

a. To implement Mathematical Expressions.

```
import math as m
x=int(input("Enter x: "))
print("Sine of no. is %f" %(m.sin(x)))
print("Cosine of no. is %f" %(m.cos(x)))
print("Tan of no. is %f" %(m.tan(x)))
print("Sine Hyperbolic of no. is %f" %(m.sinh(x)))
print("Cosine Hyperbolic of no. is %f" %(m.cosh(x)))
print("Tan Hyperbolic of no. is %f" %(m.tanh(x)))
print("Ceil of no. is %f" %(m.ceil(x)))
print("Floor of no. is %f" %(m.floor(x)))
print("Absolute value of no. is %f" %(m.fabs(x)))
print("Factorial of no. is %f" %(m.factorial(x)))
print("Square root of no. is %f" %(m.sqrt(x)))
print("Radians of no. is %f" %(m.radians(x)))
print("Pi multiply of no. is %f" %((m.pi)*x))
print("Log of no. is %f" %(m.log(x)))
print("Log10 of no. is %f" %(m.log10(x)))
print("erf of no. is %f" %(m.erf(x)))
print("erfc of no. is %f" %(m.erfc(x)))
print("exp of no. is %f" %(m.exp(x)))
print("Gamma of no. is %f" %(m.gamma(x)))
print("Degrees of no. is %f" %(m.degrees(x)))
```

```
===== RESTART: E:/dms SE IT A 39/Exp1/1.1.py =====
```

```
Enter x: 3
Sine of no. is 0.141120
Cosine of no. is -0.989992
Tan of no. is -0.142547
Sine Hyperbolic of no. is 10.017875
Cosine Hyperbolic of no. is 10.067662
Tan Hyperbolic of no. is 0.995055
Ceil of no. is 3.000000
Floor of no. is 3.000000
Absolute value of no. is 3.000000
Factorial of no. is 6.000000
Sqrt of no. is 1.732051
Radians of no. is 0.052360
Pi of no. is 9.424778
Log of no. is 1.098612
Log10 of no. is 0.477121
erf of no. is 0.999978
erfc of no. is 0.000022
exp of no. is 20.085537
Gamma of no. is 2.000000
Degrees of no. is 171.887339
```

b. To implement Byte Array, Range and String Functions.

```
print("Bytes and Byte Array")
b1=[1,2,3]
print("List to Bytes: " +str(bytes(b1)))
print("List to Bytearray: " +str(bytearray(b1)))
string="Welcome to Python"
strin1=["1","2","3"]
strin="-"
print("String to Bytesbytes: " +str(bytes(string,'utf-8')))
print("String to Bytesbytes: " +str(bytearray(string,'utf-16')))
print()
print("RANGE")
```

```

r1=range(10)
print("Range using one arugment: "+str(r1))
print("Range using one arugment with list: "+str(list(r1)))
r2=range(1,9)
print("Range using two arugment: "+str(r2))
print("Range using two arugment with list: "+str(list(r2)))
r3=range(1,10,2)
print("Range using three arugment: "+str(r3))
print("Range using three arugment with list: "+str(list(r3)))
print()
print("STRING")
print("String to lower: "+str(string.lower()))
print("String to upper: "+str(string.upper()))
print("String is lower: "+str(string.islower()))
print("String is upper: "+str(string.isupper()))
print("String is alphanumeric: "+str(string.isalnum()))
print("Strng is numeric: "+str(string.isnumeric()))
print("String is Alphabet: "+str(string.isalpha()))
print("String has Space: "+str(string.isspace()))
print("String lenget: "+str(len(string)))
print("Split string: "+str(string.split()))
print("String Split: "+str(string.split(" ")))
print("String replace: " +str(string.replace("Python","Django")))
string2=string.split("e")
print("String split at e: "+str(string2))
print("String Join: "+str(strin.join(strin1)))
print("String count o: "+str(string.count("o")))

```

Bytes and Byte Array

```
List to Bytes: b'\x01\x02\x03'
```

```
List to Bytearray: bytearray(b'\x01\x02\x03')
```

```
String to Bytesbytes: b'Welcome to Python'
```

```
String to Bytesbytes: bytearray(b'\xff\xfeW\x00e\x00l\x00c\x00o\x00m\x00e\x00 \x00t\x00
o\x00 \x00P\x00y\x00t\x00h\x00o\x00n\x00')
```

RANGE

```
Range using one arugment: range(0, 10)
```

```
Range using one arugment with list: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
Range using two arugment: range(1, 9)
```

```
Range using two arugment with list: [1, 2, 3, 4, 5, 6, 7, 8]
```

```
Range using three arugment: range(1, 10, 2)
```

```
Range using three arugment with list: [1, 3, 5, 7, 9]
```

STRING

```
String to lower: welcome to python
```

```
String to upper: WELCOME TO PYTHON
```

```
String is lower: False
```

```
String is upper: False
```

```
String is alphanumeric: False
```

```
Strng is numeric: False
```

```
String is Alphabet: False
```

```
String has Space: False
```

```
String lenget: 17
```

```
Split string: ['Welcome', 'to', 'Python']
```

```
String Split: ['Welcome', 'to', 'Python']
```

```
String replace: Welcome to Django
```

```
String split at e: ['W', 'lcom', ' to Python']
```

```
String Join: 1-2-3
```

```
String count o: 3
```

```
^^^
```

Post-Experiments Exercise Programs:

1. Write and implement a python program that accepts an integer(n) and computes The value of n+nn+nnn.

```
n=int(input('Enter a number: '))
temp=str(n)
t1=temp+temp
t2=temp+temp+temp
x=n+int(t1)+int(t2)
print('The result of n+nn+nnn for the entered number is: %d'%x)
```

```
===== RESTART: D:\SFIT\PL\nnn.py =====
Enter a number: 6
The result of n+nn+nnn for the entered number is: 738
>>> |
```

2. Write and implement a python program to compute the greatest common divisor(GCD) of two positive integers.

```
def gcd(a,b):
    if(b==0):
        return a
    else:
        return gcd(b,a%b)
a=int(input("Enter first number:"))
b=int(input("Enter second number:"))
GCD=gcd(a,b)
print("The GCD of the numbers is: ",GCD)
```

```
===== RESTART: D:\SFIT\PL\gcd.py =====
Enter first number:128
Enter second number:6
The GCD of the numbers is:  2
>>> |
```

To implement a python program to demonstrate the following sequences of

Python: a. List b. Tuple c. Dictionary d. Set

```
print("List")
l=[23,24,25,45,74,45,24,67]
print("Original List: "+str(l))
l.insert(0,3)
print("Inserting 3 at position 0 in List: "+str(l))
l.pop()
print("Popped last element from list: "+str(l))
print("Index of 1st 45 element: "+str(l.index(45)))
print("Count element 45: "+str(l.count(45)))
l.sort()
print("Sorted element in list: "+str(l))
l.sort(reverse=True)
print("Reverse sort of list: "+str(l))
l.reverse()
print("Reverse of list: "+str(l))
l1=l.copy()
print("List l="+str(l)+" List l1="+str(l1))
print("Length of List: "+str(len(l)))
l.remove(25)
print("Removed 25 from list: "+str(l))
l.clear()
print("Cleared the list: " +str(l))
print("List of range 2: "+str(list(range(2))))
print("List of range 1 to 11: "+str(list(range(1,11))))
print("List of range 1 to 11 difference of 2: "+str(list(range(1,11,2))))

print()
print("Tuple")
t=(10,25,47,78)
print("Original Tuple: "+str(t))
l=[23,24,25,45,74,45,24,67]
t1=tuple(list(l))
print("List to tuple: " +str(t1))
t3=t+t1
print("Print t+t1: "+str(t3))
print("Print full tuple t1: "+str(t1[:]))
print("Print tuple t1 from 2nd element: "+str(t1[1:]))
print("Print tuple from index 0 to 2: " +str(t1[:3]))
t2=('class','SE IT A')
t3=(t1,t2)
print("Print t1,t2): "+str(t3))
t1=('SEITA ')*4
print("Print a repeating tuple: "+str(t1))
print("Print tuple with range: ")
for i in range(3):
    print(t1)

del(t1)
print("Maximum element in tuple: "+str(max(t)))
print("Minimum element in tuple: "+str(min(t)))
print("Sum of elements in tuple: "+str(sum(t)))
print("Sorted Tuple: "+str(sorted(t)))
```

```

print("Length of tuple: "+str(len(t)))
print("Print tuple in descending order: "+str(t[::-1]))

print()
print("Dictionary")
d1={}
print("Print Blank Dictionary: "+str(d1))
d2={1:'Dominic',2:'Delta'}
print("Print Dictionary d2: "+str(d2))
d3=dict({})
print("Create Dictionary with keyword dict: "+str(d3))
d4=dict([(1,'SEITA'),(2,'SEITB')])
print("Tuple,List to Dictionary: "+str(d4))
print("Print value of key 2 from dict: "+str(d2[2]))
print("Print Keys of dict: "+str(d2.keys()))
print("Print Values of dict: "+str(d2.values()))
print("Print Items of dict: "+str(d2.items()))
d1["Hi"]="dms"
print("Add key Hi with value dms: "+str(d1))
print("Get element 1 from dict: "+str(d2.get(1)))
print("Pop element 1 from dict: "+str(d2.pop(1)))
d2.update({3:'Michael'})
print("Update key 3 with value Michael: "+str(d2))
del d2[2]
print("Delete element with key 2: "+str(d2))
del d4
print(d4)

```

```

print()
print("SET")
s={1,4,6,7}
s1={1,2,4,5,8}
print("Original set: "+str(s))
s.update([4,5])
print("Update s: "+str(s))
s.add(8)
print("Add 8 to s: "+str(s))
s.remove(6)
print("Remove 6 from s: "+str(s))
print("s union s1: "+str(s|s1))
print("s difference s1: "+str(s-s1))
print("s or s1: "+str(s^s1))
print("s intersection s1: "+str(s&s1))
print("s equals s1: "+str(s==s1))
print("s proper set of s1: "+str(s>s1))
print("s not equals s1: "+str(s!=s1))
print("s is subset of s1: "+str(s<=s1))
print("s is proper subset of s1: "+str(s>=s1))
s2=set(s)
print("s is: "+str(s)+" and its copy s2 is: "+str(s2))
f1=frozenset(s)
print("f1 frozenset of s: "+str(f1))
s.clear()
print("s after clearing: "+str(s))

```



```

List
Original List: [23, 24, 25, 45, 74, 45, 24, 67]
Inserting 3 at position 0 in List: [3, 23, 24, 25, 45, 74, 45, 24, 67]
Popped last element from list: [3, 23, 24, 25, 45, 74, 45, 24]
Index of 1st 45 element: 4
Count element 45: 2
Sorted element in list: [3, 23, 24, 24, 25, 45, 45, 74]
Reverse sort of list: [74, 45, 45, 25, 24, 24, 23, 3]
Reverse of list: [3, 23, 24, 24, 25, 45, 45, 74]
List l=[3, 23, 24, 24, 25, 45, 45, 74] List ll=[3, 23, 24, 24, 25, 45, 45, 74]
Length of List: 8
Removed 25 from list: [3, 23, 24, 24, 45, 45, 74]
Cleared the list: []
List of range 2: [0, 1]
List of range 1 to 11: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
List of range 1 to 11 difference of 2: [1, 3, 5, 7, 9]

```

```

Tuple
Original Tuple: (10, 25, 47, 78)
List to tuple: (23, 24, 25, 45, 74, 45, 24, 67)
Print t+tl: (10, 25, 47, 78, 23, 24, 25, 45, 74, 45, 24, 67)
Print full tuple tl: (23, 24, 25, 45, 74, 45, 24, 67)
Print tuple tl from 2nd element: (24, 25, 45, 74, 45, 24, 67)
Print tuple from index 0 to 2: (23, 24, 25)
Print tl,t2: ((23, 24, 25, 45, 74, 45, 24, 67), ('class', 'SE IT A'))
Print a repeating tuple: SEITA SEITA SEITA SEITA
Print tuple with range:
SEITA SEITA SEITA SEITA
SEITA SEITA SEITA SEITA
SEITA SEITA SEITA SEITA
Maximum element in tuple: 78
Minimum element in tuple: 10
Sum of elements in tuple: 160
Sorted Tuple: [10, 25, 47, 78]
Length of tuple: 4
Print tuple in descending order: (10, 25, 47)

```

```

Dictionary
Print Blank Dictionary: {}
Print Dictionary d2: {1: 'Dominic', 2: 'Delta'}
Create Dictionary with keyword dict: {}
Tuple,List to Dictionary: {1: 'SEITA', 2: 'SEITB'}
Print value of key 2 from dict: Delta
Print Keys of dict: dict_keys([1, 2])
Print Values of dict: dict_values(['Dominic', 'Delta'])
Print Items of dict: dict_items([(1, 'Dominic'), (2, 'Delta')])
Add key Hi with value dms: {'Hi': 'dms'}
Get element 1 from dict: Dominic
Pop element 1 from dict: Dominic
Update key 3 with value Michael: {2: 'Delta', 3: 'Michael'}
Delete element with key 2: {3: 'Michael'}

