4.5 READING FROM A FILE

Python provides various methods for reading data from a file. We can read character data from a text file by using the following read methods:

a) read(): To read the entire data from the file; starts reading from the cursor up to the end of the file.

b) read(n): To read 'n' characters from the file, starting from the cursor; if the file holds fewer than 'n' characters, it will read until the end of the file.

c) readline(): To read only one line from the file; starts reading from the cursor up to, and including, the end of line character.

d) readlines(): To read all lines from the file into a list; starts reading from the cursor up to the end of the file and returns a list of lines.

Let us understand these methods with the help of suitable examples using a text file 'test.txt'.

1) **read()/read(size):** read() can be used to read specific-size strings from a file. This function also returns a string read from the file. At the end of the file, again an empty string will be returned.

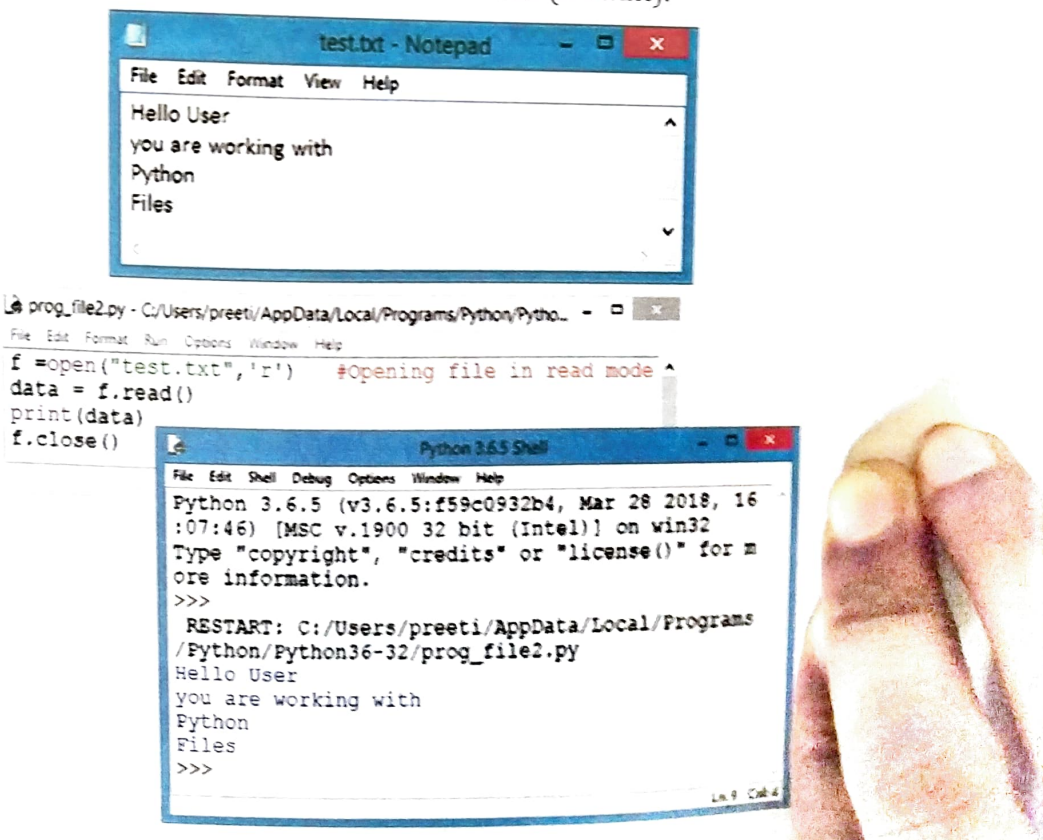
Syntax of read() function is:

```
fileObject.read() or
fileObject.read([size])
```

Here, size specifies the number of bytes to be read from the file. So, the function may be used to read a specific number of characters from the file. If the value of size is not provided or a negative value is specified as size, then the entire file will be read. One must take care of the memory size available before reading the entire content from the file.

