Ex. No: 16

Aim: *Write a script named copyfile.py. This script should prompt the user for the names of two text files. The contents of the first file should be input and written to the second file.*

Theory:

File: Files are named locations on disk to store related information. They are used to permanently store data in a non-volatile memory (e.g., hard disk).

We use files for future use of the data by permanently storing them. When we want to read from or write to a file, we need to open it first. When we are done, it needs to be closed so that the resources that are tied with the file are freed.

Hence, in Python, a file operation takes place in the following order:

- Open a file
- Read or write (perform operation)
- Close the file

Source Code:

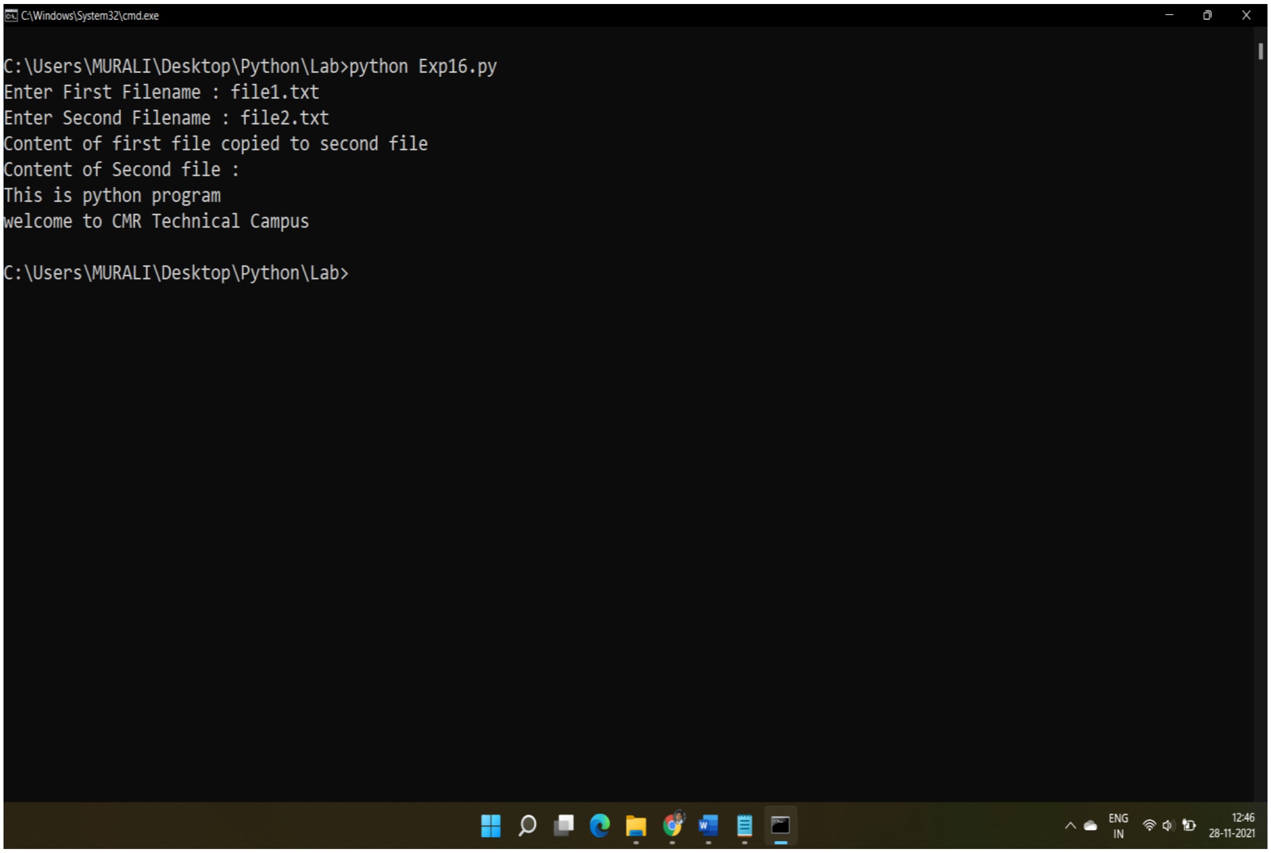
file1.py

```
This is python program
welcome to CMR Technical Campus
```

