J. BASKARAN M.Sc., B.Ed. (C.S)
jbaskaran89@gmail.com
Puducherry.

J. ILAKKIA M.Sc., M.Phil., B.Ed. (C.S)
jilakkia@gmail.com
Puducherry.

13. PYTHON AND CSV FILES

Section – A

Choose the best answer **(1 Mark)**

1. A CSV file is also known as a ….

(A) Flat File (B) 3D File (C) String File (D) Random File

2. The expansion of CRLF is

(A) Control Return and Line Feed (B) Carriage Return and Form Feed

(C) Control Router and Line Feed **(D) Carriage Return and Line Feed**

3. Which of the following module is provided by Python to do several operations on the CSV files?

(A) py (B) xls **(C) csv** (D) os

4. Which of the following mode is used when dealing with non-text files like image or exe files?

(A) Text mode **(B) Binary mode** (C) xls mode (D) csv mode

5. The command used to skip a row in a CSV file is

(A) next() (B) skip() (C) omit() (D) bounce()

6. Which of the following is a string used to terminate lines produced by writer()method of csv module?

(A) Line Terminator (B) Enter key (C) Form feed (D) Data Terminator

7. What is the output of the following program? import csv

```
d=csv.reader(open('c:\PYPRG\ch13\city.csv'))
next(d)
for row in d:
print(row)
```

if the file called "city.csv" contain the following details

```
chennai,mylapore
mumbai,andheri
```

A) chennai,mylapore **(B) mumbai,andheri**

(C) chennai
mumba

(D) chennai,mylapore
mumbai,andheri

8. Which of the following creates an object which maps data to a dictionary?

(A) listreader() (B) reader() (C) tuplereader() **(D) DictReader ()**

9. Making some changes in the data of the existing file or adding more data is called

(A)Editing (B) Appending **(C) Modification** (D) Alteration

10. What will be written inside the file test.csv using the following program import csv

```
D = [['Exam'],['Quarterly'],['Halfyearly']]
csv.register_dialect('M',lineterminator = '\n')
with open('c:\pyprg\ch13\line2.csv', 'w') as f:
wr = csv.writer(f,dialect='M')
wr.writerows(D)
f.close()
```

(A) Exam Quarterly Halfyearly

(B) Exam Quarterly Halfyearly

(C) E
Q
H

**(D) Exam,
Quarterly,
Halfyearly**

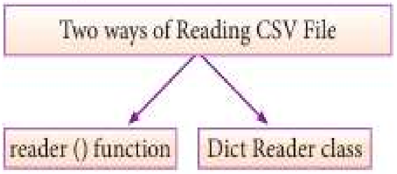
Section-B

Answer the following questions **(2 Marks)**

1. What is CSV File?

- A CSV file is a human readable text file where each line has a number of fields, separated by commas or some other delimiter.
- A CSV file is also known as a Flat File that can be imported to and exported from programs that store data in tables, such as ***Microsoft Excel*** or ***OpenOfficeCalc.***

2. Mention the two ways to read a CSV file using Python.



3. Mention the default modes of the File.

- The default is reading (‘r’) in text mode.
- In this mode, while reading from the file the data would be in the format of **strings.**

4. What is use of next() function?

- **"next()"command** is used to avoid or skip the first row or row heading.
- **Example:** While sorting the row heading is also get sorted, to avoid that the first is skipped using next().
- Then the list is sorted and displayed.

5. How will you sort more than one column from a csv file? Give an example statement.

- To sort by more than one column you can use **itemgetter** with multiple indices.

Syntax: **operator.itemgetter(col_no)**

Example: sortedlist = sorted (data, key=operator.itemgetter(1))

Section-C

Answer the following questions **(3 Marks)**

1. Write a note on open() function of python. What is the difference between the two methods?

- Python has a built-in function **open()** to open a file.
- This function returns a file object, also called a handle, as it is used to read or modify the file accordingly.
- The **default is reading** in text mode.
- In this mode, while reading from the file the data would be in the format of **strings.**
- On the other hand, binary mode returns bytes and this is the mode to be used when dealing with non-text files like image or exe files.

2. Write a Python program to modify an existing file.

- In this program, the third row of "student.csv" is modified and saved.
- First the "student.csv" file is read by using csv.reader() function.
- Then, the list() stores each row of the file.
- The statement "lines[3] = row", changed the third row of the file with the new content in "row".
- The file object writer using writerows (lines) writes the values of the list to "student.csv" file.