```

```

Traceback (most recent call last):
  File "D:\Google Drive\dms24081999\dominicstrikefighter G-Drive\IMP Documents\St.Francis Institute E
ngineering\Semester-4\Subjects\PL\Practical\2\Code\2.py", line 79, in <module>
    print(d4)
NameError: name 'd4' is not defined
>>>

```

Post-Experiments Exercise Programs:

1. Write a Python program to print a specified list after removing the 0th, 4th and 5th elements.

```
color = ['Red', 'Green', 'White', 'Black', 'Pink', 'Yellow']
color = [x for (i,x) in enumerate(color) if i not in (0,4,5)]
print("After removing the 0th, 4th and 5th elements: "+str(color))
```

```
===== RESTART: D:\SFIT\PL\rmlist.py =====
After removing the 0th, 4th and 5th elements: ['Green', 'White', 'Black']
>>> |
```

2. Write and implement a python program to compute the greatest common divisor(GCD) of two positive integers.

```
import operator
d = {1: 2, 3: 4, 4: 3, 2: 1, 0: 0}
print('Original dictionary : ',d)
sorted_d = sorted(d.items(), key=operator.itemgetter(0))
print('Dictionary in ascending order by value : ',sorted_d)
sorted_d = sorted(d.items(), key=operator.itemgetter(0),reverse=True)
print('Dictionary in descending order by value : ',sorted_d)
```

```
===== RESTART: D:\SFIT\PL\prrr.py =====
Original dictionary :  {1: 2, 3: 4, 4: 3, 2: 1, 0: 0}
Dictionary in ascending order by value :  [(0, 0), (1, 2), (2, 1), (3, 4), (4, 3)]
Dictionary in descending order by value :  [(4, 3), (3, 4), (2, 1), (1, 2), (0, 0)]
>>> |
```

1. To implement a python program to demonstrate the following patterns.
2. To print “Python Programming” when the given range is divisible by 3 and 5, to print “Python” when range is divisible by 3 and to print “Programming” when range is divisible by 5.

```
print("Print * pattern")
n=int(input("Enter n: "))
for i in range(0,n):
    for j in range(0,i+1):
        print("*",end=' ')
    print("")
for i in range(n,0,-1):
    for j in range(0,i-1):
        print("*", end=" ")
    print("")
print("Print powers of 2 pattern")
num=int(input("Enter the number: "))
counter=0
for x in range(0,num):
    for y in range(0,x+1):
        print(counter,end=" ")
        counter=2**(x+1)
    print()
print("\nPrint odd numbers pattern")
num=int(input("Enter the number: "))
counter=1
for x in range(0,num):
    for y in range(0,x+1):
        print(counter,end=" ")
        counter=counter+2
    print()
print("\nPrint ABC pattern")
lastNumber=int(input("Enter last number: "))
asciiNumber=65
for i in range(0,lastNumber):
    for j in range(0,i+1):
        character=chr(asciiNumber)
        print(character,end=" ")
        asciiNumber+=1
    print("")
print("\nPrint Python Programming pattern")
n=int(input("Enter a number: "))
for PythonProgramming in range(n):
    if PythonProgramming%3==0 and PythonProgramming%5==0:
        print("Python Programming")
        continue
    elif PythonProgramming%3==0:
        print("Python")
        continue
    elif PythonProgramming%5==0:
        print("Programming")
        continue
    print("Python-3 Programming")
```


OUTPUT:

```
Print * pattern
Enter n: 5
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*

Print powers of 2 pattern
Enter the number: 4
0
2 4
4 8 8
8 16 16 16

Print odd numbers pattern
Enter the number: 4
1
3 5
7 9 11 |
13 15 17 19

Print ABC pattern
Enter last number: 6
A
B C
D E F
G H I J
K L M N O
P Q R S T U

Print Python Programming pattern
Enter a number: 18
Python Programming
Python-3 Programming
Python-3 Programming
Python
Python-3 Programming
Programming
Python
Python-3 Programming
Python-3 Programming
Python
Programming
Python-3 Programming
Python
Python-3 Programming
Python-3 Programming
Python Programming
Python-3 Programming
Python-3 Programming
```

Post-Experiments Exercise Programs:

1. Write a Python program to print multiplication table of a number given by user.

```
n=int(input("Enter number to print tables: "))
```

```
for i in range(1,11):
```

```
    print(str(n)+" x "+str(i)+" = "+str(n*i))
```

```
Enter number to print tables: 9
```

```
9 x 1 = 9
```

```
9 x 2 = 18
```

```
9 x 3 = 27
```

```
9 x 4 = 36
```

```
9 x 5 = 45
```

```
9 x 6 = 54
```

```
9 x 7 = 63
```

```
9 x 8 = 72
```

```
9 x 9 = 81
```

```
9 x 10 = 90
```

```
>>> |
```

2. Write a Python program to print Fibonacci series of 10 numbers.

```
n=int(input("Enter a number: "))
```

```
n1 = 0
```

```
n2 = 1
```

```
count = 0
```

```
if n<= 0:
```

```
    print("Please enter a positive integer")
```

```
elif n== 1:
```

```
    print("Fibonacci sequence upto "+str(n)+" terms: ")
```

```
    print(n1)
```

```
else:
```

```
    print("Fibonacci sequence upto "+str(n)+" terms: ")
```

```
    while count < n:
```

```
        print(n1)
```

```
        n3 = n1 + n2
```

```
        n1 = n2
```

```
        n2 = n3
```

```
        count += 1
```

```
Enter a number: 10
```

```
Fibonacci sequence upto 10 terms:
```

```
0
```

```
1
```

```
1
```

```
2
```

```
3
```

```
5
```

```
8
```

```
13
```

```
21
```

```
34
```

```
>>> |
```

Code:

Generate One Time Password

```
import math, random          #import math and random package
def genOTP() :                #define otp function
    s1 = '0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ'    #initialize s1
    OTP = ""
    length = len(s1)          #calculate length of string s1
    for i in range(6) :       #for i in 0 to 5 for 6 random values
        OTP += s1[math.floor(random.random()* length)]    #get random character or number
    return OTP                #return variable OTP
print("OTP of length 6:", genOTP())    #call the function to generate otp
```

Towers of Hanoi

```
print("\nTowers of Hanoi")
#To implement Towers of Hanoi
def hanoi(disks, source, auxiliary, target):    #define hanoi function
    if disks == 1:
        print('Move disk 1 from peg {} to peg {}'.format(source, target))
        return
    hanoi(disks - 1, source, target, auxiliary)
    print('Move disk {} from peg {} to peg {}'.format(disks, source, target))
    hanoi(disks - 1, auxiliary, source, target)
try:
    disks = int(input('Enter number of disks: '))
    hanoi(disks, 'A', 'B', 'C')
except RecursionError:
    print("You have entered negative number")
```

Binary Search Algorithm

```
print("\nBinary Search")
#To implement Binary Search Algorithm
def binarysearch(sorted_list, length, key):
    start = 0
    end = length-1
    while start <= end:
        mid = int((start + end)/2)
        if key == sorted_list[mid]:
            print("Entered number %d is present at position: %d" % (key, mid))
            return -1
        elif key < sorted_list[mid]:
            end = mid - 1
        elif key > sorted_list[mid]:
            start = mid + 1
    print("\nElement not found!")
    return -1
lst = []
size = int(input("Enter size of list: "))
for n in range(size):
    numbers = int(input("Enter any number: "))
    lst.append(numbers)
lst.sort()
print('The list will be sorted, the sorted list is:', lst)
x = int(input("Enter the number to search: "))
binarysearch(lst, size, x)
```

To implement:

i. Factorial of a no(Recursion)

```
def fact(x):  
    if x==0:  
        return(1)  
    else :  
        return(x*fact(x-1))
```

ii. Reverse of a No. iii. Palindrome of a No.

```
def rev(x):  
    s=0  
    while x!=0:  
        r=x%10                      #remainder of x/10  
        x=x//10                      #quotient without decimals after point  
        s=s*10+r  
    return(s)
```

iv. Sum of n nos

```
def sum(x):  
    s=0  
    for i in range(1,x+1):  
        s=s+i  
    return(s)  
  
while(True):  
    s=input("\nPress q to quit c to continue: ")  
    s=str(s)  
    if s=='q':  
        break  
    elif s=='c':  
        print("***MENU**")  
        print("1.Factorial")  
        print("2.Reverse of no")  
        print("3.Pallindrome")  
        print("4.Sum of n nos")  
        print("5.Exit")  
        ch=int(input("Enter a choice: "))  
        if ch==1:  
            x=int(input("To find factorial, enter a no.: "))  
            if x<0:  
                print("Invalid")  
            else:  
                y=fact(x)  
                print(y)  
        elif ch==2:  
            y=int(input("To find reverse, enter a no.: "))  
            print("Reverse is: ",rev(y))  
        elif ch==3:  
            t=int(input("To find pallindrome, enter a no.: "))  
            p=rev(t)  
            if p==t:  
                print("Its a Pallindrome")  
            else:  
                print("Its not a pallindrome")  
        elif ch==4:  
            n=int(input("To find sum of n nos, enter a no.: "))  
            print("Sum of n nos: ",sum(n))  
        elif ch==5:
```

```
break
else:
    print("Invalid try again")
```

Output:

OTP of length 6: bqju13

Towers of Hanoi

Enter number of disks: 3

Move disk 1 from peg A to peg C.

Move disk 2 from peg A to peg B.

Move disk 1 from peg C to peg B.

Move disk 3 from peg A to peg C.

Move disk 1 from peg B to peg A.

Move disk 2 from peg B to peg C.

Move disk 1 from peg A to peg C.

Binary Search

Enter size of list: 5

Enter any number: 3

Enter any number: 6

Enter any number: 4

Enter any number: 1

Enter any number: 8

The list will be sorted, the sorted list is: [1, 3, 4, 6, 8]

Enter the number to search: 6

Entered number 6 is present at position: 3

Press q to quit c to continue: c

MENU

1.Factorial

2.Reverse of no

3.Pallindrome

4.Sum of n nos

5.Exit

Enter a choice: 1

To find factorial, enter a no.: 5

120

Press q to quit c to continue: c

MENU

1.Factorial

2.Reverse of no

3.Pallindrome

4.Sum of n nos

5.Exit

Enter a choice: 2

To find reverse, enter a no.: 658

Reverse is: 856

Press q to quit c to continue: c

****MENU****

1.Factorial

2.Reverse of no

3.Pallindrome

4.Sum of n nos

5.Exit

Enter a choice: 3

To find pallindrome, enter a no.: 1331

Its a Pallindrome

Press q to quit c to continue: c

****MENU****

1.Factorial

2.Reverse of no

3.Pallindrome

4.Sum of n nos

5.Exit

Enter a choice: 3

To find pallindrome, enter a no.: 1332

Its not a pallindrome

Press q to quit c to continue: c

****MENU****

1.Factorial

2.Reverse of no

3.Pallindrome

4.Sum of n nos

5.Exit

Enter a choice: 4

To find sum of n nos, enter a no.: 6

Sum of n nos: 21

Press q to quit c to continue: q

>>>

Post-Experiments Exercise Programs:

1. Write a Python program to convert Decimal to Binary using Recursion.

```
def convertToBinary(n):
    if n > 1:
        convertToBinary(n//2)
    print(n % 2,end = "")
dec = int(input("Enter a decimal number: "))
convertToBinary(dec)
```

Output:

```
xp\1.py
Enter a decimal number: 13
1101
>>> |
```

2. Write a Python program to display calendar using Functions.

```
import calendar
print ("The calender of year 2019 is: ")
print (calendar.calendar(2019,2,1,8))
```

Output:

```
2019

January
Mo Tu We Th Fr Sa Su
    1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31

February
Mo Tu We Th Fr Sa Su
        1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28

March
Mo Tu We Th Fr Sa Su
        1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31

April
Mo Tu We Th Fr Sa Su
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30

May
Mo Tu We Th Fr Sa Su
        1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31

June
Mo Tu We Th Fr Sa Su
        1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30

July
Mo Tu We Th Fr Sa Su
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31

August
Mo Tu We Th Fr Sa Su
        1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31

September
Mo Tu We Th Fr Sa Su
        1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30

October
Mo Tu We Th Fr Sa Su
    1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31

November
Mo Tu We Th Fr Sa Su
        1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30

December
Mo Tu We Th Fr Sa Su
        1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30 31
```