Exp16.py

```
file1=input("Enter First Filename : ")
file2=input("Enter Second Filename : ")

fn1 = open(file1, 'r')
fn2 = open(file2, 'w')

cont = fn1.readlines()
for i in range(0, len(cont)):
    fn2.write(cont[i])
fn2.close()

print("Content of first file copied to second file ")
fn2 = open(file2, 'r')
cont1 = fn2.read()
print("Content of Second file :")
print(cont1)

fn1.close()
fn2.close()
```

Output:



```
C:\Users\MURALI\Desktop\Python\Lab>python Exp16.py
Enter First Filename : file1.txt
Enter Second Filename : file2.txt
Content of first file copied to second file
Content of Second file :
This is python program
welcome to CMR Technical Campus

C:\Users\MURALI\Desktop\Python\Lab>
```

Ex. No: 17

Aim: *Write a program that inputs a text file. The program should print all of the unique words in the file in alphabetical order.*

Source Code:

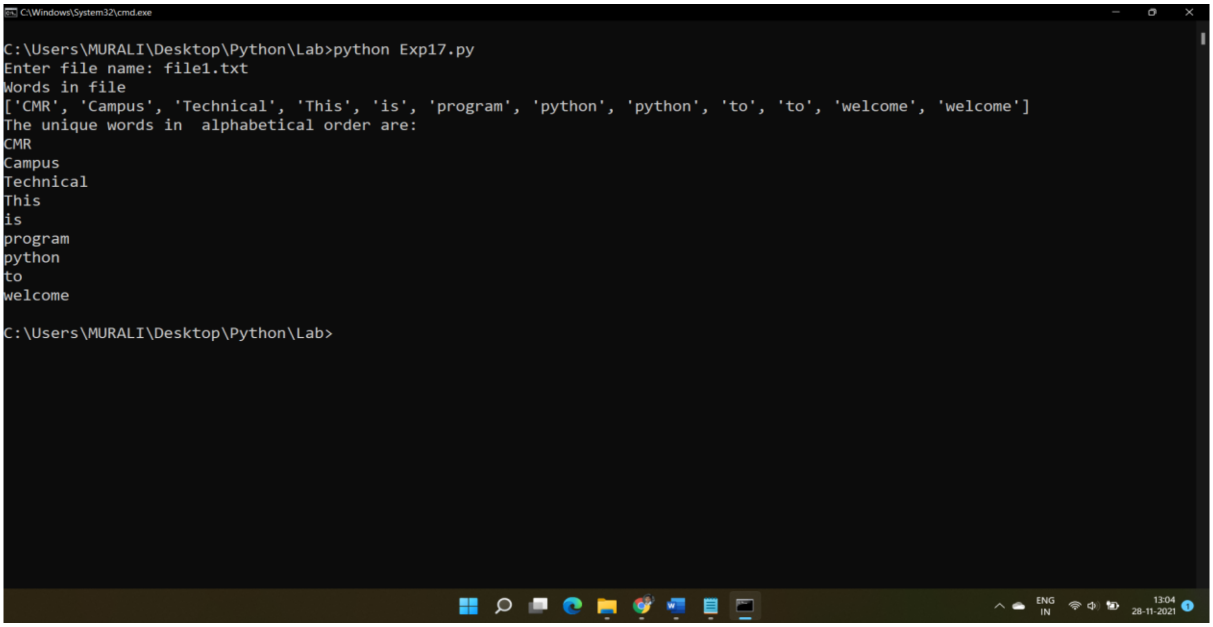
file1.py

```
This is python program
welcome to CMR Technical Campus
welcome to python
```

Exp17.py

```
fname = input("Enter file name: ")
fh = open(fname)
lst = list()
words=[];
for line in fh:
    words += line.split()
words.sort()
print("Words in file are:")
print(words)

print("The unique words in  alphabetical order are:")
for w in words:
    if w in lst:
        continue
    else:
        lst.append(w)
        print(w)
```