PROGRAM: student.csv

```
import csv
row = ['3', 'Meena','Bangalore']
with open('student.csv', 'r') as readFile:
reader = csv.reader(readFile)

lines = list(reader) # list()- to store each row of data as a list
lines[3] = row
with open(‘student.csv’, ‘w’) as writeFile:
# returns the writer object which converts the user data with delimiter
writer = csv.writer(writeFile)
#writerows()method writes multiple rows to a csv file
writer.writerows(lines)
readFile.close()
writeFile.close()
```

3. Write a Python program to read a CSV file with default delimiter comma (,).

```
#importing csv
import csv                  #opening the csv file which is in different location with read mode
with open('c:\\pyprg\\sample1.csv', 'r') as F:
                            #other way to open the file is f= ('c:\\pyprg\\sample1.csv', 'r')
reader = csv.reader(F)      # printing each line of the Data row by row
print(row)
F.close()
```

OUTPUT:

```
['SNO', 'NAME', 'CITY']
['12101', 'RAM', 'CHENNAI']
['12102', 'LAVANYA', 'TIRUCHY']
['12103', 'LAKSHMAN', 'MADURAI']
```

4. What is the difference between the write mode and append mode.

Write Mode	Append Mode
• 'w'	• 'a'
• Open a file for writing.	• Open for appending at the end of the file without truncating it.
• Creates a new file if it does not exist or truncates the file if it exists.	• Creates a new file if it does not exist.

5. What is the difference between reader() and DictReader() function?

Reader():

- The reader function is designed to take each line of the file and make a list of all columns.
- Using this method one can read data from csv files of different formats like quotes (" "), pipe (|) and comma (,).
- csv. Reader work with list/tuple.
- **Syntax:** csv.reader(fileobject,delimiter,fmtparams)

DictReader():

- DictReader works by reading the first line of the CSV and using each comma separated value in this line as a dictionary key.
- DictReader is a class of csv module is used to read a CSV file into a dictionary.
- It creates an object which maps data to a dictionary.
- csv.DictReader work with dictionary.

Section - D

Answer the following questions: **(5 Marks)**

1. Differentiate Excel file and CSV file.

Excel	CSV
• Excel is a binary file that holds information about all the worksheets in a file, including both content and formatting.	• CSV format is a plain text format with a series of values separated by commas.
• XLS files can only be read by applications that have been especially written to read their format, and can only be written in the same way.	• CSV can be opened with any text editor in Windows like notepad, MS Excel, OpenOffice, etc.
• Excel is a spreadsheet that saves files into its own proprietary format viz. xls or xlsx	• CSV is a format for saving tabular information into a delimited text file with extension .csv
• Excel consumes more memory while importing data	• Importing CSV files can be much faster, and it also consumes less memory

2. Tabulate the different mode with its meaning.

Python File Modes:

Mode	Description
'r'	• Open a file for reading. (default)
'w'	• Open a file for writing. Creates a new file if it does not exist or truncates the file if it exists.
'x'	• Open a file for exclusive creation. If the file already exists, the operation fails.
'a'	• Open for appending at the end of the file without truncating it. Creates a new file if it does not exist.
't'	• Opren in text mode. (default)
'b'	• Open in binary mode.
'+'	• Open a file for updating (reading and writing)

3. Write the different methods to read a File in Python.

- Contents of CSV file can be read with the help of **csv.reader()** method.
- **The reader function is designed to take each line of the file and make a list of all columns.**
- Using this method one can read data from csv files of different formats like,

  1. CSV file **-** data with default delimiter comma (,)
  2. CSV file **-** data with Space at the beginning
  3. CSV file **-** data with quotes
  4. CSV file **-** data with custom Delimiters

- **The syntax for csv.reader() is** csv.reader(fileobject,delimiter,fmtparams**)**

i) CSV file with default delimiter comma (,)

The following program read a file called “sample1.csv” with default delimiter comma (,) and print row by row.