Code:

```
class CollegeName:#class containing static method
    college = "SFIT"
    @staticmethod
    def static():#static method
        print ("College Name is {}".format(CollegeName.college))
class Record:#outer class declaration
    class StudentMarks:#inner class declaration
        def __init__(self,roll=0,name=0,age=0,gender=0,marks1 = 0,marks2 =
0,marks3=0,marks4=0,marks5=0):
            self.roll=int(input('Enter your Roll Number: '))
            self.name=input('Enter your Name: ')
            self.age=int(input('Enter your Age: '))
            self.gender=input('Enter your Gender: ')
            self.LDMarks = int(input('Enter your Logic Design marks: '))
            self.DBMSMarks = int(input('Enter your DBMS marks: '))
            self.PCOMMarks = int(input('Enter your PCOM marks: '))
            self.AM3Marks = int(input('Enter your Applied Maths3 marks: '))
            self.DSAMarks = int(input('Enter your DSA marks: '))
        def getData(self):
            print("Marks1: {}, Marks2: {}, Marks3: {}, Marks4: {}, Marks5:
{}".format(self.LDMarks,self.DBMSMarks,self.PCOMMarks,self.AM3Marks,self.DSAMarks))
        def display_record(self):
            print ("Your Roll number is",self.roll)
            print ("Your Name is",self.name)
            print ("Your Age is",self.age)
            print ("Your Gender is",self.gender)
            CollegeName.static()
        def calculate_percentage(self):
            total=self.LDMarks+self.DBMSMarks+self.PCOMMarks+self.AM3Marks+self.DSAMarks
            avg=total/5
            print("Total marks=",total)
            print("Average of Marks:=",avg)
            print("Percentage of marks=",(total/500)*100)
            print ("Hello Mr./Ms.",self.name)
            if(avg>=75):
                print ("You got Distinction class")
            elif(avg>=60):
                print ("You got First class")
            elif(avg>=50):
                print ("You got Second class")
            elif(avg>=40):
                print ("You got Passed")
            else:
                print ("You got failed")
print ("Details of Student are:")
r=Record()
s11 = r.StudentMarks()# Create a new StudentMarks object
s11.getData() # Call getData() function
```

```

s11.display_record() #Calling class method
s11.calculate_percentage() # Call calculate_percentage() function
print(id(s11),'\\n')#points to memory location of objects
s12 = r.StudentMarks()# Create a new StudentMarks object
s12.getData() # Call getData() function
s12.display_record() #Calling class method
s12.calculate_percentage() # Call calculate_percentage() function
print(id(s12))#points to memory location of objects

```

Output:

```

Details of Student are:
Enter your Roll Number: 39
Enter your Name: Dominic
Enter your Age: 19
Enter your Gender: Male
Enter your Logic Design marks: 50
Enter your DBMS marks: 46
Enter your PCOM marks: 32
Enter your Applied Maths3 marks: 49
Enter your DSA marks: 59
Marks1: 50, Marks2: 46, Marks3: 32, Marks4: 49, Marks5: 59
Your Roll number is 39
Your Name is Dominic
Your Age is 19
Your Gender is Male
College Name is SFIT
Total marks= 236
Average of Marks:= 47.2
Percentage of marks= 47.199999999999996
Hello Mr./Ms. Dominic
You got Passed
2000160947392

Enter your Roll Number: 61
Enter your Name: Evita
Enter your Age: 19
Enter your Gender: Female
Enter your Logic Design marks: 60
Enter your DBMS marks: 55
Enter your PCOM marks: 65
Enter your Applied Maths3 marks: 70
Enter your DSA marks: 68
Marks1: 60, Marks2: 55, Marks3: 65, Marks4: 70, Marks5: 68
Your Roll number is 61
Your Name is Evita
Your Age is 19
Your Gender is Female
College Name is SFIT
Total marks= 318
Average of Marks:= 63.6
Percentage of marks= 63.6
Hello Mr./Ms. Evita
You got First class
2000161442168
>>> |

```

Post-Experiments Exercise Programs:

1. Write a Python class to get all possible unique subsets from a set of distinct integers.

Code:

```
class Sub:
    def f1(self,s1):
        return self.f2([],sorted(s1))
    def f2(self,curr,s1):
        if s1:
            return self.f2(curr,s1[1:])+self.f2(curr+[s1[0]],s1[1:])
        return [curr]
a=[]
n=int(input("Enter number of Elements of list: "))
for i in range(0,n):
    b=int(input("Enter Element: "))
    a.append(b)
print("Subsets: ")
print(Sub().f1(a))
```

Output:

```
Enter number of Elements of list: 3
Enter Element: 1
Enter Element: 2
Enter Element: 3
Subsets:
[[], [3], [2], [2, 3], [1], [1, 3], [1, 2], [1, 2, 3]]
>>> |
```

2. Write a Python class named Circle constructed by a radius and two methods which will compute the area and the perimeter of a circle.

Code:

```
import math as m
class Circle:
    def __init__(self,radius=0):
        self.radius=int(input("Enter the radius: "))
    def perimeter(self):
        print("Perimeter of the circle is: {:.2f}".format(2*m.pi*self.radius))
    def area(self):
        print("Area of the circle is: {:.2f}".format(m.pi*(self.radius**2)))
c=Circle()
c.perimeter()
c.area()
```

Output:

```
Enter the radius: 5
Perimeter of the circle is: 31.42
Area of the circle is: 78.54
>>> |
```

CODE:

```
class Person:                                #define parent class
    def __init__(self, fname, lname):
        print ("Calling Person constructor")
        self.fname=fname
        self.lname=lname
    def display(self):                        #Overridden Method (Method Over riding)
        print ("Overriding display() method in Person Class")
        print("Your full name is: ", self.fname, self.lname)
    def e_details(self):
        print ("---Calling Parent method using Child object---")
        self.empid=input("Enter your employee ID: ")
        self.empname=input("Enter your employee name: ")
        self.dept=input("Enter your department details: ")
        self.city=input("Enter your city: ")
        print (" ")
        print ("Employee details are:\n")
        print ("Your employee ID is ",self.empid)
        print ("Your name is ",self.empname)
        print ("Your department is ",self.dept)
        print ("Your city is ",self.city)
    def salary_sum(self, a=None,b=None,c=None):
        print (" ")
        print ("---Method overloading with a=10000,b=20000 and c=30000---")
        if a is not None and b is not None and c is not None :
            print ('Sum of a,b and c is',a+b+c)
        elif a is not None and b is not None:
            print ('Sum of a and b is',a+b)
        else:
            print ('Sum not possible..Value of a is',a)

# Child1 with nothing extra
class Employee(Person):                      # define child class (Inheritance)
    pass
e = Employee("Flora", "Silveira")           # instance of child
e.display()                                 # calls Overridden method
e.salary_sum(10000,20000,30000)             # Method overloading with three parameters
e.salary_sum(10000,20000)                   # Method overloading with two parameters
e.salary_sum(10000)                         # Method overloading with one parameters

# Part2.....# Child2 with overriding method
class Employee1(Person):                    # define child class (Inheritance)
    def __init__(self):
        print ("Calling child constructor")
    def display(self):                      #Overridden Method (Method Overriding)
        print (" ")
        print ("---Method Overriding: Overriding display method in child class---")
    def c_details(self):
        print ("\n---Calling using child object---")
        self.compname=input("Enter Your Company Name -: ")
        self.caddress=input("Enter Your Company address-: ")
        print (" ")
        print ("Your Company Details Are:")
        print ("Your Company name is",self.compname)
```

```

    print ("Your Company Address is",self.caddress)
e1 = Employee1()                # instance of child
e1.c_details()                  # calls parent's method
e1.e_details()                  # calls parent's method
e1.display()                    # calls Overridden method
e1.salary_sum(10000,20000,30000) # Method overloading with three parameters
e1.salary_sum(10000,20000)       # Method overloading with two parameters
e1.salary_sum(10000)             # Method overloading with one parameters

```

```

# Part3.....# Child3 with modified __init__ using super()
class Employee2(Person):        # define child class (Inheritance)
    def __init__(self,fname,lname,age):
        print ("Calling child constructor")
        self.age=age
        super().__init__(fname,lname)
    def display(self):            #Overridden Method (Method Overriding)
        print (" ")
        print ("---Method Overriding: Overriding display method in child class---")
        print("Your full name is: ", self.fname, self.lname)
        print("Your age is: ", self.age)
e2 = Employee2("Dominic","Silveira",19) # instance of child
e2.e_details()                        # calls parent's method
e2.display()                          # calls Overridden method

```

```

# PartIV..... Second class .... use of common method in polymorphism
class Professional:              # define parent class
    def __init__(self, fname1, lname1, Profession):
        print ("Calling Professional constructor")
        self.fname1=fname1
        self.lname1=lname1
        self.Profession=Profession
    def display(self):            #Overridden Method (Method Over riding)
        print ("Overriding display() method in Professional Class")
        print("Your full name is: ", self.fname1, self.lname1)
        print("Your Profession is: ", self.Profession)

```

```

def common_method(obj1):
    obj1.display()

```

```

e = Employee2("Evita", "Rodrigues",19) # instance of child of person class
pr1=Professional("Bhagya","Doshi","Vice General")
common_method(e)
print("\n")
common_method(pr1)

```


OUTPUT:

```
Calling Person constructor
Overriding display() method in Person Class
Your full name is:  Flora Silveira

---Method overloading with a=10000,b=20000 and c=30000---
Sum of a,b and c is 60000

---Method overloading with a=10000,b=20000 and c=30000---
Sum of a and b is 30000

---Method overloading with a=10000,b=20000 and c=30000---
Sum not possible..Value of a is 10000
Calling child constructor

---Calling using child object---
Enter Your Company Name -: Amazon
Enter Your Company address-: Delhi

Your Company Details Are:
Your Company name is Amazon
Your Company Address is Delhi
---Calling Parent method using Child object---
Enter your employee ID: 171032
Enter your employee name: Dominic
Enter your department details: INFT
Enter your city: Mumbai

Employee details are:

Your employee ID is 171032
Your name is Dominic
Your department is INFT
Your city is Mumbai

---Method Overriding: Overriding display method in child class---

---Method overloading with a=10000,b=20000 and c=30000---
Sum of a,b and c is 60000

---Method overloading with a=10000,b=20000 and c=30000---
Sum of a and b is 30000

---Method overloading with a=10000,b=20000 and c=30000---
Sum not possible..Value of a is 10000
Calling child constructor
Calling Person constructor
---Calling Parent method using Child object---
Enter your employee ID: 171034
Enter your employee name: Evita
Enter your department details: INFT
Enter your city: Mumbai

Employee details are:

Your employee ID is 171034
Your name is Evita
Your department is INFT
Your city is Mumbai

---Method Overriding: Overriding display method in child class---
Your full name is:  Dominic Silveira
Your age is: 19
\\ \\ \\
```

```
Calling child constructor
Calling Person constructor
---Calling Parent method using Child object---
Enter your employee ID: 171032
Enter your employee name: Dominic
Enter your department details: INFT
Enter your city: Mumbai

Employee details are:

Your employee ID is 171032
Your name is Dominic
Your department is INFT
Your city is Mumbai

---Method Overriding: Overriding display method in child class---
Your full name is: Dominic Silveira
Your age is: 19
Calling child constructor
Calling Person constructor
Calling Professional constructor

---Method Overriding: Overriding display method in child class---
Your full name is: Evita Rodrigues
Your age is: 19

Overriding display() method in Professional Class
Your full name is: Bhagya Doshi
Your Profession is: Vice General
~ ~ ~ |
```

POST EXPERIMENT PROGRAM:

Write a Python program to declare a class Calculate which calculates the Area of Circle, Triangle and Rectangle (Use Method Overloading).

CODE:

```
import math as m
class Calculate:
    def area(self,a=0,b=0,c=0):
        if(a!=0 and b!=0 and c!=0):
            s=(a+b+c)/2
            a1=m.sqrt(s*(s-a)*(s-b)*(s-c))
            print("Area of the Triangle is: ",a1)
        elif(a!=0 and b!=0 ):
            a2=a*b
            print("Area of the Rectangle is: ",a2)
        elif(a!=0):
            a3=m.pi*a**2
            print("Area of the Circle is: ",a3)
        else:
            print("Enter one or two int or float arguments!")
c=Calculate()
c.area(5)
c.area(10,5)
c.area(0.5,5)
```

OUTPUT:

```
Area of the Circle is:  78.53981633974483
Area of the Rectangle is:  50
Area of the Rectangle is:  2.5
>>>
```

Experiment – 7: Python program to implement MS-Excel and Python Connectivity.