Output:



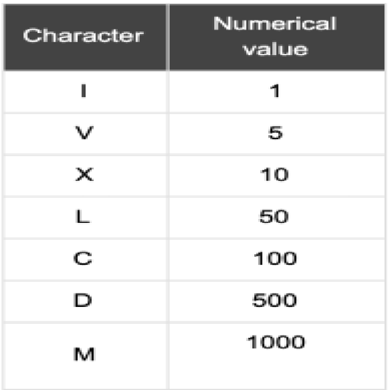
```
C:\Users\MURALI\Desktop\Python\Lab>python Exp17.py
Enter file name: file1.txt
Words in file
['CMR', 'Campus', 'Technical', 'This', 'is', 'program', 'python', 'python', 'to', 'to', 'welcome', 'welcome']
The unique words in  alphabetical order are:
CMR
Campus
Technical
This
is
program
python
to
welcome

C:\Users\MURALI\Desktop\Python\Lab>
```

Ex. No: 18

Aim: *Write a Python class to convert an integer to a roman numeral.*

Theory:

Roman Number: Roman numerals are represented by seven different symbols: I, V, X, L, C, D and M.



Character	Numerical value
I	1
V	5
X	10
L	50
C	100
D	500
M	1000

Rules: Roman numerals are usually written in highest to lowest from left to right except for some special cases where the left character is less than the right character.

Example: ‘IV’ is equivalent to 4 not ‘IIII’. In such cases, subtract the left character value from the right character value. ‘IV’ will be 5-1 = 4, same for ‘IX’ = 10-1 = 9.

Below are the cases –

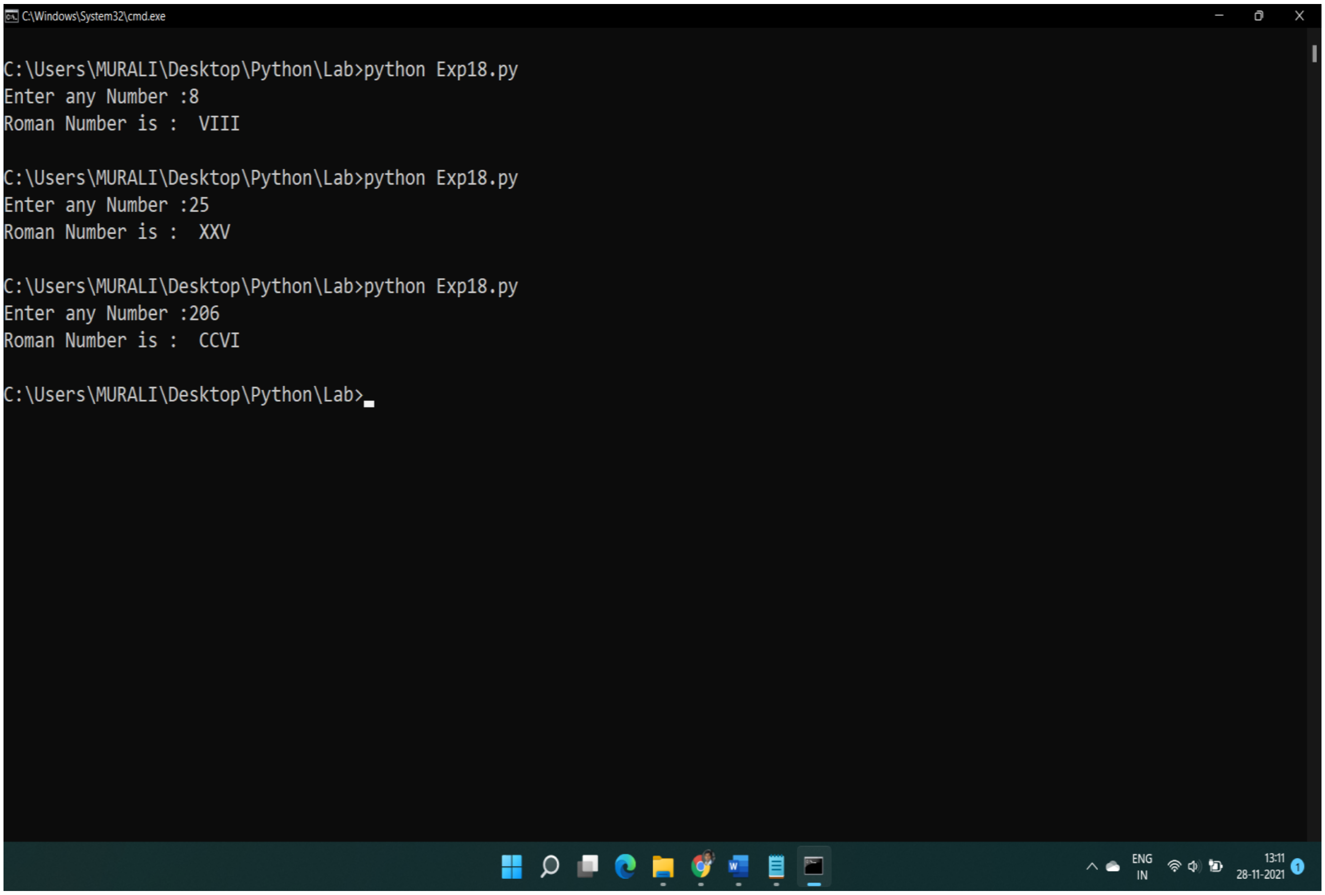
I can be placed before V or X, represents subtract one, so IV (5-1) = 4 and 9 is IX (10-1) =9.
X can be placed before L or C represents subtract ten, so XL (50-10) = 40 and XC (100-10) =90.
C placed before D or M represents subtract hundred, so CD (500-100) =400 and CM (1000-100) =900.