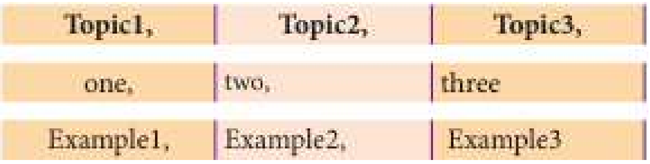
```
import csv
with open('c:\\pyprg\\sample1.csv', 'r') as F:
reader = csv.reader(F)
print(row)
F.close()
```

OUTPUT:

```
['SNO', 'NAME', 'CITY']
['12101', 'RAM', 'CHENNAI']
['12102', 'LAVANYA', 'TIRUCHY']
['12103', 'LAKSHMAN', 'MADURAI']
```

ii) CSV files- data with Spaces at the beginning

Consider the following file “sample2.csv” containing the following data when opened through notepad



Topic1,	Topic2,	Topic3,
one,	two,	three
Example1,	Example2,	Example3

The following program read the file through Python using “csv.reader()”.

```
import csv
csv.register_dialect('myDialect',delimiter = ',',skipinitialspace=True)
F=open('c:\\pyprg\\sample2.csv','r')
reader = csv.reader(F, dialect='myDialect')
for row in reader:
print(row)
F.close()
```

OUTPUT:

```
['Topic1', 'Topic2', 'Topic3']
['one', 'two', 'three']
['Example1', 'Example2', 'Example3']
```

- These whitespaces in the data can be removed, by registering new dialects using **csv.register_dialect()** class of csv module.
- **A dialect describes the format of the csv file that is to be read.**
- In dialects the parameter **“skipinitialspace”** is used for removing whitespaces after the delimiter.

iii) CSV File-Data With Quotes

- You can read the csv file with quotes, by registering new dialects using csv.register_dialect() class of csv module.
- Here, we have quotes.csv file with following data.

SNO,Quotes

1, "The secret to getting ahead is getting started."

2, "Excellence is a continuous process and not an accident."

The following Program read “quotes.csv” file, where delimiter is comma (,) but the quotes are within quotes (“ “).

```
import csv
csv.register_dialect('myDialect',delimiter = ',',quoting=csv.QUOTE_ALL,
skipinitialspace=True)
f=open('c:\\pyprg\\quotes.csv','r')
reader = csv.reader(f, dialect='myDialect')
for row in reader:
print(row)
```

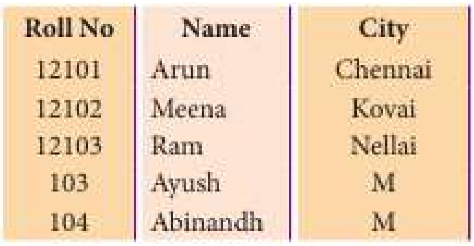
OUTPUT:

```
['SNO', 'Quotes']
['1', 'The secret to getting ahead is getting started.']
['2', 'Excellence is a continuous process and not an accident.']
```

- In the above program, register a dialect with name myDialect.
- Then, we used **csv. QUOTE_ALL to display all the characters** after double quotes.

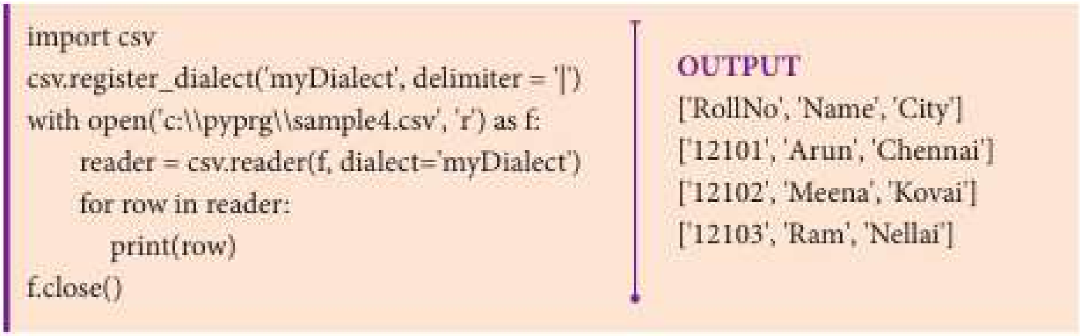
iv) CSV files with Custom Delimiters

- You can read CSV file having custom delimiter by registering a new dialect with the help of csv.register_dialect().