CODE:

```
import sys
from collections import namedtuple
# Save a book to file
def SaveBook(book):
    f = open("EmployeeData.csv", "a+")
    f.write(book[0]+' '+book[1]+' '+book[2]+' '+book[3]+'\n')
    f.close()
# Retrieve a single employee record from the database
def GetRecord(input_id):
    with open("EmployeeData.csv", "r") as f:
        for line in f:
            line = line.rstrip()
            PID, Name, Email, department = line.split(",")
            if(PID == input_id):
                return line
# Prompts the user for adding book details
def AddBook():
    Book = namedtuple("ExcelBook", "PID,Name,Email,Department")
    PID = input("Enter the ID: ")
    Name = input("Enter Name of the Employee: ")
    Email = input("Enter the Email of Employee: ")
    department = input("Enter the department: ")
    newBook = Book(PID, Name, Email, department)
    SaveBook(newBook)
    print("Employee was added successfullly")
# Fetches the getrecord() when a employee id is given as input
def DisplayBook():
    input_id = input("Enter the ID of Employee to display: ")
    PID, Name, Email, department = GetRecord(input_id).split(",")
    if (PID == input_id):
        print('{0: <8}'.format(PID) + '{0: <12}'.format(Name) +
              '{0: <30}'.format(Email) + '{0: <6}'.format(department))
# To delete a record from the book
def DeleteBook():
    input_id = input("Enter the ID of the Employee to delete: ")
    f = open("EmployeeData.csv", "r+")
    d = f.readlines()
    f.seek(0)
    for line in d:
        record = line.rstrip()
        PID, Name, Email, department = record.split(",")
        if PID != input_id:
            f.write(line)
    f.truncate()
    f.close()
    print("Book was successfully deleted from the database!")
# To view all the Employees in the database
def ViewBooks():
    with open("EmployeeData.csv", "r") as f:
        for line in f:
            line = line.rstrip()
```

```

        PID, Name, Email, department = line.split(",")
    if not line:
        continue
    print('{0: <8}'.format(PID) + '{0: <12}'.format(Name) +
          '{0: <30}'.format(Email) + '{0: <6}'.format(department))

# Function to search for an Employee in the database
def Search():
    criteria = input("Enter a search criteria: ")
    with open("EmployeeData.csv", "r") as f:
        for line in f:
            line = line.rstrip()
            if line.upper().find(criteria.upper()) != -1:
                PID, Name, Email, department = line.split(",")
                print('{0: <8}'.format(PID) + '{0: <12}'.format(Name) +
                      '{0: <30}'.format(Email) + '{0: <6}'.format(department))

# Function to display the Menu
def DisplayMenu():
    print("\n***CHOOSE AN OPERATION.***")
    print("1. ADD AN EMPLOYEE")
    print("2. DISPLAY EMPLOYEE DETAILS")
    print("3. DELETE AN EMPLOYEE")
    print("4. VIEW EMPLOYEE")
    print("5. SEARCH FOR AN EMPLOYEE")
    print("6. GET TOTAL NUMBER")
    print("7. EXIT")

# Get the total number of Employees
def GetTotal():
    return sum(1 for line in open('EmployeeData.csv'))

choice = 0
while (choice != 7):
    DisplayMenu()
    choice = input("Select an operation (1,2,3,4,5,6,7): ")
    if choice == '7':
        sys.exit()
    if choice == '1':
        AddBook()
    elif choice == '2':
        DisplayBook()
    elif choice == '3':
        DeleteBook()
    elif choice == '4':
        ViewBooks()
    elif choice == '5':
        Search()
    elif choice == '6':
        print(GetTotal())
    else:
        print("Invalid input")

```

OUTPUT:

```
***CHOOSE AN OPERATION.***
1. ADD AN EMPLOYEE
2. DISPLAY EMPLOYEE DETAILS
3. DELETE AN EMPLOYEE
4. VIEW EMPLOYEE
5. SEARCH FOR AN EMPLOYEE
6. GET TOTAL NUMBER
7. EXIT
Select an operation (1,2,3,4,5,6,7): 1
Enter the ID: 61
Enter Name of the Employee: Evita
Enter the Email of Employee: rodriguesevita9@gmail.com
Enter the department: INFT
Employee was added successfullly

***CHOOSE AN OPERATION.***
1. ADD AN EMPLOYEE
2. DISPLAY EMPLOYEE DETAILS
3. DELETE AN EMPLOYEE
4. VIEW EMPLOYEE
5. SEARCH FOR AN EMPLOYEE
6. GET TOTAL NUMBER
7. EXIT
Select an operation (1,2,3,4,5,6,7): 2
Enter the ID of Employee to display: 22
22      Malcolm      malcolmdsouza376@gmail.com      INFT

***CHOOSE AN OPERATION.***
1. ADD AN EMPLOYEE
2. DISPLAY EMPLOYEE DETAILS
3. DELETE AN EMPLOYEE
4. VIEW EMPLOYEE
5. SEARCH FOR AN EMPLOYEE
6. GET TOTAL NUMBER
7. EXIT
Select an operation (1,2,3,4,5,6,7): 3
Enter the ID of the Employee to delete: 37
Book was successfully deleted from the database!

***CHOOSE AN OPERATION.***
1. ADD AN EMPLOYEE
2. DISPLAY EMPLOYEE DETAILS
3. DELETE AN EMPLOYEE
4. VIEW EMPLOYEE
5. SEARCH FOR AN EMPLOYEE
6. GET TOTAL NUMBER
7. EXIT
Select an operation (1,2,3,4,5,6,7): 4
23      Sanket      sanket.dalvi.ssd@gmail.com      CMPN
12      Pratiksha   pratikshamore@gmail.com      INFT
39      Dominic    dms24081999@gmail.com      INFT
67      Gurman     gurmanksokhi@gmail.com      INFT
22      Malcolm    malcolmdsouza376@gmail.com      INFT
61      Evita      rodriguesevita9@gmail.com      INFT

***CHOOSE AN OPERATION.***
1. ADD AN EMPLOYEE
2. DISPLAY EMPLOYEE DETAILS
3. DELETE AN EMPLOYEE
4. VIEW EMPLOYEE
5. SEARCH FOR AN EMPLOYEE
6. GET TOTAL NUMBER
7. EXIT
```


Select an operation (1,2,3,4,5,6,7): 5
Enter a search criteria: 39
39 Dominic dms24081999@gmail.com INFT

CHOOSE AN OPERATION.

1. ADD AN EMPLOYEE
2. DISPLAY EMPLOYEE DETAILS
3. DELETE AN EMPLOYEE
4. VIEW EMPLOYEE
5. SEARCH FOR AN EMPLOYEE
6. GET TOTAL NUMBER
7. EXIT

Select an operation (1,2,3,4,5,6,7): 5
Enter a search criteria: 10

CHOOSE AN OPERATION.

1. ADD AN EMPLOYEE
2. DISPLAY EMPLOYEE DETAILS
3. DELETE AN EMPLOYEE
4. VIEW EMPLOYEE
5. SEARCH FOR AN EMPLOYEE
6. GET TOTAL NUMBER
7. EXIT

Select an operation (1,2,3,4,5,6,7): 6
6

CHOOSE AN OPERATION.

1. ADD AN EMPLOYEE
2. DISPLAY EMPLOYEE DETAILS
3. DELETE AN EMPLOYEE
4. VIEW EMPLOYEE
5. SEARCH FOR AN EMPLOYEE
6. GET TOTAL NUMBER
7. EXIT

Select an operation (1,2,3,4,5,6,7): 7
>>>

CHOOSE AN OPERATION.

1. ADD AN EMPLOYEE
2. DISPLAY EMPLOYEE DETAILS
3. DELETE AN EMPLOYEE
4. VIEW EMPLOYEE
5. SEARCH FOR AN EMPLOYEE
6. GET TOTAL NUMBER
7. EXIT

Select an operation (1,2,3,4,5,6,7): 7
~::~

	A	B	C	D	E
1	23	Sanket	sanket.dalvi.ssd@gmail.com	CMPN	
2	12	Pratiksha	pratikshamore@gmail.com	INFT	
3	39	Dominic	dms24081999@gmail.com	INFT	
4	67	Gurman	gurmanksochi@gmail.com	INFT	
5	22	Malcolm	malcolmdsouza376@gmail.com	INFT	
6	61	Evita	rodriguesevita9@gmail.com	INFT	
7					

POST EXPERIMENT PROGRAMS:

1. Write a Python program to read last two lines of a file.

```
f1 = open("Data.txt", "r")
lines = f1.readlines()
last_lines = lines[-2:]
for line in last_lines:
    print(line, end="")
f1.close()
```

OUTPUT:

```
Enter file name: file1.txt
programming
language
>>> |
```

2. Write a Python program to count the frequency of words in a file.

```
import re
import string
frequency={}
fname=input("Enter file name: ")
document_text=open(fname,'r')
text_string=document_text.read().lower()
match_pattern=re.findall(r'\b[a-z]{1,15}\b',text_string)
for word in match_pattern:
    count=frequency.get(word,0)
    frequency[word]=count+1
frequency_list=frequency.keys()
for words in frequency_list:
    print(words,"->",frequency[words])
```

OUTPUT:

```
Enter file name: file2.txt
welcome -> 1
to -> 1
python -> 2
programming -> 2
language -> 2
is -> 1
the -> 1
most -> 1
popular -> 1
>>>
```

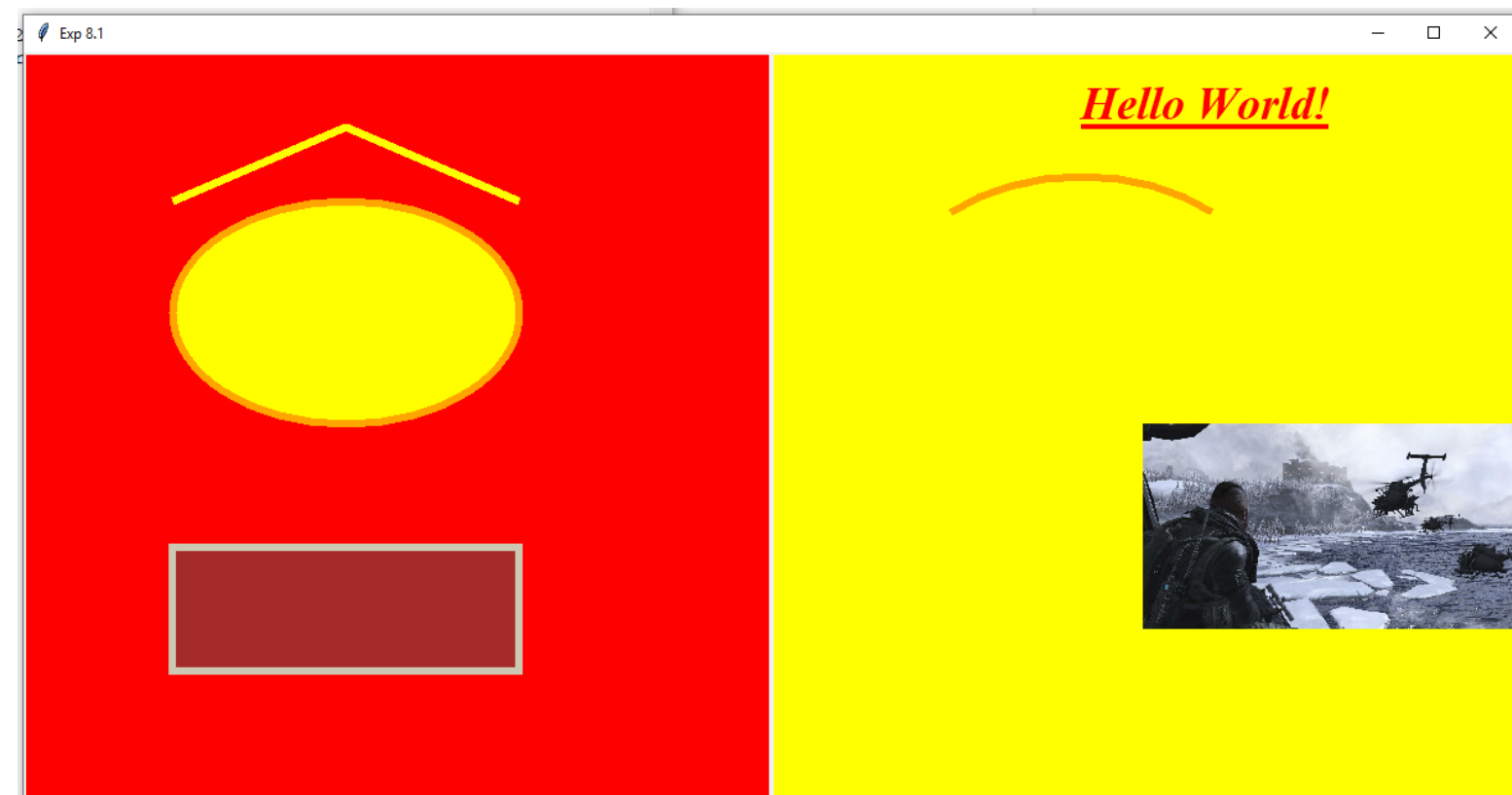
PYTHON EXP8

A) To implement a python program for GUI Canvas Application.