Source Code:

Exp18.py

```
class convert:
    num_map = [(1000, 'M'), (900, 'CM'), (500, 'D'), (400, 'CD'), (100, 'C'), (90, 'XC'), (50, 'L'), (40,
                     'XL'), (10, 'X'), (9, 'IX'), (5, 'V'), (4, 'IV'), (1, 'I')]

    def num2roman(self,num):
        roman = ''
        while num > 0:
            for i, r in self.num_map:
                while num >= i:
                    roman += r
                    num -= i
        return roman


num=int(input("Enter any Number :"))
print("Roman Number is : ",convert().num2roman(num))
```

Output:



```
C:\Users\MURALI\Desktop\Python\Lab>python Exp18.py
Enter any Number :8
Roman Number is :  VIII

C:\Users\MURALI\Desktop\Python\Lab>python Exp18.py
Enter any Number :25
Roman Number is :  XXV

C:\Users\MURALI\Desktop\Python\Lab>python Exp18.py
Enter any Number :206
Roman Number is :  CCVI

C:\Users\MURALI\Desktop\Python\Lab>
```

Ex. No: 19

Aim: *Write a Python class to implement pow (x, n)*

Theory:

Class: A user-defined prototype for an object that defines a set of attributes that characterize any object of the class. The attributes are data members (class variables and instance variables) and methods, accessed via dot notation.

Data member: A class variable or instance variable that holds data associated with a class and its objects.

Method: A special kind of function that is defined in a class definition.

Object: A unique instance of a data structure that's defined by its class. An object comprises both data members (class variables and instance variables) and methods.