Roll No	Name	City
12101	Arun	Chennai
12102	Meena	Kovai
12103	Ram	Nellai
103	Ayush	M
104	Abinandh	M

- In the following file called “sample4.csv”,each column is separated with | (Pipe symbol)



```
import csv
csv.register_dialect('myDialect', delimiter = '|')
with open('c:\\pyprg\\sample4.csv', 'r') as f:
    reader = csv.reader(f, dialect='myDialect')
    for row in reader:
        print(row)
f.close()
```

OUTPUT

```
['RollNo', 'Name', 'City']
['12101', 'Arun', 'Chennai']
['12102', 'Meena', 'Kovai']
['12103', 'Ram', 'Nellai']
```

4. Write a Python program to write a CSV File with custom quotes.

```
import csv
info = [[‘SNO’, ‘Person’, ‘DOB’],
[‘1’, ‘Madhu’, ‘18/12/2001’],
[‘2’, ‘Sowmya’,’19/2/1998’],
[‘3’, ‘Sangeetha’,’20/3/1999’],
[‘4’, ‘Eshwar’, ‘21/4/2000’],
[‘5’, ‘Anand’, ‘22/5/2001’]]
csv.register_dialect(‘myDialect’,quoting=csv.QUOTE_ALL)
with open(‘c:\\pyprg\\ch13\\person.csv’, ‘w’) as f:
writer = csv.writer(f, dialect=’myDialect’)
for row in info:
writer.writerow(row)
f.close()
```

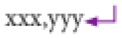
<u>OUTPUT :</u>

```
“SNO”,”Person”,”DOB” ”1”,”Madhu”,”18/12/2001” ”2”,”Sowmya”,”19/2/1998”
     ”3”,”Sangeetha”,”20/3/1999” ”4”,”Eshwar”,”21/4/2000”
“5”,”Anand”,”22/5/2001”
```

5. Write the rules to be followed to format the data in a CSV file.

1. Each record (row of data) is to be located on a separate line, delimited by a line break by pressing enter key.

<u>For example:</u>

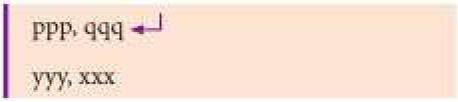


xxx,yyy↵

↵ denotes enter Key to be pressed

2. The last record in the file may or may not have an ending line break.

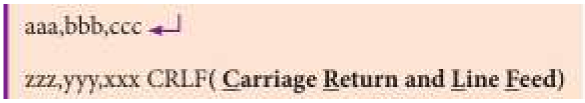
For example:



```
ppp, qqq ↵
yyy, xxx
```

3.

- There may be an optional header line appearing as the first line of the file with the same format as normal record lines.
- The header will contain names corresponding to the fields in the file and should contain the same number of fields as the records in the rest of the file.
- **For example:** field_name1,field_name2,field_name3



```
aaa,bbb,ccc ↵
zzz,yyy,xxx CRLF( Carriage Return and Line Feed)
```

4.

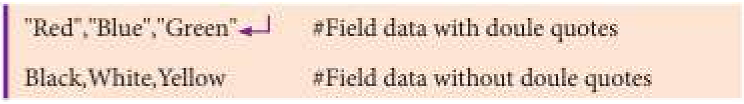
- Within the header and each record, there may be one or more fields, separated by commas.
- Spaces are considered part of a field and should not be ignored.
- The last field in the record must not be followed by a comma.

For example: Red , Blue

5.

- Each field may or may not be enclosed in double quotes.
- If fields are not enclosed with double quotes, then double quotes may not appear inside the fields.

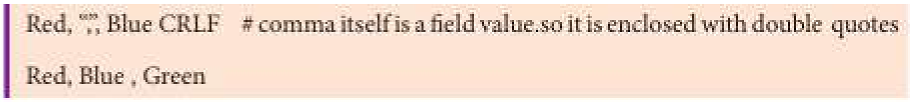
For example:



```
"Red","Blue","Green"↵     #Field data with doule quotes
Black,White,Yellow        #Field data without doule quotes
```

6.

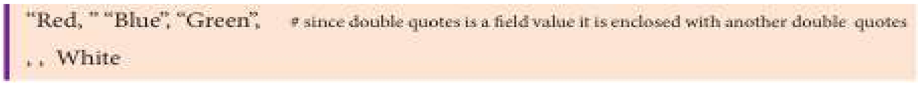
- Fields containing line breaks (CRLF), double quotes, and commas should be enclosed in double-quotes.
- **For example:**



```
Red, ",", Blue CRLF    # comma itself is a field value.so it is enclosed with double  quotes
Red, Blue , Green
```

7.

- If double-quotes are used to enclose fields, then a double-quote appearing inside a field must be preceded with another double quote.
- **For example:**



```
"Red, " "Blue", "Green",     # since double quotes is a field value it is enclosed with another double  quotes
, ,  White
```