File Handling Mode (Read, Write, Create, Append)

Modes	Uses	Definition
"r"	Read	Opens a file for reading, error if the file does not exist
"a"	Append	Opens a file for appending, creates the file if it does not exist
"w"	Write	Opens a file for writing, creates the file if it does not exist
"x"	Create	Creates the specified file, returns an error if the file exists
"rb"	Read binary	Opens a file for reading only in binary format. The file pointer is placed at the beginning of the file. This is the default mode.
"r+"	Reading and Writing	Opens a file for both reading and writing. The file pointer placed at the beginning of the file.
"rb+"	Reading and Writing binary	Opens a file for both reading and writing in binary format. The file pointer placed at the beginning of the file.
"wb"	Writing only in binary format	Opens a file for writing only in binary format. Overwrites the file if the file exists. If the file does not exist, creates a new file for writing.
"w+"	Both writing and reading	Opens a file for both writing and reading. Overwrites the existing file if the file exists. If the file does not exist, creates a new file for reading and writing.
"wb+"	Both writing and reading in binary	Opens a file for both writing and reading in binary format. Overwrites the existing file if the file exists. If the file does not exist, creates a new file for reading and writing.
"ab"	Appending in binary	Opens a file for appending in binary format. The file pointer is at the end of the file if the file exists. That is, the

		file is in the append mode. If the file does not exist, it creates a new file for writing.
"a+"	Both appending and reading	Opens a file for both appending and reading. The file pointer is at the end of the file if the file exists. The file opens in the append mode. If the file does not exist, it creates a new file for reading and writing.
"ab+"	Both appending and reading in binary	Opens a file for both appending and reading in binary format. The file pointer is at the end of the file if the file exists. The file opens in the append mode. If the file does not exist, it creates a new file for reading and writing.

File Handling Functions (Open, Read, Write, Remove)

Functions	Syntax	Definition
open()	f = open("demofile.txt")	The open() function takes two parameters; *filename*, and *mode*.
read()	f = open("demofile.txt", "r") print(f.read())	The open() function returns a file object, which has a read()method for reading the content of the file
write()	f = open("demofile2.txt", "a") f.write("Now the file has more content!")	To write to an existing file, you must add a parameter to the open() function
remove()	import os os.remove("demofile.txt")	To delete a file, you must import the OS module, and run its os.remove() function

File Handling concepts

1. Predict the output:

file1.txt =

```
Hello! Welcome to Bennett university.
This file is for testing purposes.
Good Luck!
```

```
f = open("C:/Users/home/Desktop/Formatted/file1.txt", "r")
print(f.read())
```

1. Output:

Hello! Welcome to Bennett university.
This file is for testing purposes.
Good Luck!

2. Predict the output:

```
f = open("C:/Users/home/Desktop/Formatted/file1.txt", "r")
print(f.read(7))
print(f.read(10))
```

2. Output:

Hello!
Welcome to

3. Predict the output:

```
f = open("C:/Users/home/Desktop/Formatted/file1.txt", "r")
print(f.readline())
```

3. Output:

Hello! Welcome to Bennett university.

4. Predict the output:

```
f = open("C:/Users/home/Desktop/Formatted/file1.txt", "r")
for x in f:
  print(x)
```

4. Output:

Hello! Welcome to Bennett university.

This file is for testing purposes.

Good Luck!

5. Predict the output:

```
f = open("C:/Users/home/Desktop/Formatted/file1.txt", "w")
f.write("This is the write command\n")
f.write("It allows us to write in a particular file")
f.close()
f = open("C:/Users/home/Desktop/Formatted/file1.txt", "r")
print(f.read())
```

5. Output:

This is the write command
It allows us to write in a particular file

6. Write a Python program using function to read an entire text file.

Sol.

```
def file_read(fname):
        txt = open(fname)
        print(txt.read())

file_read("C:/Users/jagendra.singh/Desktop/yyy/file1.txt")
```

7. Predict the output:

file1:
It is the second-most populous country.
Python is a widely used high-level, general-purpose, interpreted, dynamic programming language.
Its design philosophy emphasizes code readability, and its syntax allows programmers to express concepts in fewer lines of code than possible in languages such as C++ or Java.
Python supports multiple programming paradigms, including object-oriented, imperative and functional programming or procedural styles.

```
def file_read(fname, nlines):
        from itertools import islice
        with open(fname) as f:
                for line in islice(f, nlines):
                        print(line)
file_read("C:/Users/jagendra.singh/Desktop/yyy/file1.txt",2)
```

7. Output:

It is the second-most populous country.
Python is a widely used high-level, general-purpose, interpreted, dynamic programming language.

