LAB # 13

FILE PROCESSING

OBJECTIVE

To explore methods to access, open/close , modes contained in external files

THEORY

When a program is terminated, the entire data is lost. Storing in a file will preserve your data even if the program terminates.

File Handling in Python

Python allows users to handle files to read and write files, along with many other file handling options. Python treats file differently as **text** or **binary** and this is important. Each line of code includes a sequence of characters and they form text file. Each line of a file is terminated with a special character, called the EOL or End of Line characters like comma {,} or newline character. It ends the current line and tells the interpreter a new one has begun..

File Operations

There are different operations that can be carried out on a file, these are:

- Creation of a new file
- Opening an existing file
- Reading from a file
- Writing to a file
- Closing a file

Open() function

Open () function in Python to open a file in read or write mode.

syntax: open(Directory:\\filename, mode)

There are various kinds of mode, that python provides:

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'	Open in text mode. (default)
'b'	Open in binary mode.
'+'	Open a file for updating (reading and writing)
'rb'	Opens a file for reading binary data.
'wb'	Opens a file for writing binary data.

Example

```
# a file named "python", will be opened with the reading mode
file = open('python.txt', 'r')
# This will print every line one by one in the file
for each in file:
      print (each)
```

Read() mode

There is more than one way to read a file in Python. To extract a string that contains all characters in the file then it can be use as;

syntax: file.read()

Example

```
# Python code to illustrate read() mode
file = open("python.text", "r")
print file.read()
```

Another way to read a file is to call a certain number of characters like in the following code the interpreter will read the first five characters of stored data and return it as a string:

Example

```
# Python code to illustrate read() mode character wise
file = open("python.txt", "r")
print file.read(5)
```

Write() mode

To manipulate the file, write the following in your Python environment:

syntax: file.write()

The close() command terminates all the resources in use and frees the system of this particular program.

syntax: file.close()

Example:

```
# Python code to create a file
file = open(' python.txt','w')
file.write("This is the write command")
file.write("It allows us to write in a particular file")
file.close()
```

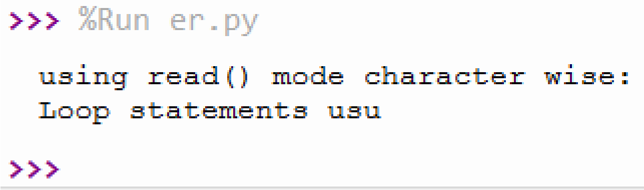
EXERCISE

A. Point out the errors, if any, and paste the output also in the following Python programs.

1. Code

```
file=open('python.txt','r')
print("using read() mode character wise:")
s1=file.read(19)
print(s1)
```

Output:



```
>>> %Run er.py

  using read() mode character wise:
  Loop statements usu

>>>
```

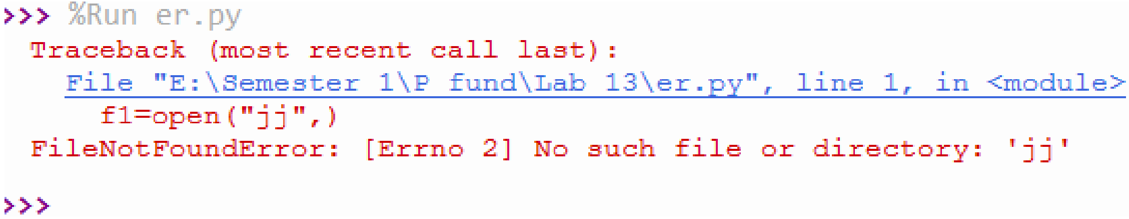
No error

2. Code

```
f1=open("jj",)
f1.write("something")
```

Output:

Extension of file is missing.



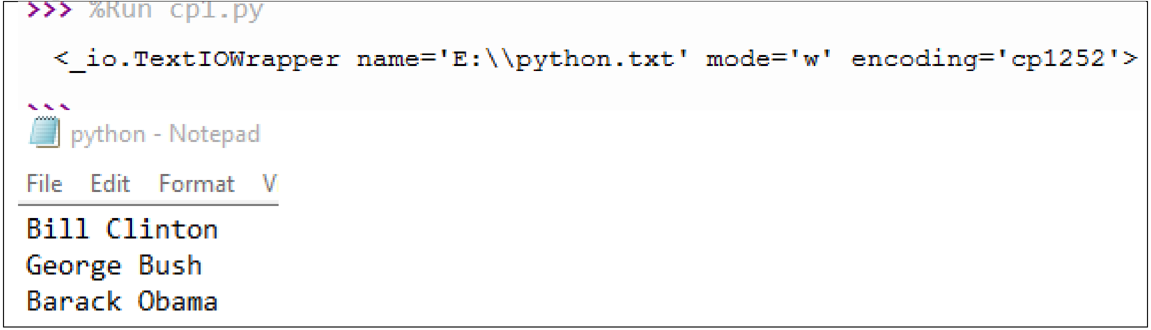
```
>>> %Run er.py
  Traceback (most recent call last):
    File "E:\Semester 1\P fund\Lab 13\er.py", line 1, in <module>
      f1=open("jj",)
  FileNotFoundError: [Errno 2] No such file or directory: 'jj'

>>>
```

B. Creat a text file named python, Write the following code. Execute it and show the output. (You can use the Snipping Tool to take a snapshot of your txt.file)