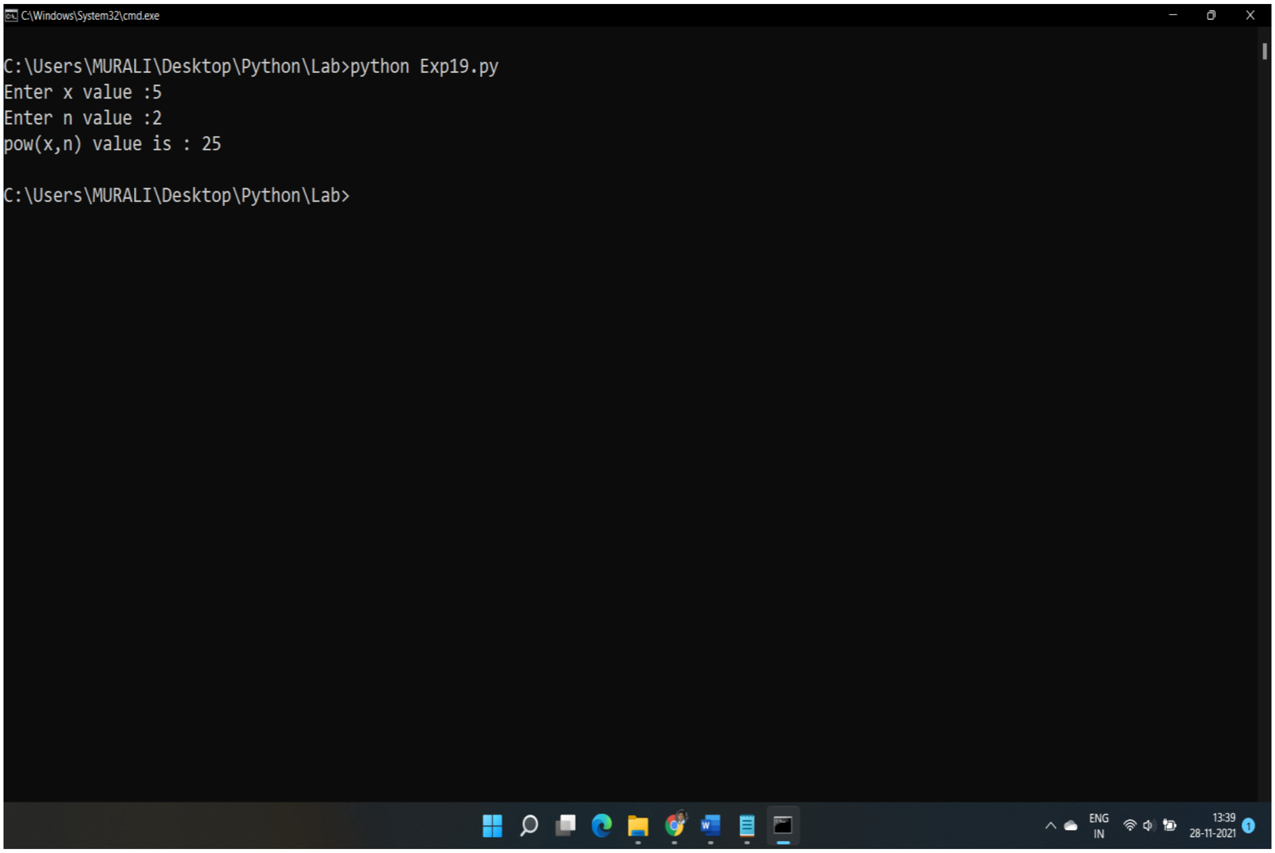
Self-Parameter: The self-parameter is a reference to the current instance of the class, and is used to access variables that belongs to the class.

Source Code:

Exp19.py

```
class py_pow:
  def powr(self, x, n):
      if x==0 or x==1 or n==1:
          return x

      if x==-1:
          if n%2 ==0:
              return 1
          else:
              return -1

      if n==0:
          return 1

      if n<0:
          return 1/self.powr(x,-n)

      val = self.powr(x,n//2)
      if n%2 ==0:
          return val*val
      return val*val*x

x=int(input("Enter x value :"))
n=int(input("Enter n value :"))
print("pow(x,n) value is :",py_pow().powr(x,n))
```

Output:



```
C:\Users\MURALI\Desktop\Python\Lab>python Exp19.py
Enter x value :5
Enter n value :2
pow(x,n) value is : 25

C:\Users\MURALI\Desktop\Python\Lab>
```