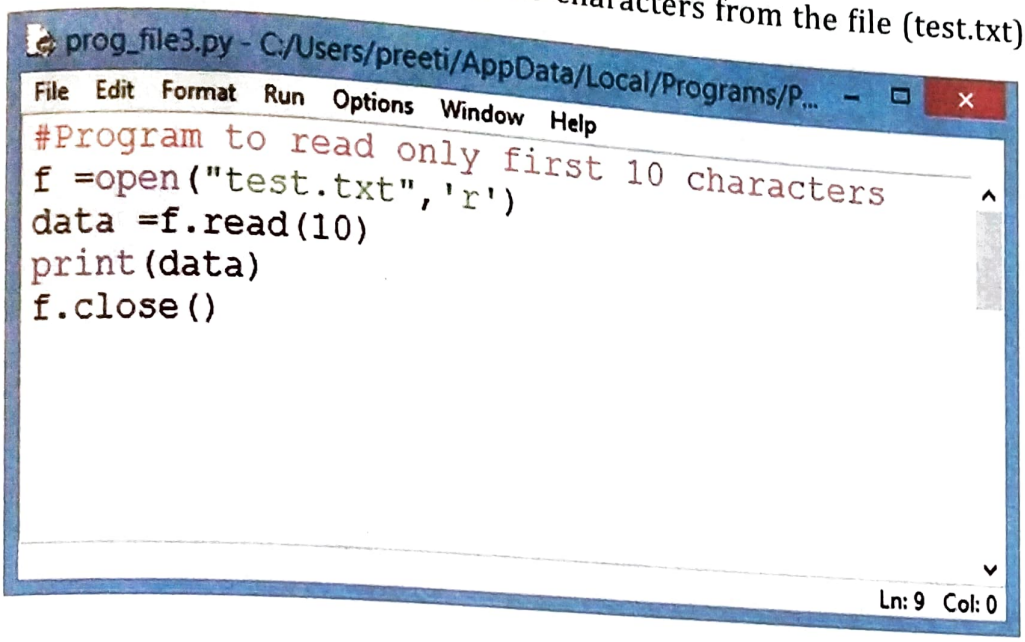
Practical Implementation-1

To illustrate read() by reading the entire data from a file (test.txt).

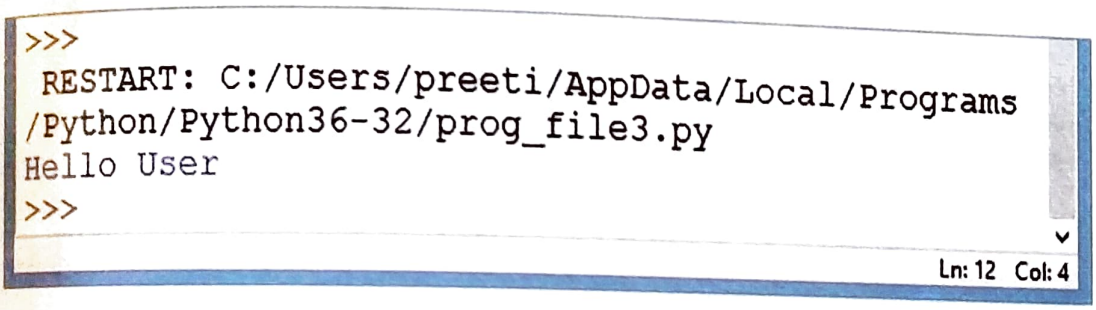


Practical Implementation–2

To illustrate read(n) by reading only the first 10 characters from the file (test.txt).



Output:



2) **readline:** readline() will return a line read, as a string from the file. First call to function will return the first line, second call the next line, and so on. We must remember that file object keeps track from where reading/writing of data should happen. For readline(); a line is terminated by '\n' (*i.e.* new line character). The new line character is also read from the file and post-fixed in the string. When end of file is reached, readline() will return an empty string. The syntax for readline() is:

Syntax:

```
fileObject.readline()
```

Since the method returns a string, it should be returned and stored using a variable as shown below:

```
>>> x = file.readline()
```