CODE:

```
from tkinter import *
root=Tk()
root.title("Exp 8.1")
c1=Canvas(root,bg="red",height=600,width=600)
c1.pack(side=LEFT)
l=c1.create_line(120,120,260,60,400,120,width=6,fill="yellow")
o=c1.create_oval(120,120,400,300,width=6,fill="yellow",outline="Orange",activefill="green")
r=c1.create_rectangle(120,400,400,500,width=6,fill="brown",outline="cornsilk3",activefill="chocolate")
c2=Canvas(root,bg="yellow",height=600,width=600)
c2.pack(side=RIGHT)
p=c2.create_polygon(320,320,320,420,420,320,width=6,fill="yellow",outline="cornsilk2",activefill="lightblue")
a=c2.create_arc(100,100,400,300,width=6,start=45, extent=90,outline="orange",style="arc")
f=("Times",-37,"bold italic underline")
t=c2.create_text(350,40,text="Hello World!",font=f,fill="red",activefill="green")
file1=PhotoImage(file="1.png",height=300,width=300)
file2=PhotoImage(file="2.png",height=300,width=300)
id=c2.create_image(600,600,anchor=SE,image=file1,activeimage=file2)
root.mainloop()
```

OUTPUT:



B) To implement a python program for GUI Frame Application.

CODE:

```
from tkinter import *
from tkinter import DISABLED,ACTIVE
from tkinter import messagebox
class MyEntry:
    def __init__(self,root):
        self.f=Frame(root,height=440, width=615)
        self.f.propagate(0)
        self.f.pack(side=LEFT)
        self.l1=Label(text='Enter Your PID:')
        self.l2=Label(text='Enter Your Login Password:')
        self.l5=Label(text='Select Your Dept.:')
        self.l3=Label(text='CET Score above 80?')
        self.l4=Label(text='Select Your Optional Courses:')
        self.l5=Label(text='Select Your Division:')
        self.e1=Entry(self.f,width=18,fg="black",bg="gray", font=('arial', 12))
        self.e2=Entry(self.f,width=18,fg="black",bg="gray", font=('arial', 12),show="*")
        self.var=IntVar()
        self.r1=Radiobutton(self.f,text='Yes',variable=self.var,value=1,command=self.disable)
        self.r2=Radiobutton(self.f,text='No',variable=self.var,value=2,command=self.disable)
        self.var1=IntVar()
        self.var2=IntVar()
        self.var3=IntVar()
        self.var4=IntVar()
        self.c1=Checkbutton(self.f,text='Data Structures',variable=self.var1)
        self.c2=Checkbutton(self.f,text='Image Processing',variable=self.var2)
        self.c3=Checkbutton(self.f,text='Logic Design',variable=self.var3)
        self.c4=Checkbutton(self.f,text='Computer Graphics',variable=self.var4)
        self.img=PhotoImage(file="print.png")
        self.b=Button(self.f,text="Print Details",image=self.img,command=self.display)
        self.b.pack()
        self.str1=StringVar()
        self.s1=Spinbox(self.f,values=('A','B','C','D'),textvariable=self.str1,width=15,fg='black',bg='gray',font=('Arial', 12,'bold italic'))
        self.w=Listbox(height=3)
        self.w.insert(1,'INFT')
        self.w.insert(2,'CMPN')
        self.w.insert(3,'EXTC')
        self.w.bind("<Double-Button-1>", self.OnDouble)
        self.w.pack()
        self.l1.place(x=50,y=30)
        self.e1.place(x=210,y=30)
        self.l2.place(x=50,y=60)
        self.e2.place(x=210,y=60)
        self.l3.place(x=50,y=90)
        self.l4.place(x=50,y=130)
        self.r1.place(x=210,y=90)
        self.r2.place(x=270,y=90)
        self.l5.place(x=50,y=160)
        self.c1.place(x=210,y=130)
        self.c2.place(x=290,y=130)
        self.c3.place(x=380,y=130)
        self.c4.place(x=470,y=130)
```

```

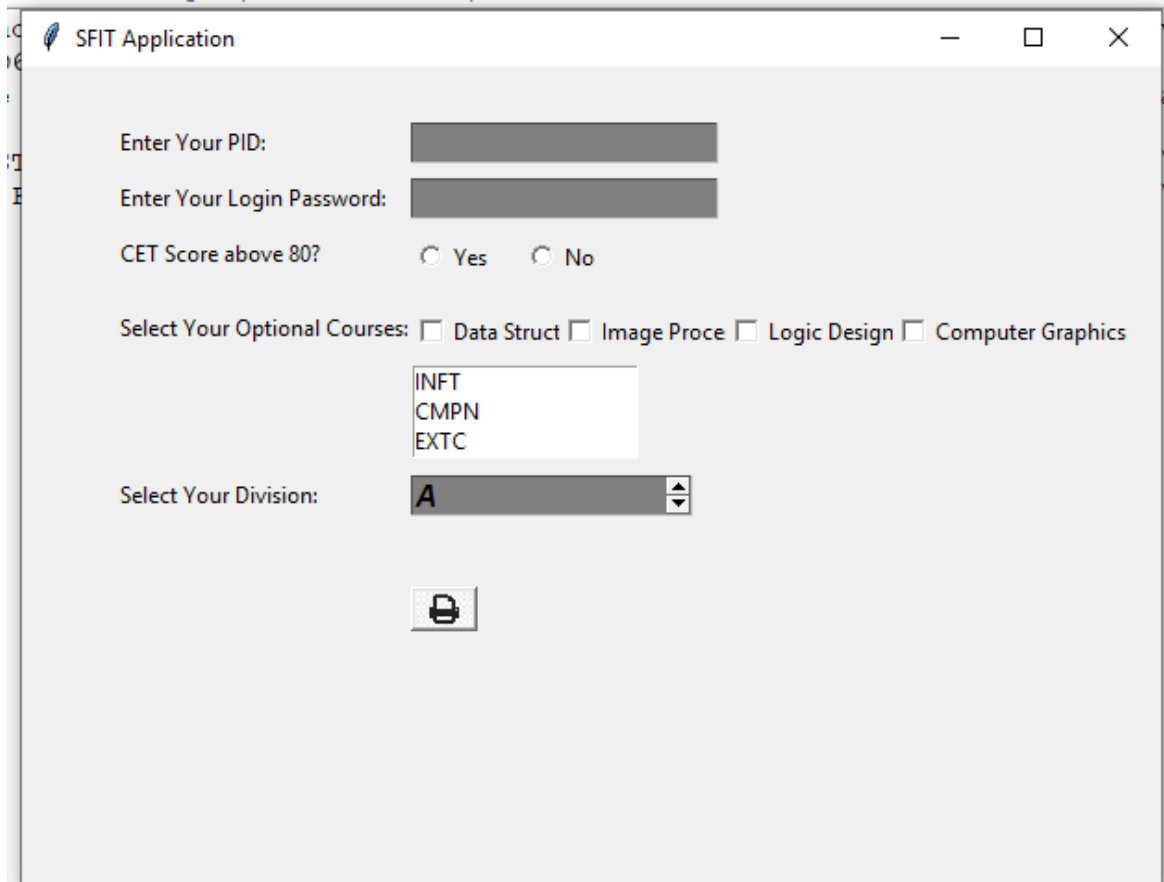
self.b.place(x=210,y=280)
self.w.place(x=210,y=160)
self.l5.place(x=50,y=220)
self.s1.place(x=210,y=220)
def OnDouble(self, event):
    widget = event.widget
    selection=widget.curselection()
    value = widget.get(selection[0])
    messagebox.showinfo("Dept","You've selected "+str(value))
def display(self):
    str1=self.e1.get()
    str2=self.e2.get()
    x1=self.var1.get()
    x2=self.var2.get()
    x3=self.var3.get()
    x4=self.var4.get()
    x5=self.str1.get()
    str6=self.str1.get()
    str5=""
    if x1==1:
        str5+='Data Structures,'
    if x2==1:
        str5+='Image Processing,'
    if x3==1:
        str5+='Logic Design,'
    if x4==1:
        str5+='Computer Graphics,'
    l7=Label(text='Your PID is: '+str1,font=('Verdana',10,'bold'),fg='red').place(x=50,y=310)
    l8=Label(text='Your Password is: '+str2,font=('Verdana',10,'bold'),fg='red').place(x=50,y=340)
    l11=Label(text='Your selected optional course is '+str5,font=('Verdana',10,'bold'),fg='red').place(x=50,y=370)
    l12=Label(text='Your division is '+str6,font=('Verdana',10,'bold'),fg='red').place(x=50,y=400)
def disable(self):
    str1=self.e1.get()
    str2=self.e2.get()
    x1=self.var1.get()
    x2=self.var2.get()
    x3=self.var3.get()
    x4=self.var4.get()
    x=self.var.get()
    str4=""
    if x==1:
        str4+='Your are elligible for engineering courses.'
        self.c1.config(state=ACTIVE)
        self.c2.config(state=ACTIVE)
        self.c3.config(state=ACTIVE)
        self.c4.config(state=ACTIVE)
        self.b.config(state=ACTIVE)
        self.w.config(state=NORMAL)
        self.s1.config(state=NORMAL)
    if x==2:
        str4+='Your are not elligible for engineering courses.'
        self.c1.config(state=DISABLED)
        self.c2.config(state=DISABLED)
        self.c3.config(state=DISABLED)
        self.c4.config(state=DISABLED)

```

```
self.b.config(state=DISABLED)
self.w.config(state=DISABLED)
self.s1.config(state=DISABLED)
l12=Label(text=str4,font=('Verdana',10,'bold'),fg='red').place(x=50,y=250)
```

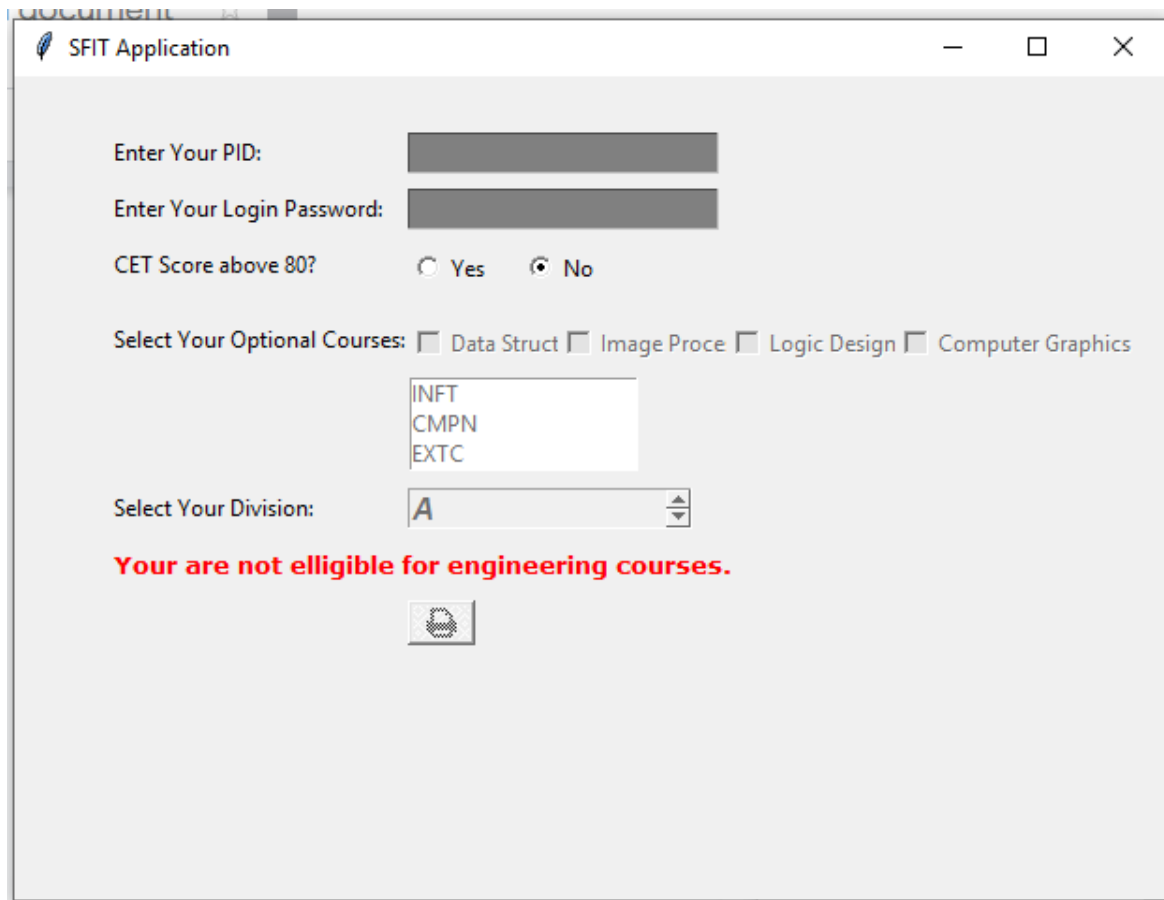
```
root=Tk()#create root window
root.title("SFIT Application") #Giving a title to bar
mb=MyEntry(root)#creating an object to class
root.mainloop()#root window handles the mouse events
```

OUTPUT:



The screenshot shows a window titled "SFIT Application". It contains the following elements:

- "Enter Your PID:" followed by a text entry field.
- "Enter Your Login Password:" followed by a text entry field.
- "CET Score above 80?" with two radio buttons: "Yes" (unselected) and "No" (unselected).
- "Select Your Optional Courses:" followed by four checkboxes: "Data Struct", "Image Proce", "Logic Design", and "Computer Graphics", all of which are unchecked.
- A list box containing the items "INFT", "CMPN", and "EXTC".
- "Select Your Division:" followed by a dropdown menu showing the letter "A".
- A button with a printer icon.



