Chapter 13

A.

1. C
2. C
3. C
4. C
5. D
6. C
7. B
8. C
9. C
10. D

B.

1. True
2. True
3. True
4. False
5. True
6. False
7. False

C.

1. A file is nothing but collection of records. Record is group of related data items. The data items may contain records related to students, employees, customers, etc. Usually a file is kept on a permanent storage device, e.g. a hard drive disk. A unique name and path is used by human users or in programs or scripts to access a file for reading and modification purposes. Thus the main advantage of file is to store information into it which can be accessed any time.

2. Different operations such as reading data from the file and writing data to the files can be performed. But prior performing operations on to the file, it has to be opened. Syntax to open file is as follows

```
file object = open(File_Name, [Access_Mode],[Buffering])
```

Thus above syntax will help us to open file. The following table describes various modes to open a file.

Mode	Description
R	Open file for reading

W	Open new file for writing. If file already exist, its contents are destroyed.
A	Opens a file for appending data from the end of the file.
Wb	Opens a file for writing a binary data
Rb	Opens a file for reading a binary data

Example:

```
F1 = open("Salary.txt","r") #Open File from Current Directory
```

3. Syntax to open, write text and close file as follws.

Syntax to Open File

```
file object = open(File_Name, [Access_Mode],[Buffering])
```

Example:

```
F1 = open("City.txt","r") #Open File from Current Directory
```

Syntax to Write text into the File

```
FileObject.write(String)
```

Example:

```
F1 = open("City.txt","w")

F1.write("Delhi")
```

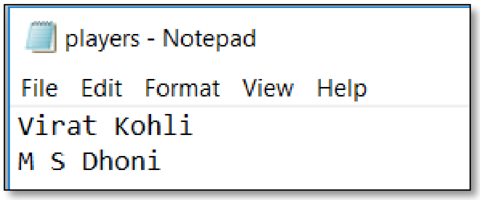
Syntax to close file

```
FileObject.close()
```

Example:

```
F1 = open("City.txt","r")
F1.write("Delhi")
F1.close()
```

4. The append **'a'** mode of a file is used to append data to the end of the existing file. Let us consider suppose we have following names of players added to the file players.txt

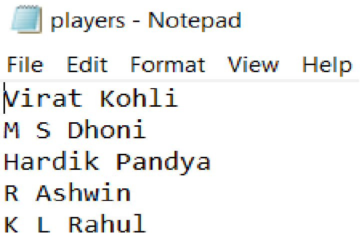


```
Virat Kohli
M S Dhoni
```

Now we will make use of append mode and write program to add following names of players to the file player.txt

Hardik Pandya, R Ashwin, K L Rahul

```
fp1=open('players.txt','a')   # Open file in append file
fp1.write('\nHardik Pandya')# Append contents to a file
fp1.write('\nR Ashwin')
fp1.write('\nK L Rahul')
fp1.close() #Close file
```



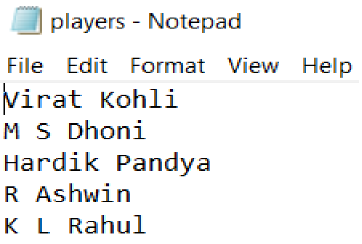
```
Virat Kohli
M S Dhoni
Hardik Pandya
R Ashwin
K L Rahul
```

Contents of File after Appending

5. The uses of seek() functions are as follows
   a. Programmer can set cursor at any position within the file.
   b. Placing cursor at any point helps programmer to read the contents of file from the point where the cursor has specified.
   c. Thus seek helps to skips certain characters from start of the file or end of the file to read the remaining contents of file.

Example:

Read all the names of players from file players.txt except first name using seek function.



```
Virat Kohli
M S Dhoni
Hardik Pandya
R Ashwin
K L Rahul
```

```
fp1=open('players.txt','r')  # Open file in append file
fp1.seek(12,0)
print(fp1.read())
```

6. The **seek()** method is used to set the file pointer to specified position within the file. The syntax of seek() function is as follows

Syntax:

```
File_object.seek(offset , whence)
```

7. Refer answer to Q9

8. Binary files don't have text in them. They might have pictures, music, or some kind of data. Even there are no newlines in binary files binary files. Therefore we cannot use readline() and readlines() method on binary file. Access mode 'r' is required to open normal text files. But in order to open binary files one should include **'b' i.e. 'rb'** to read binary file and **'wb'** to write into binary file.

9. The inbuilt functions supported by python for file handling are as follows

In build Function Name	Meaning
str readline()	Returns the next line of the file as a string.
list readlines()	Returns a list containing all the lines in the file.
str read([int number])	Returns the specified number of characters from the file. If the argument is omitted, then entire contents of file are read.

Write (str s)	Write string s to the file.
close()	Close the file

10. The following program describes insertion of numbers from 1 to 5 into the file demo.txt.

```
def main():
    obj1 = open("Demo.txt","w") #Open File in Write mode
    for x in range(1,6): # Iterates from 1 to 5
        x=str(x)           # Convert Number to String
        obj1.write(x)      # Write Number to a output file
        obj1.write(" ")    # Space to separate Numbers
    obj1.close()           # Close  File
main()  # Call to main function
```

The program opens Demo.txt file W mode i.e. write mode. The for loop iterates for 20 times to write numbers from 1 to 6 into the file. The numbers are converted to strings before using **str** method before being written to the file.

D.

1.

```
fp1 = open("Salary.txt","r")
num =  (fp1.readlines())
fp1.close()
fp1 = open("Salary.txt","r")
size = len(num)
sum = 0
for i in range(size):
    num1 = int(fp1.readline())
    print(num1)
    sum = sum + (int)(num1)
print('Sum of all the Salaries of Employees:')
print(sum)
```

2.

```
def Find_Smallest(fp1):
    fp1 = open('Demo1.txt','r') #Open File in read Mode
    small = "                                                       "
    L = 0
    count = 0
    for line in fp1:
        count= count + 1
        print(' Line No: ',count)
        print(line)
       print(' Number of Character = ',len(line))
```

```
        print('-----------------------------------')
        if(len(line) < len(small)):
           small = line
           L = line
    print(L, 'is the Smallest Line with', len(small),'characters')
fp = open('Demo1.txt','r')
Find_Smallest(fp)
```

3.

```
IP_File = open('Demo.txt','r')
Out_File = open('Demo2.txt','w')
for line in IP_File:
    if  line[0] not in 'ABCDEFGHIJKLMNOPQRSTUVWXYZ':
        Out_File.write(line)
Out_File.close()
```

4.

```
IP_File = open('Demo.txt','r')
Out_File = open('Demo2.txt','w')
for line in IP_File:
    Out_File.write(line)
Out_File.close()
```

5.

```
IP_File = input('Enter the name of a Python file:')
print(IP_File)
IP_File = open(IP_File,'r')
Out_File = input('Enter the name of a Python file')
Out_File = open(Out_File,'w')
for line in IP_File:
    pos = line.find("#")
    if pos > -1:
        line  = line[0:pos]
        line = line +'\n'
    Out_File.write(line)
IP_File.close()
Out_File.close()
```