
Practical Set -6

1. Create a file with file name sample.txt, accept some data from the user and store it in the file.

Get data from the user

```
data = input("Enter some data: ")
# Write data to file
with open("sample.txt", "w") as file:
    file.write(data)
print("Data has been saved to sample.txt")
```

2. Display the data stored in the sample.txt file (created in question 1).

Read data from file

```
with open("sample.txt", "r") as file:
    data = file.read()
# Display the data
print("Data stored in sample.txt:")
print(data)
```

3. Accept some data from the user and append it into the file sample.txt (created in question 1),also the data in the file.

Accept some data from the user

```
new_data = input("Enter some new data: ")
# Append data to file
with open("sample.txt", "a") as file:
    file.write("\n" + new_data)
# Read data from file
with open("sample.txt", "r") as file:
    all_data = file.read()
# Display all the data in the file
print("All data stored in sample.txt:")
print(all_data)
```

4. Accept the file name from the user, check the availability of the file: i).

If the file exists display the data on the screen, ii). If the file is not available, display the appropriate message.

```
import os
# Accept file name from user
file_name = input("Enter the file name: ")
# Check if the file exists
if os.path.exists(file_name):
    # Read data from file
    with open(file_name, "r") as file:
```

Practical Set -6

```
        data = file.read()
    # Display the data
    print("Data stored in", file_name, ":")
    print(data)
else:
    print("File", file_name, "does not exist.")

# 5. Accept the file name from the user, check the availability of the file:
# a. If the file exists, display: i). No. of characters, ii). No. of words
and iii). No. of lines# b. If the file does exist, than display the
appropriate message.

import os

# Accept file name from user
file_name = input("Enter the file name: ")
# Check if the file exists
if os.path.exists(file_name):

    # Open the file and read its contents
    with open(file_name, "r") as file:
        content = file.read()
    # Count characters, words, and lines
    num_chars = len(content)
    num_words = len(content.split())
    num_lines = content.count('\n') + 1 # Adding 1 for the last line without
'\n'
    # Display the results
    print("Number of characters:", num_chars)
    print("Number of words:", num_words)
    print("Number of lines:", num_lines)
else:
    print("File", file_name, "does not exist.")

6. Create and open the binary file with 'with' option. Store names of all the
subjects you study in semester 2. Ask user to enter the subject number they
wanted to see and display that subject name.

# with open('sub.bin','wb') as f:
#     n = int(input('Enter Number Of Subject in Sem - 2 :'))
#     for i in range(n):
#         sub = input('Enter Subject of Sem -2 : ')
#         ln = len(sub)
#         sub = sub.encode()
```

Practical Set -6

```
#         f.write(sub)
with open('sub.bin','rb') as f:
    str = f.read()
    print("Sub : ",str.decode()," ")

# 7. Create a file named 'img1', store an image into it. Open another file named
'img2', copy the same image as in the file 'img1'. Also store both files into the
zip file named 'imp_img'.
from zipfile import *
f = open('img1.png','rb')
f1 = open('img2.png','wb')

zip = ZipFile('Imp_img.zip','w',ZIP_DEFLATED)
img = f.read()
f1.write(img)

zip.write('img1.png')
zip.write('img2.png')
f.close()
f1.close()
zip.close()

8. Create a file with 'with' option, name it as 'marks.dat'.
    i). Accept subject name and marks from the user, store the data in the file.
    ii).      Give three options to the user: a). To view whole file, b).
    Accept and edit the marks of a subject user want to change.
    iii). Exit
import os
import mmap
import sys

print('1. Enter Subject Name and Marks...')
print('2. Show A File...')
print('3. Edit Marks of Subject...')
print('4. Exit')

ch = input('Enter Your Choice : ')
# with open('marks.dat','w') as f:

if ch == '1':
```

Practical Set -6

```
with open('marks.dat','w') as f:

    n = int(input('Enter Number Of Subject You Want To Insert : '))
    for i in range(n):
        sub = input('Enter Subject Name :')
        # sub = sub.encode()
        # f.write(sub)

        mark = input('Enter Mark of Subject : ')
        # mark = mark.encode()
        # f.write(mark)

        f.write("{}: {}\n".format(sub, mark))
        # print(sub.decode())
        # print(mark.decode())
# with open('marks.dat','r') as f:

# mm = mmap.mmap(f.fileno(),0, access=mmap.ACCESS_READ)
if ch == '2':
    with open('marks.dat','r') as f:
        # print(mm.readline().decode())
        for line in f:
            print(line.strip())
if ch == '4':
    sys.exit()

# 9. Create a regular expression that:
# a). Identifies and display the string starting with 's' and having 4
characters.

import re

str1 = "sparrows sing sweetly as they soar swiftly through the sky."
result = re.findall(r's[\w]*',str1)
print(result)

# b). Splits the string where some special characters are found.
import re

str1 = "Hello!Iam MCAStudent. Iam Study In AtmiyaUniversity."
res = re.split(r'[\W_]', str1 )
print(res)
```