This screenshot shows the same "SFIT Application" window after a validation step. The state is as follows:

- The "CET Score above 80?" radio buttons now have "No" selected.
- The "Data Struct" checkbox is now checked.
- The "Image Proce" checkbox is now checked.
- The "Logic Design" checkbox is now checked.
- The "Computer Graphics" checkbox is now checked.
- The dropdown menu for "Select Your Division:" still shows "A".
- A new red text label, "Your are not elligible for engineering courses.", has been added below the division dropdown.
- The printer icon button remains at the bottom.

SFIT Application

Enter Your PID:


Enter Your Login Password:

CET Score above 80? ☒ Yes ☐ No


Select Your Optional Courses: ☐ Data Struct ☐ Image Proce ☐ Logic Design ☐ Computer Graphics

Select Your Division:

Your are elligible for engineering courses.



Dept

 You've selected CMPN

OK

SFIT Application

Enter Your PID:


Enter Your Login Password:

CET Score above 80? ☒ Yes ☐ No

Select Your Optional Courses: ☒ Data Struct ☐ Image Proce ☒ Logic Design ☐ Computer Graphics

Select Your Division:

Your are elligible for engineering courses.



Your PID is: 123456

Your Password is: 123

Your selected optional course is Data Structures,Logic Design,

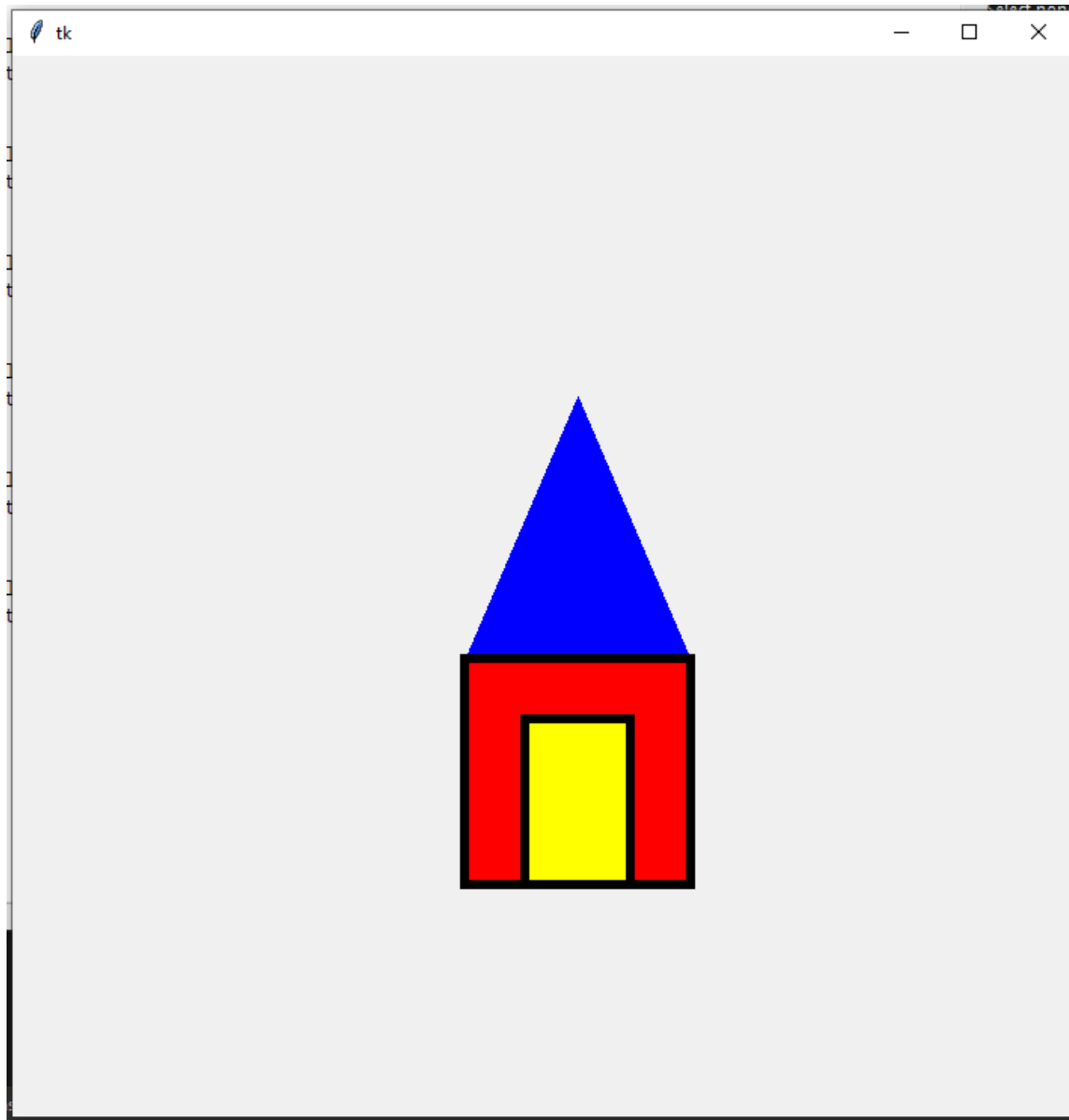
Your division is C

PYTHON POST EXPERIMENT QUESTIONS:

1. Write a Python program to draw the house on a canvas using Tkinter.

```
from tkinter import *
root=Tk()
c=Canvas(root,height=700,width=700)
id=c.create_polygon(375,225,300,400,450,400,width=6,fill="blue")
id=c.create_rectangle(300,400,450,550,width=6,fill="red")
id=c.create_rectangle(340,440,410,550,width=6,fill="yellow")
c.pack()
root.mainloop()
```

OUTPUT:



2. Take a Login and Password from the user in the first Frame and display it in the second Frame on the labels which appears only on clicking OK button and clear both the labels by clicking RESET button.

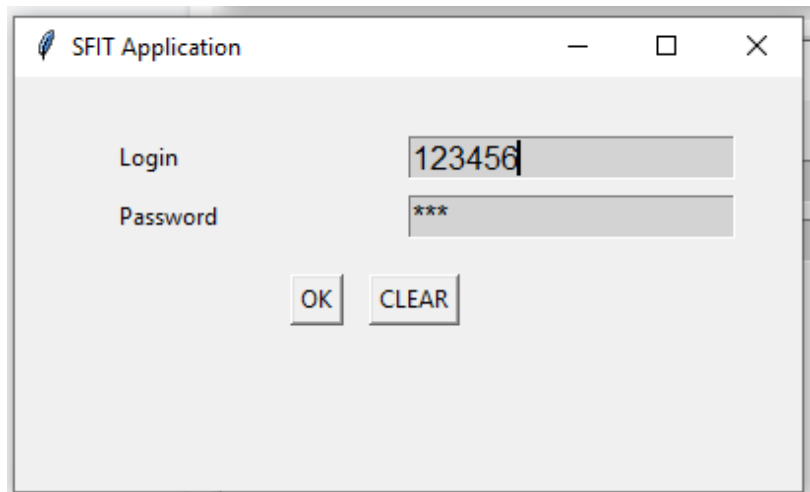
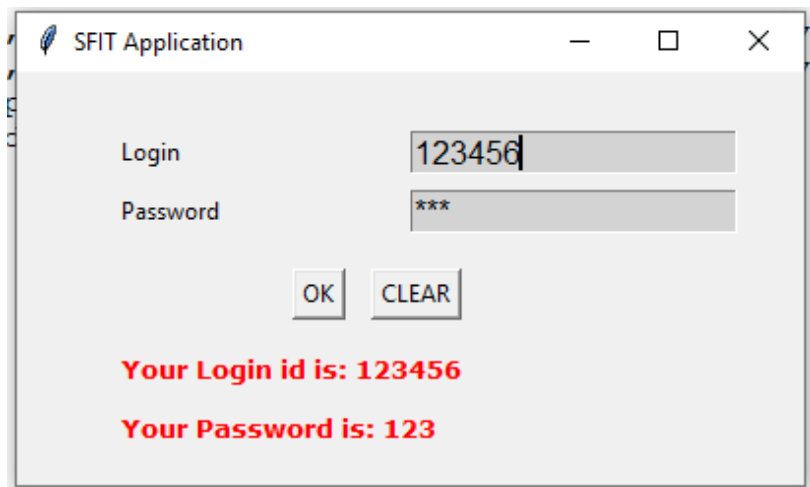
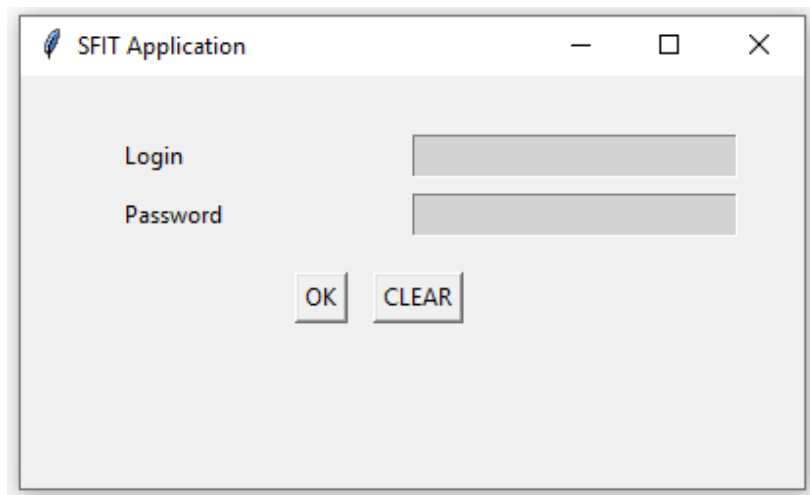
```
from tkinter import *
root = Tk()
frame1 = Frame(root,height=130, width=400)
frame1.pack()
frame2 = Frame(root,height=80, width=400)
frame2.pack()
l1=Label(frame1,text='Login')
l2=Label(frame1,text='Password')
e1=Entry(frame1,width=18,fg="black",bg="lightgray", font=('arial',12))
e2=Entry(frame1,width=18,fg="black",bg="lightgray", font=('arial',12),show="*")
```

```

l1.place(x=50,y=30)
e1.place(x=200,y=30)
l2.place(x=50,y=60)
e2.place(x=200,y=60)
l3=Label(frame2,text="")
l4=Label(frame2,text="")
def display1():
    l3=Label(frame2,text='Your Login id is: '+e1.get(),font=('Verdana',10,'bold'),fg='red').place(x=50,y=10)
    l4=Label(frame2,text='Your Password is: '+e2.get(),font=('Verdana',10,'bold'),fg='red').place(x=50,y=40)
def display2():
    l3=Label(frame2,font=('Verdana',10,'bold'),fg='red',width=100).place(x=50,y=10)
    l4=Label(frame2,font=('Verdana',10,'bold'),fg='red',width=100).place(x=50,y=40)
b1=Button(frame1,text="OK",command=display1)
b2=Button(frame1,text="CLEAR",command=display2)
b1.place(x=140,y=100)
b2.place(x=180,y=100)

```

OUTPUT:



1. Python program to implement CRUD Operations on Python using MySQL Database Connectivity.

```
C:\xampp\mysql\bin>mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 2
Server version: 10.1.33-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database student;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| dms1     |
| information_schema |
| mysql    |
| performance_schema |
| phpmyadmin |
| student  |
| test     |
+-----+
7 rows in set (0.00 sec)

MariaDB [(none)]> use student;
Database changed
MariaDB [student]> create table student(pid int, name varchar(20));
Query OK, 0 rows affected (0.14 sec)

MariaDB [student]> desc student;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| pid   | int(11)       | YES  |     | NULL    |       |
| name  | varchar(20)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.01 sec)

MariaDB [student]> insert into student values(171034, "Evita");
Query OK, 1 row affected (0.02 sec)

MariaDB [student]> insert into student values(171061, "Malcolm");
Query OK, 1 row affected (0.01 sec)

MariaDB [student]> insert into student values(171032, "dms");
Query OK, 1 row affected (0.02 sec)

MariaDB [student]> select * from student;
+-----+-----+
| pid   | name   |
+-----+-----+
| 171034 | Evita  |
| 171061 | Malcolm |
| 171032 | dms    |
+-----+-----+
3 rows in set (0.00 sec)

MariaDB [student]>
```

CODE:

```
import pymysql
con = pymysql.connect(host="localhost", user="root", password="", db="student")
cur = con.cursor()
cur.execute("insert into student(pid, name) values(171086, 'Pratiksha')")
print("Insert Query executed successfully.")
print("Printing all the fields of the table.")
cur.execute("select * from student")
rows = cur.fetchall()
print(rows)
print()
cur.execute("UPDATE student SET pid=171033 WHERE name='dms'")
print("Update Query executed successfully.")
print("Printing all the fields of the table.")
cur.execute("select * from student")
rows = cur.fetchall()
print(rows)
print()
cur.execute("select name from student where pid=171034")
print(cur.fetchone())
print("Select Query executed successfully.")
print()
cur.execute("DELETE FROM student WHERE pid=171061")
print("Delete Query executed successfully.")
con.commit()
print()
cur.execute("create table StudentDetails(roll int, phone int)")
print("StudentDetails table created successfully.")
cur.close()
con.close()
```

OUTPUT:

```
D:\>python mysql.py
Insert Query executed successfully.
Printing all the fields of the table.
((171034, 'Evita'), (171061, 'Malcolm'), (171032, 'dms'), (171086, 'Pratiksha'))

Update Query executed successfully.
Printing all the fields of the table.
((171034, 'Evita'), (171061, 'Malcolm'), (171033, 'dms'), (171086, 'Pratiksha'))

('Evita',)
Select Query executed successfully.

Delete Query executed successfully.

StudentDetails table created successfully.

D:\>
```

```
MariaDB [student]> select * from student;
+-----+-----+
| pid   | name  |
+-----+-----+
| 171034 | Evita |
| 171033 | dms   |
| 171086 | Pratiksha |
+-----+-----+
3 rows in set (0.00 sec)
```

```

MariaDB [student]> show tables;
+-----+
| Tables_in_student |
+-----+
| student            |
| studentdetails     |
+-----+
2 rows in set (0.00 sec)

MariaDB [student]> desc StudentDetails;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| roll  | int(11) | YES  |     | NULL    |       |
| phone | int(11) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.01 sec)

MariaDB [student]> drop table student;
Query OK, 0 rows affected (0.08 sec)

MariaDB [student]> show tables;
+-----+
| Tables_in_student |
+-----+
| studentdetails     |
+-----+
1 row in set (0.00 sec)

MariaDB [student]>

```

2. Python program to implement CRUD Operations on Python using sqlite3.

CODE:

```

import sqlite3
con = sqlite3.connect('student.db')
cur = con.cursor()
cur.execute("create table marks(subject varchar(20), score int)")
print("Table marks created successfully.")
cur.execute("insert into Marks values('Maths', 20)")
print("Insert query executed successfully.")
cur.execute("insert into Marks values('COA', 20)")
print("Insert query executed successfully.")
cur.execute("insert into Marks values('CN', 20)")
print("Insert query executed successfully.")
cur.execute("select * from marks")
rows = cur.fetchall()
print("Fetching all rows of the table.")
print(rows)
cur.execute("UPDATE marks SET score=18 WHERE subject='CN'")
print("Update query executed successfully.")
cur.execute("select * from marks")
rows = cur.fetchall()
print("Fetching all rows of the table.")
print(rows)
cur.execute("select subject from marks where score=18")
print("Select subject where score=18: ",cur.fetchone())
print("Select query executed successfully.")
cur.execute("DELETE FROM marks WHERE score=18")

```

```
print("Delete Query executed successfully.")
cur.execute("select * from marks")
rows = cur.fetchall()
print("Fetching all rows of the table.")
print(rows)
cur.execute("create table details(address varchar(50), pincode int)")
print("Table details created successfully.")
cur.execute("drop table marks")
print("Table marks dropped successfully.")
cur.close()
con.close()
```

OUTPUT:

```
Table marks created successfully.
Insert query executed successfully.
Insert query executed successfully.
Insert query executed successfully.
Fetching all rows of the table.
[('Maths', 20), ('COA', 20), ('CN', 20)]
Update query executed successfully.
Fetching all rows of the table.
[('Maths', 20), ('COA', 20), ('CN', 18)]
Select subject where score=18: ('CN',)
Select query executed successfully.
Delete Query executed successfully.
Fetching all rows of the table.
[('Maths', 20), ('COA', 20)]
Table details created successfully.
Table marks dropped successfully.
>>> |
```

POST EXPERIMENT EXERCISE PROGRAMS:

1. Write a Python program to create a table Employee and calculate department wise total and also grand total of all the employees from the Employees table.

CODE:

```
import sqlite3
con = sqlite3.connect('employees.db')
cur = con.cursor()
cur.execute("create table if not exists employee(name varchar(15), salary int, dept varchar(20))")
print("Table Employee created successfully.")
cur.execute("insert or replace into employee values('Evita', 5000, 'INFT')")
print("Insert query executed successfully.")
cur.execute("insert or replace into employee values('Sanket', 4900, 'CMPN')")
print("Insert query executed successfully.")
cur.execute("insert or replace into employee values('dms', 4500, 'INFT')")
print("Insert query executed successfully.")
cur.execute("insert or replace into employee values('Manas', 4550, 'CMPN')")
print("Insert query executed successfully.")
print()
cur.execute("select salary from employee")
rows = cur.fetchall()
salary=0
for field in rows:
    salary=salary+field[0]
print("Total Salary of all dept.:",salary)
cur.execute("select distinct dept from employee")
rows = cur.fetchall()
for field in rows:
    cur.execute("select salary from employee where dept='"+field[0]+"")
    rows1 = cur.fetchall()
    salary=0
    for field1 in rows1:
        salary=salary+field1[0]
    print("Total Salary of",field[0],"dept.:",salary)
cur.close()
con.close()
```

OUTPUT:

```
Table Employee created successfully.
Insert query executed successfully.
Insert query executed successfully.
Insert query executed successfully.
Insert query executed successfully.

Total Salary of all dept.: 18950
Total Salary of INFT dept.: 9500
Total Salary of CMPN dept.: 9450
>>> |
```


a. Client-Server Chat Application using TCP.

TCP Server

```
import socket
host='127.0.0.1'
port=8000
s=socket.socket()
s.bind((host,port))
print("Server is waiting")
s.listen(1)
c,ad=s.accept()
print("Client is connected")
while True:
    data=c.recv(1024)
    if not data:
        break
    print("From Client ",str(data.decode()))
    data1=input("From Server ")
    c.send(data1.encode())
c.close()
```

TCP Client

```
import socket
host='127.0.0.1'
port=8000
s=socket.socket()
s.connect((host,port))
str1=input("Enter your message: ")
while str1!='exit':
    s.send(str1.encode())
    data=s.recv(1024)
    data1=data.decode()
    print("From Server ",data1)
    str1=input("Enter the data: ")
s.close()
```

Output:

<pre>tcpserver.py Server is waiting Client is connected From Client Hi From Server Hello From Client How are you? From Server Fine</pre>	<pre>tcpclient.py Enter your message: Hi From Server Hello Enter the data: How are you? From Server Fine Enter the data: </pre>
--	--

b. Client-Server Chat Application using UDP.

UDP Server

```
from socket import *
port=12000
serversocket=socket(AF_INET,SOCK_DGRAM)
serversocket.bind(("",port))
print("Server is waiting")
while True:
    msg,ad=serversocket.recvfrom(2048)
    if not msg:
        break
```

```

print("From Client "+str(msg.decode()))
data=input("From Server ")
serversocket.sendto(data.encode(),ad)
serversocket.close()

```

UDP Client

```

from socket import *
host='localhost'
port=12000
clientsocket=socket(AF_INET,SOCK_DGRAM)
msg=input("Enter the message: ")
while msg!='exit':
    clientsocket.sendto(msg.encode(),(host,port))
    data,ad=clientsocket.recvfrom(2048)
    print("From Server ",data)
    msg=input("Enter data: ")
clientsocket.close()

```

Output:

<pre> udpserver.py Server is waiting From Client Hi From Server Hello From Client How are you? From Server Fine </pre>	<pre> udpclient.py Enter the message: Hi From Server b'Hello' Enter data: How are you? From Server b'Fine' Enter data: </pre>
--	---

c. Convert the CASE of the message sent from Client to the Server using TCP.

TCP Server

```

import socket
host='127.0.0.1'
port=6767
s=socket.socket()
s.bind((host,port))
print("Server is waiting")
s.listen()
c,ad=s.accept()
print("Client connected")
while True:
    data=c.recv(1024)
    d1=data.decode()
    data1=d1.upper()
    c.send((data1.encode()))
c.close()

```

TCP Client

```

import socket
host='127.0.0.1'
port=6767
s=socket.socket()
s.connect((host,port))
data=input("Enter your message: ")
s.send(data.encode())
data1=s.recv(1024)
print((data1.decode()))
s.close()

```

Output:

tcpserver.py	tcpclient.py
Server is waiting	Enter your message: hii
Client connected	HII
	>>>

d. Convert the CASE of the message sent from Client to the Server using UDP.

UDP Server

```
from socket import *
serverport=12000
serversoc=socket(AF_INET,SOCK_DGRAM)
serversoc.bind(('',serverport))
print("Server is waiting")
while True:
    message,clientaddr=serversoc.recvfrom(2048)
    modimsg=message.upper()
    serversoc.sendto(modimsg,clientaddr)
serversoc.close()
```

UDP Client

```
from socket import *
servername='localhost'
serverport=12000
clientsoc=socket(AF_INET,SOCK_DGRAM)
message=input("Input lowercase sentence: ")
clientsoc.sendto(message.encode(),(servername,serverport))
modimsg,serveraddr=clientsoc.recvfrom(2048)
print(modimsg)
clientsoc.close()
```

Output:

udpserver.py	udpclient.py
Server is waiting	Input lowercase sentence: hii
	b'HII'
	>>>

POST EXPERIMENT EXERCISE QUESTIONS:

1. Write a Python program to copy a file from client to the server using TCP Sockets.

Server

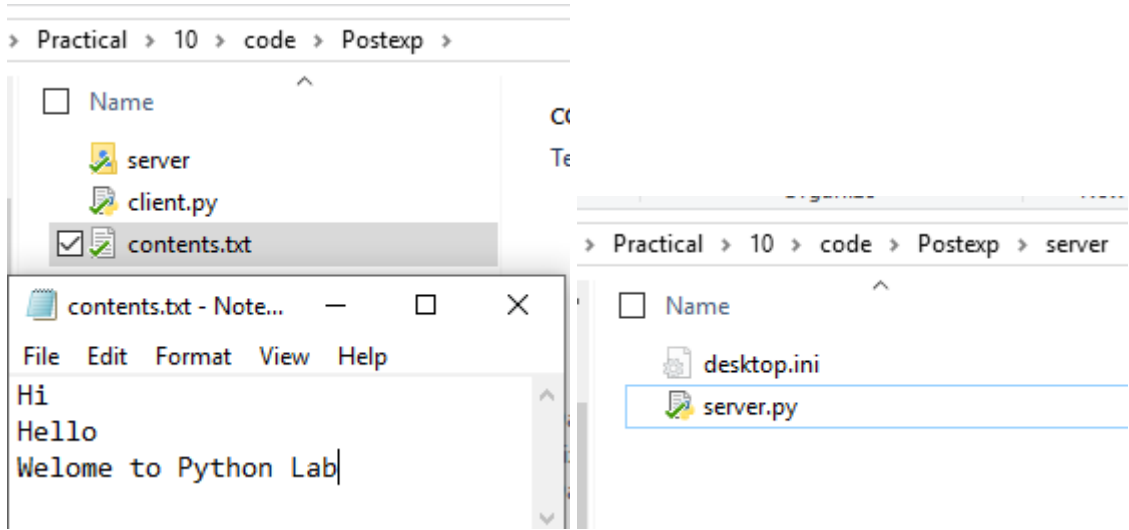
```
import socket          # Import socket module
s = socket.socket()     # Create a socket object
host = socket.gethostname() # Get local machine name
port = 12543           # Reserve a port for your service.
s.bind((host, port))   # Bind to the port
f = open('rec.txt','wb')
s.listen(5)            # Now wait for client connection.
while True:
    c, addr = s.accept() # Establish connection with client.
    print ('Got connection from', addr)
    print ("Receiving...")
    l = c.recv(1024)
    while (l):
        print( "Receiving...")
        f.write(l)
        l = c.recv(1024)
    f.close()
    print ("Done Receiving")
    f = open("rec.txt", "rb")
    print(f.read())
    c.close()           # Close the connection
```