1. Code

```
def main():
   # Open file for output
       outfile=open('D:\\python.txt','w')
   # Write data to the file
     outfile.write("Bill Clinton\n")
     outfile.write("George Bush\n")
      outfile.write("Barack Obama")
      print(outfile)
       # Close the output file
     outfile.close()
main()
```

Output:



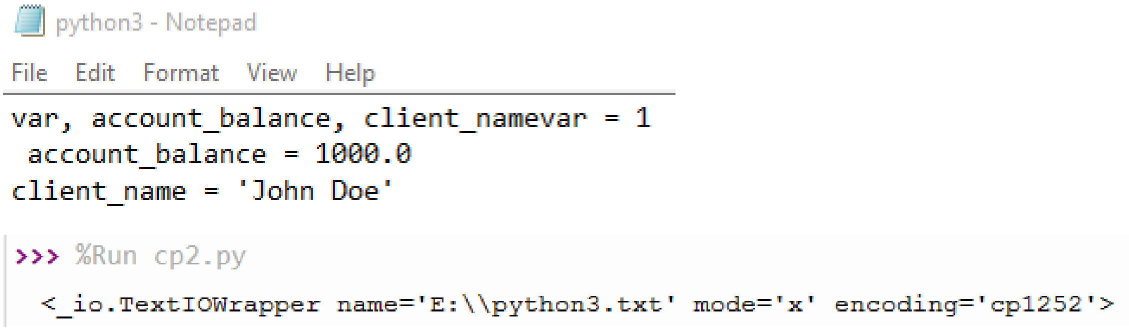
```
>>> %Run cp1.py

  <_io.TextIOWrapper name='E:\\python.txt' mode='w' encoding='cp1252'>
```

```
Bill Clinton
George Bush
Barack Obama
```

2. Code

```
def main():
# Open file for output
    outfile=open('D:\\python.txt','x')
# Write data to the file
    outfile.write("var, account_balance, client_name")
    outfile.write("var     =     1\n     account_balance     =
1000.0\nclient_name = 'John Doe'")
    print(outfile)
    # Close the output file
    outfile.close()
main()
```

Output:



```
var, account_balance, client_namevar = 1
 account_balance = 1000.0
client_name = 'John Doe'
```

```
>>> %Run cp2.py
  <_io.TextIOWrapper name='E:\\python3.txt' mode='x' encoding='cp1252'>
```

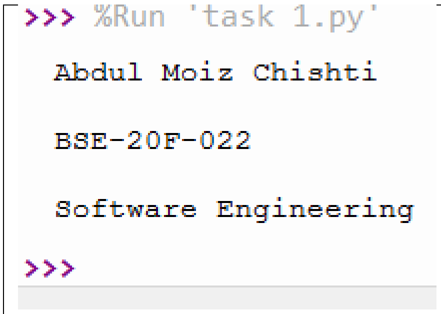
C. Write Python programs for the following:

1. Write a program that create a function called “file_read”, to read an entire text file.

CODE:

```
file_read=open('file_read.txt','r')
for each in file_read:
   print(each)
```

OUTPUT:



```
>>> %Run 'task 1.py'
  Abdul Moiz Chishti

  BSE-20F-022

  Software Engineering

>>>
```

2. Write a program that reads the content and replace any word in a string with different word.

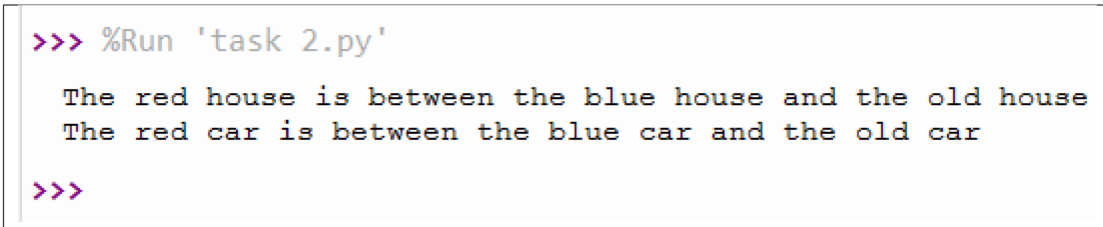
Example:
Replace ‘dog’ with ‘cat’in a sentence

CODE:

```
string="The red house is between the blue house and the old house"
```

```
print(string)

print(string.replace("house","car"))
```

OUTPUT:



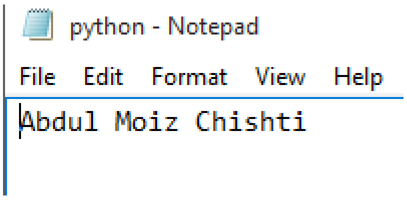
```
>>> %Run 'task 2.py'

  The red house is between the blue house and the old house
  The red car is between the blue car and the old car

>>>
```

3. Write a program that prompts the user for their name. When they respond, write their name to a file called guest.txt.

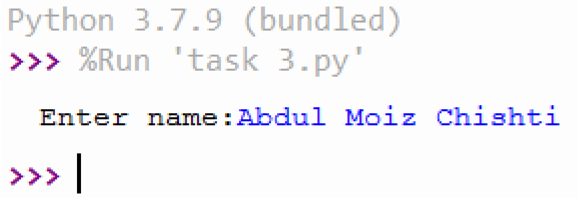
CODE:

```
user=input("Enter name")

file = open(' python.txt','w')

file.write(user)


file.close()
```

OUTPUT:



```
Abdul Moiz Chishti
```



```
Python 3.7.9 (bundled)
>>> %Run 'task 3.py'

  Enter name:Abdul Moiz Chishti

>>>
```