Practical Set -6

```
# c). Display the word starting with number.
import re
str1 = "1one two three four 5five "
res = re.findall(r'\b\d\w*\b', str1)
print(res)

# d). Display the word having 3 or 4 or 5 characters.
import re

str1 = "one two three four five. this is a number of one to five."
res = re.findall(r'\b\w{3,5}\b', str1)
print(res)

# e). Display only the dates from the string.
import re

str1 = "001 abc 02-03-2022, 002 bcd 04-12-2011, 003 efg 4-2-10"

res = re.findall(r'\d{2}-\d{2}-\d{4}', str1)
print(res)

# f). Create a string with name of the person and his Aadhar number, display
only Aadhar number.
import re

str1 = "abc 123456789012, bcd 234567890123, efg 1243567809"

res = re.findall(r'\d{12}', str1)
print(res)

# g). Display all the words that starts with 'at' or 'ap'.
import re

str1 = "atmiya university android application "
res = re.findall(r'a[tp][\w]*', str1)
print(res)

# h). Check if the string starts with 'at' than display appropriate message and
otherwise.
import re

str1 = "atmiya university android application "
res = re.findall(r'at[\w]*', str1)
if res:
```

Practical Set -6

```
    print('String Start With AT..')
else:
    print('Sorry, But String Not Start With AT..')

# 10. Do as directed:
# a). Name the package used to deal with data frame.
=> Pandas
# b). Name the package used to deal with data .xlsx file.
=> Openpyxl
# c). Name the function used to read the .csv file.
=> read.csv()
# d). Name the function used to read the .xlsx file.
=> read.excel()
# e). Name the function used to read the tuple.
=>

# 11. Create a dictionary which stores (at least 10 records)empid, name, city,
salary and perform
# following operations:
# a). Display first three records
# b). Display last five records
# c). Display only Name and City
# d). Display employee who belongs to Mumbai
# e). Display employee name who belongs to Mumbai
# f). Display employee whose salary is more than 25000

import pandas as pd
employees = {
    101: {'name': 'John', 'city': 'New York', 'salary': 6000},
    102: {'name': 'Alice', 'city': 'Los Angeles', 'salary': 75000},
    103: {'name': 'Bob', 'city': 'Chicago', 'salary': 5000},
    104: {'name': 'Emily', 'city': 'Houston', 'salary': 62000},
    105: {'name': 'Michael', 'city': 'San Francisco', 'salary': 80000},
    106: {'name': 'Sophia', 'city': 'Mumbai', 'salary': 7000},
    107: {'name': 'David', 'city': 'Seattle', 'salary': 72000},
    108: {'name': 'Emma', 'city': 'Mumbai', 'salary': 9000},
```

Practical Set -6

```
109: {'name': 'Daniel', 'city': 'Atlanta', 'salary': 8000},
110: {'name': 'Olivia', 'city': 'Mumbai', 'salary': 63000}
}

df = pd.DataFrame.from_dict(employees, orient='index')

print('=====')

print('1.Display First Three Record')
print(df.head(3))
print('=====')

print('2. Display Last Five Records')
print(df.tail(5))
print('=====')

print('3. Display Only Name And City ')
print(df[['name', 'city']])

print('=====')

print('4. Display employee who belongs to Mumbai')
mu = df[df['city'] == 'Mumbai']
print(mu)

print('=====')

print('5. Display employee name who belongs to Mumbai')
# df = pd.DataFrame.from_dict(employees, orient='index')
nmu = df.loc[df['city'] != 'Mumbai']
print(nmu)

print('=====')

print('6. Display employee whose salary is more than 25000')
sal = df[df['salary'] > 25000]
print(sal)
```

Practical Set -6

12. Create an xlsx file store marks of five subjects, plot the data on the bar graph.

```
import xlrd
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_excel('Mark.xlsx')
print(df)
```

```
x = df['subject']
y = df['marks']
```

```
plt.bar(x,y,label="Mark Sheet")
plt.xlabel('Subject')
plt.ylabel('Marks')
```

```
plt.title('Mark Sheet')
plt.legend()
plt.show()
```

13. Take five income source of the Government and display it on the pie chart.

```
import xlrd
import pandas as pd
import matplotlib.pyplot as plt
```

```
df = pd.read_excel('Government.xlsx')
print(df)
x = df['source']
y = df['amount']
```

```
plt.bar(x,y,label="")
plt.xlabel('Source')
plt.ylabel('Amount')
plt.title('Government Income')
plt.legend()
plt.show()
```

14. Draw the line chart representing BSE (Bombay Stock Exchange) index in last 10 years.

```
import matplotlib.pyplot as plt
years = [2014,2015,2016,2017,2018,2019,2020,2021,2022,2023]
bsc = [89,59,74,84,85,94,16,45,36,47]
```

Practical Set -6

```
plt.plot(years,bsc,color='teal')
plt.title("Bombay Stock Exchange")
plt.xlabel('Years')
plt.ylabel('BSC')
plt.legend()
plt.show()
```

15. Plot the grouped bar graph using the appropriate data.

```
import matplotlib.pyplot as plt
# Employee id and their salary of Production department
x = [101,102,105,107,110]
y = [25000,26000,20000,21000,23000]

# Employee id and their salary of QA department
x1 = [103,104,106,108,109]
y1 = [19000,16000,18000,22000,21000]

plt.bar(x,y,label="Production Department",color="teal")
plt.bar(x1,y1,label="QA Department",color="yellow")
plt.xlabel('Employee Id')

plt.ylabel('Employee Salary')
plt.title('TCS')
plt.legend()
plt.show()
```