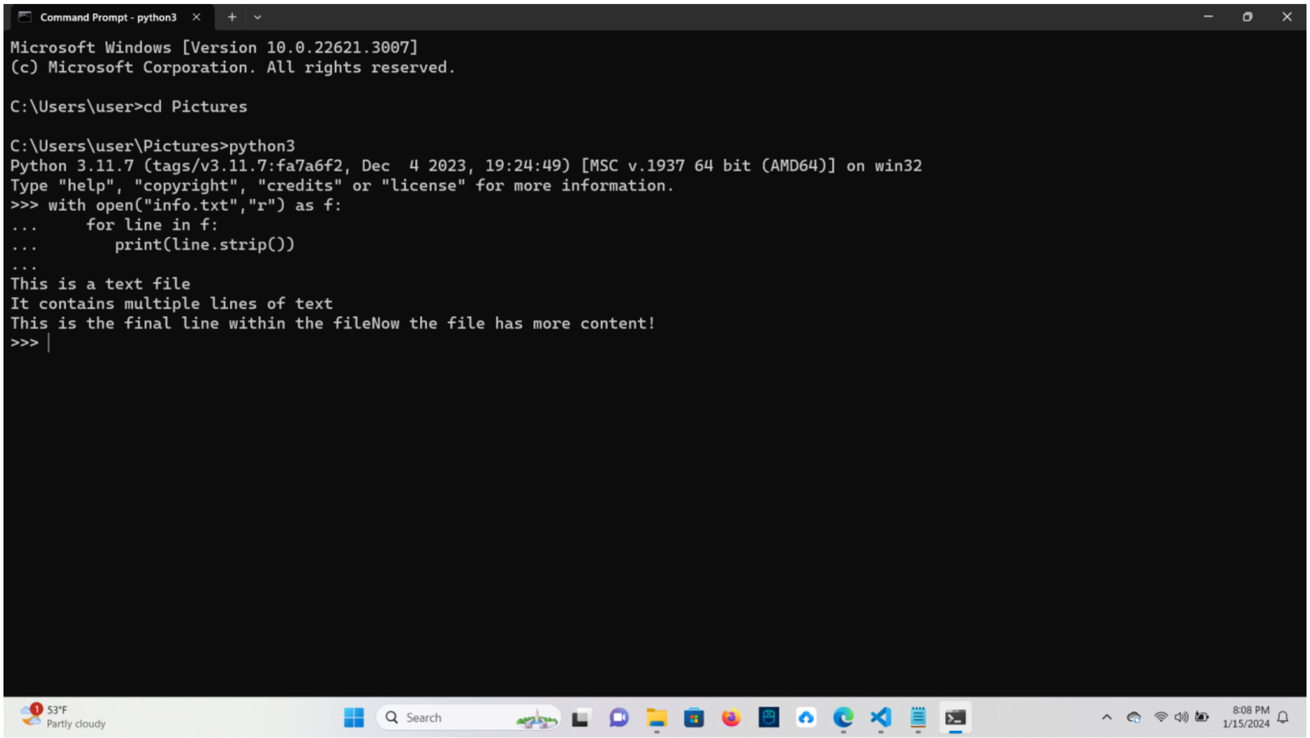
f. seek (0,2)

When a file is opened in text mode, only those values returned by tell () should be used for seeking purposes, any other offset value produces undefined behavior. Also, only a 'whence' value of 0 is allowed (the exception being when seeking to the very end of the file, as in the above example). Therefore, random access tends to be of more use when working within binary rather than text files. Also note that files opened in append (a) mode always write to the end of the file, hence seek () is of limited use in this case.

## Handling Exceptions

**TASK**: write a small program that opens the info.txt file, then reads and displays each line of text using a for...in loop. Rather than explicitly call the close () method, use the ‘with’ statement to wrap the file handling code.

**Ans**



```
Microsoft Windows [Version 10.0.22621.3007]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>cd Pictures

C:\Users\user\Pictures>python3
Python 3.11.7 (tags/v3.11.7:fa7a6f2, Dec  4 2023, 19:24:49) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> with open("info.txt","r") as f:
...     for line in f:
...         print(line.strip())
...
This is a text file
It contains multiple lines of text
This is the final line within the fileNow the file has more content!
>>>
```

## Key Terminology

**TASK**: Look at each of the phrases below and ensure you understand what each of these means. For any that you do not understand, do a little research to find a definition of each term. This research may involve looking back over these notes, or the associated lecture notes. It may also involve searching for these terms on the Internet.

● ‘f-string’

**Ans** It is a way to make it easier for users to add variables, comma and do padding.

● Format Specifier

**Ans** A substring of format that starts with a percent sign (%) and ends with one of the conversion characters.

● File modes

**Ans** The modes use for manipulating the content of file in any specific system.

● Binary Files

**Ans** The files that are used to store the containing a sequence of 1s and 0s rather than letters or numbers like ASCII text.

● Random Access

**Ans** It is the ability to access an arbitrary element of a sequence in equal time.

● Exceptions

Ans One kind of error that arises when correctly formed Python code generates an error that is an exception.