8. Predict the output:

file1:
It is the second-most populous country.
Python is a widely used high-level, general-purpose, interpreted, dynamic programming language.
Its design philosophy emphasizes code readability, and its syntax allows programmers to express concepts in fewer lines of code than possible in languages such as C++ or Java.
Python supports multiple programming paradigms, including object-oriented, imperative and functional programming or procedural styles.

```
def file_read(fname):
        with open(fname) as f:
                content_list = f.readlines()
                print(content_list)

file_read("C:/Users/jagendra.singh/Desktop/yyy/file1.txt")
```

8. Output:

['It is the second-most populous country.\n', 'Python is a widely used high-level, general-purpose, interpreted, dynamic programming language.\n', 'Its design philosophy emphasizes code readability, and its syntax allows programmers to express concepts in fewer lines of code than possible in\n', 'languages such as C++ or Java. \n', 'Python supports multiple programming paradigms, including object-oriented, imperative and functional programming or procedural styles. \n', '\n']

9. Predict the output:

file1:
It is the second-most populous country.
Python is a widely used high-level, general-purpose, interpreted, dynamic programming language.

```
from collections import Counter
def abc(fname):
        with open(fname) as f:
                return Counter(f.read().split())

print(abc("C:/Users/jagendra.singh/Desktop/yyy/file1.txt"))
```

9. Output:

Counter({'is': 2, 'It': 1, 'the': 1, 'second-most': 1, 'populous': 1, 'country.': 1, 'Python': 1, 'a': 1, 'widely': 1, 'used': 1, 'high-level,': 1, 'general-purpose,': 1, 'interpreted,': 1, 'dynamic': 1, 'programming': 1, 'language.': 1})

10. Predict the output:

file2:

It is the second-most populous country. Python is a widely used high-level, general-purpose, interpreted, dynamic programming language.

```
def count(filepath):
    with open(filepath) as f:
        data = f.read()
        data.replace(",", " ")
        return len(data.split(" "))
print(count("C:/Users/jagendra.singh/Desktop/yyy/file2.txt"))
```

10. Output:

16

11. Predict the output:

file2:

It is the second-most populous country. Python is a widely used high-level, general-purpose, interpreted, dynamic programming language.

```
import random
def random_line(fname):
    lines = open(fname).read().splitlines()
    return random.choice(lines)
print(random_line('C:/Users/jagendra.singh/Desktop/yyy/file2.txt'))
```

11. Output:

```
It is the second-most populous country.
```

12. Write a Python program to write a list to a file.

Input: `color = ['Red', 'Green', 'White', 'Black', 'Pink', 'Yellow']`

Output:

```
Red
Green
White
Black
Pink
Yellow
```

Sol.

```
color = ['Red', 'Green', 'White', 'Black', 'Pink', 'Yellow']
with open('C:/Users/jagendra.singh/Desktop/yyy/file5.txt', "w") as myfile:
        for c in color:
                myfile.write("%s\n" % c)

content = open('C:/Users/jagendra.singh/Desktop/yyy/file5.txt')
print(content.read())
```

13. Write a Python program to remove newline characters from a file and put them in list.

Input: file:

It is the second-most populous country. Python is a widely used high-level, general-purpose, interpreted, dynamic programming language.

Output: ['It is the second-most populous country.', 'Python is a widely used high-level, general-purpose, interpreted, dynamic programming language.']

Sol.

```
def remove_newlines(fname):
    flist = open(fname).readlines()
    return [s.rstrip('\n') for s in flist]

print(remove_newlines('C:/Users/jagendra.singh/Desktop/yyy/file3.txt'))
```