or

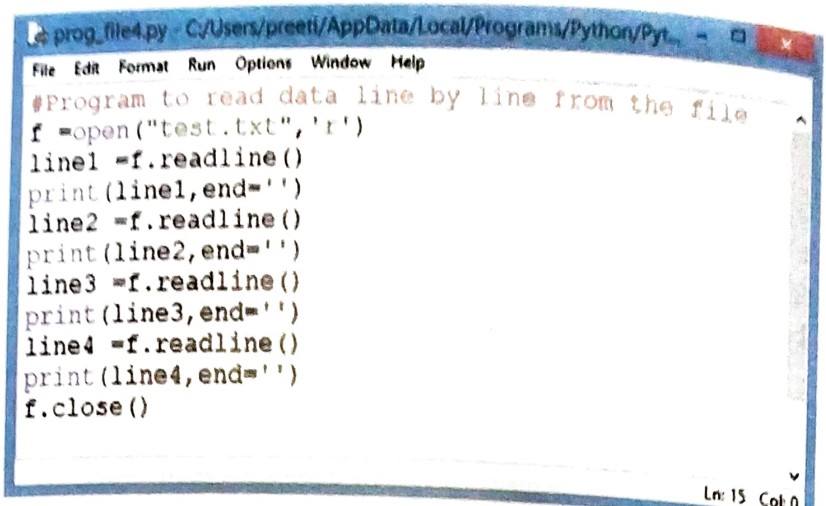
```
>>> print(file.readline())
```

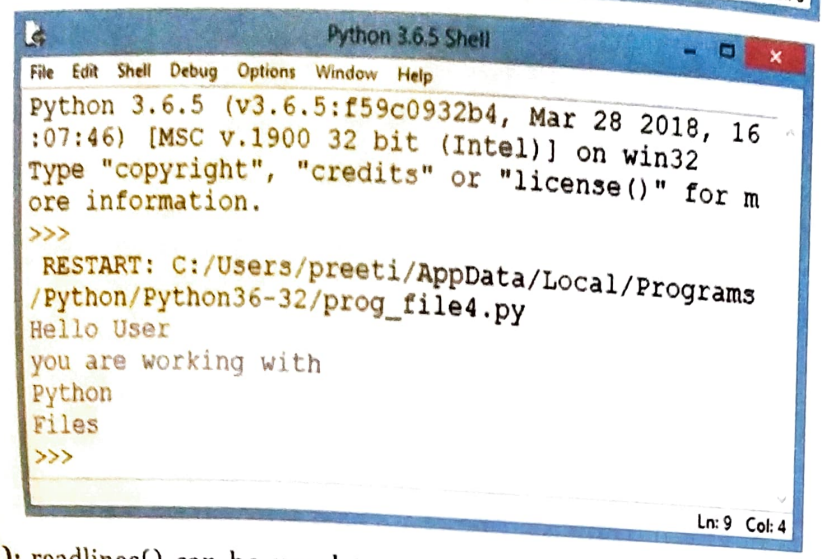
For reading an entire file using readline(), we will have to loop over the file object. This is a memory efficient, simple and fast way of reading the file, like:

```
>>> for line in file:
...     print(line)
```

Practical Implementation–3

To read data line by line using readline() method.





3) **readlines():** readlines() can be used to read the entire content of the file. We need to be careful while using it with respect to the size of memory required before using the function. The method will return a list of strings, each separated by \n. The syntax is:

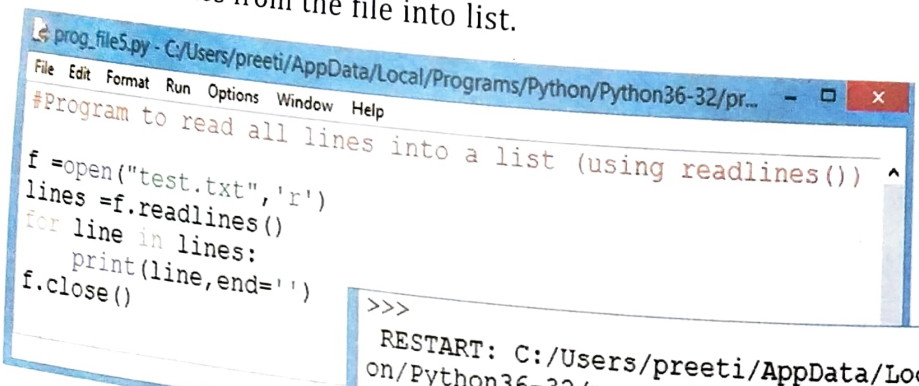
Syntax:

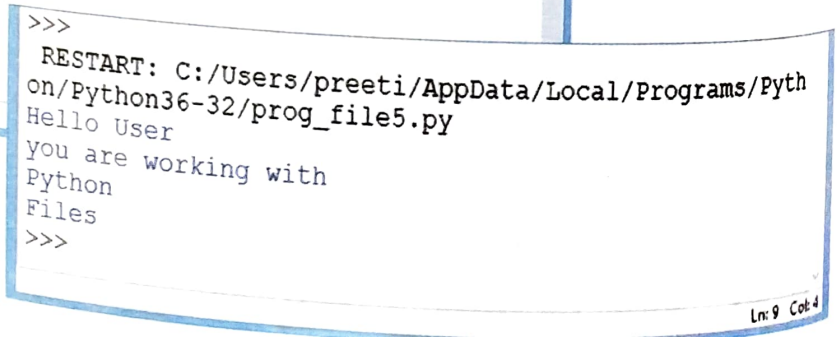
```
fileObject.readlines()
```

as it returns a list, which can then be used for manipulation.

Practical Implementation–4

To read all the lines from the file into list.





Practical Implementation–5

Program to display the contents of the file starting from 2nd character into the list. (Modification of Practical Implementation–4).

*prog_file6.py - C:/Users/preeti/AppData/Local/Programs/Python/Python3...

File Edit Format Run Options Window Help

```
#Program to read data from 2nd character into a list.

f =open("test.txt",'r')
print(f.read(2))
print(f.readlines())
print(f.read(3))
print("Remaining data")
print(f.read())
```

Ln: 10 Col: 0

Python 3.6.5 Shell

File Edit Shell Debug Options Window Help

```
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46) [MSC v.1
900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
 RESTART: C:/Users/preeti/AppData/Local/Programs/Python/Python36
-32/prog_file6.py
He
['llo User\n', 'you are working with\n', 'Python\n', 'Files']

Remaining data

>>>
```

Ln: 10 Col: 4