Ex. No: 20

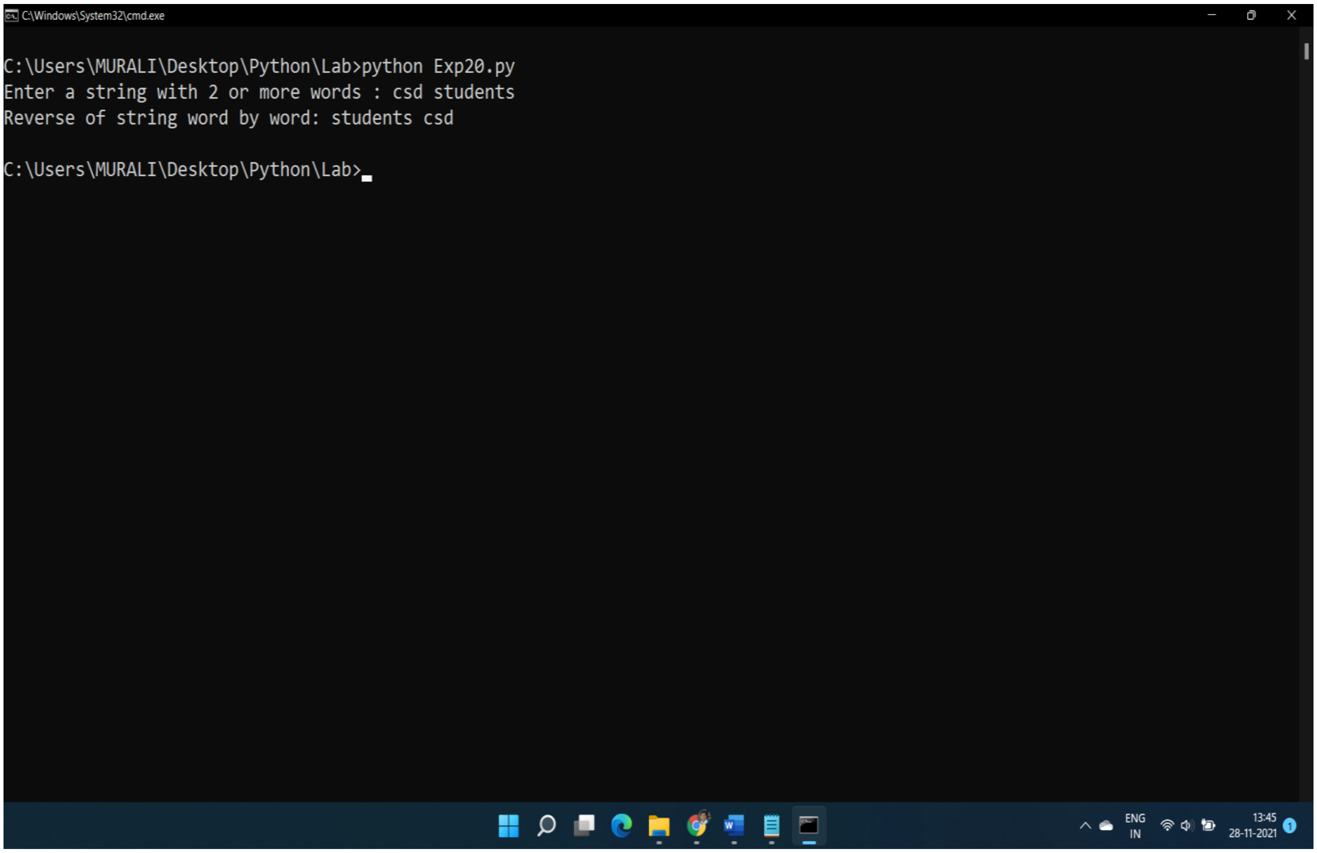
Aim: *Write a Python class to reverse a string word by word.*

Source Code:

Exp20.py

```
class py_reverse:
    def revr(self, strs):
        sp=strs.split()
        sp.reverse()
        res=" ".join(sp)
        return res

str1=input("Enter a string with 2 or more words : ")
print("Reverse of string word by word:",py_reverse().revr(str1))
```

Output:



```
C:\Users\MURALI\Desktop\Python\Lab>python Exp20.py
Enter a string with 2 or more words : csd students
Reverse of string word by word: students csd

C:\Users\MURALI\Desktop\Python\Lab>
```