#Client

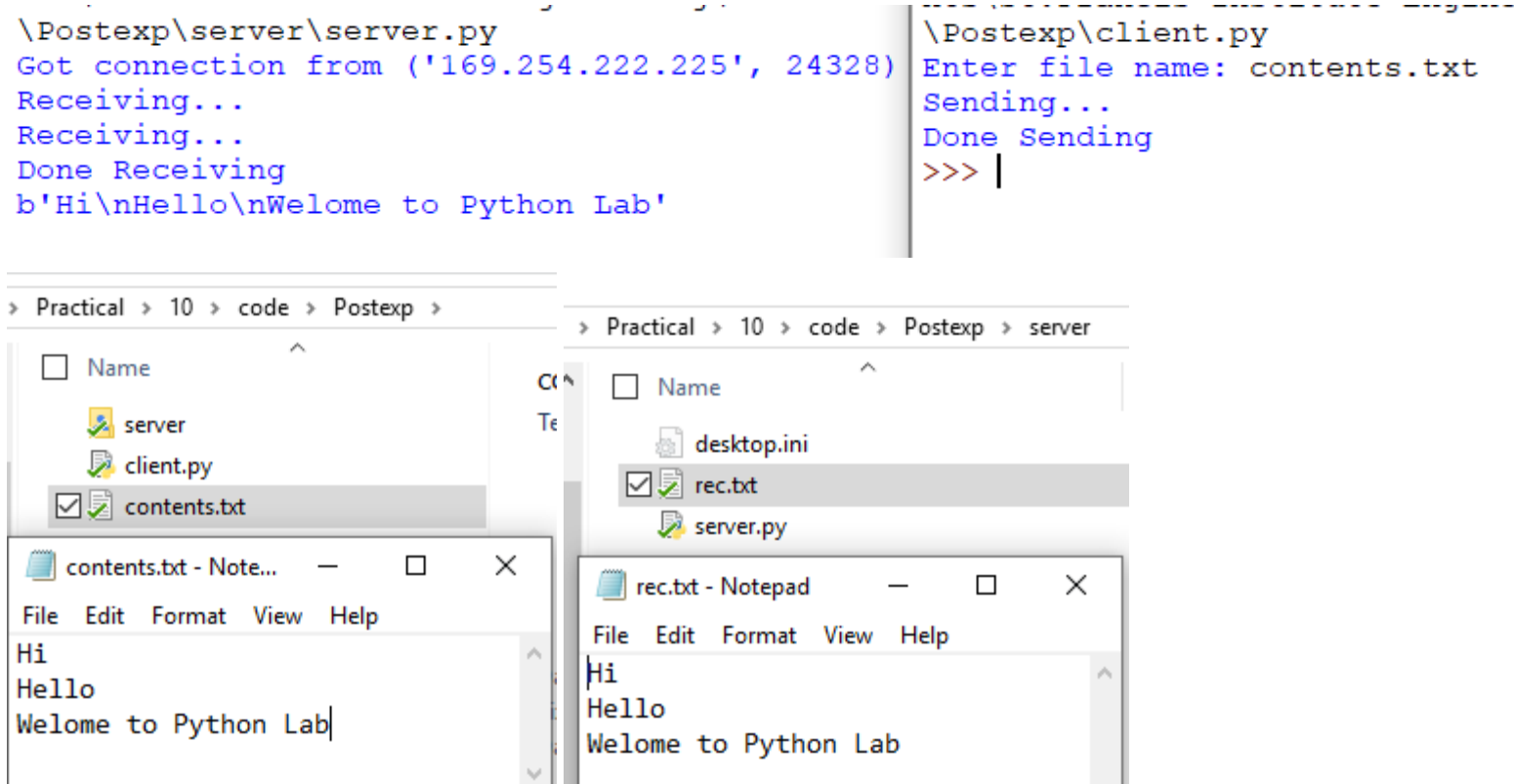
```
import socket          # Import socket module
s = socket.socket()     # Create a socket object
host = socket.gethostname() # Get local machine name
port = 12543           # Reserve a port for your service.
s.connect((host, port))
file=input("Enter file name: ")
f = open(file,'rb')
l = f.read(1024)
while (l):
    print ('Sending...')
    s.sendall(l)
    l = f.read(1024)
f.close()
print ("Done Sending")
s.shutdown(socket.SHUT_WR)          # Close the socket whe
s.close()
```

Output:

Before Execution:



After Execution:



Experiment – 11: Python program to implement Data Visualization using different charts.

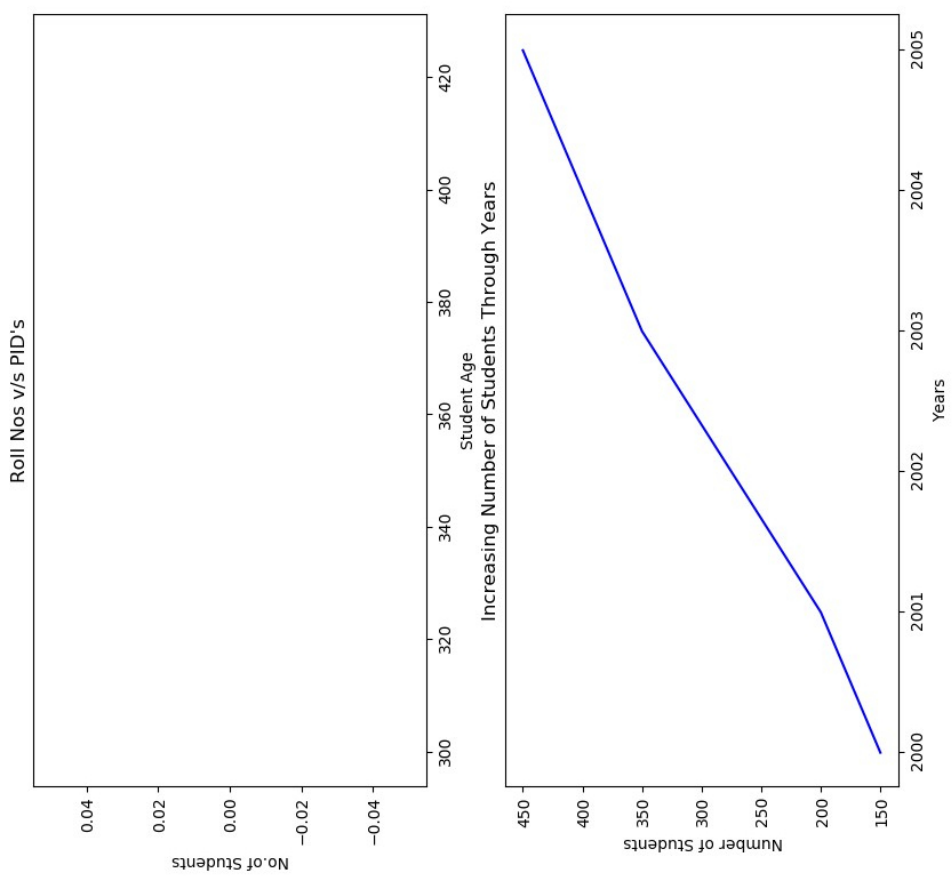
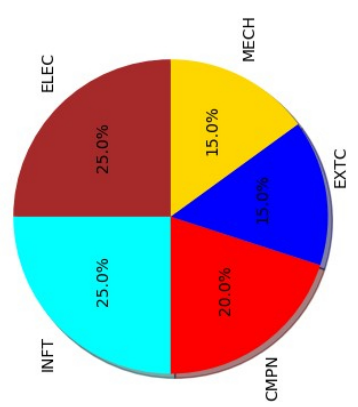
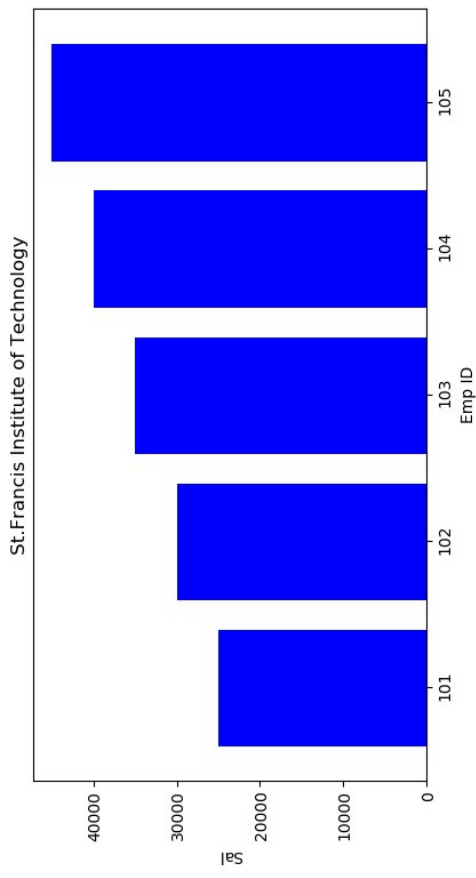
CODE:

```
import matplotlib.pyplot as plt
import matplotlib.pyplot as plt1
import pandas as pd
empdata={"Employee ID":[101,102,103,104,105],"Employee
Name":["A","B","C","D","E"],"Salary":[25000,30000,35000,40000,45000]} #Creating Data Frame using Dictionary
df=pd.DataFrame(empdata)
print("The Data Frame Created Using Dictionary")
print(df)
#Extracting x and y co-ordinates
x=df["Employee ID"]
y=df["Salary"]
plt.subplot(2, 2, 1)
#Plotting Bar Graph
plt.bar(x,y,label="Employee Salaries",color='blue')
plt.xlabel("Emp ID")
plt.ylabel("Sal")
plt.title("St.Francis Institute of Technology")
#List to be plotted for histogram
stud_age=[19,20,21,22,23]
bins=[300,350,400,425]
plt.subplot(2, 2, 2)
#Plot histogram
plt.hist(stud_age,bins,histtype='stepfilled',rwidth=2,color='cyan')
plt.xlabel("Student Age")
plt.ylabel("No.of Students")
plt.title("Roll Nos v/s PID's")
#Lists required for PIE Chart
slices=[25,20,15,15,25]
depts=['INFT','CMPN','EXTC','MECH','ELEC']
color=['cyan','red','blue','gold','brown']
plt.subplot(2,2,3)
plt.pie(slices,labels=depts,colors=color,autopct="%.1f%%",startangle=90,shadow=True) #Plotting Pie Chart
#Rows and Columns of Line Graph
years=[2000,2001,2002,2003,2004,2005]
tot_students=[150,200,275,350,400,450]
plt.subplot(2,2,4)
#Plot Line Graph
plt.plot(years,tot_students,"blue")
plt.title("Increasing Number of Students Through Years")
plt.xlabel("Years")
plt.ylabel("Number of Students")
plt.show() #Displaying the graphs
```

OUTPUT:

```
The Data Frame Created Using Dictionary
   Employee ID Employee Name  Salary
0           101            A   25000
1           102            B   30000
2           103            C   35000
3           104            D   40000
4           105            E   45000
```

Figure 1



POST EXPERIMENT PROGRAMS:

1. Write a Python program to create Data Frame using Dictionary, List of Tuples.

```
>>> import pandas as pd
>>> stud={"pid": [2001,2002,2003,2004,2005,2006], "roll": [1,2,3,4,5,6], "name": ['A', 'B', 'C', 'D', 'E', 'F']}
>>> df=pd.DataFrame(stud)
>>> df
   pid  roll name
0  2001    1    A
1  2002    2    B
2  2003    3    C
3  2004    4    D
4  2005    5    E
5  2006    6    F
>>>

>>> import pandas as pd
>>> student=[(2001,1,'A'),(2002,2,'B'),(2003,3,'C'),(2004,4,'D'),(2005,5,'E'),(2006,6,'F')]
>>> df=pd.DataFrame(student,columns=["pid","roll","name"])
>>> df
   pid  roll name
0  2001    1    A
1  2002    2    B
2  2003    3    C
3  2004    4    D
4  2005    5    E
5  2006    6    F
>>>
```

2. Write a Python program to implement any 5 operations on Data Frame.

```
>>> import pandas as pd
>>> student=[(2001,1,'A'),(2002,2,'B'),(2003,3,'C'),(2004,4,'D'),(2005,5,'E'),(2006,6,'F')]
>>> df=pd.DataFrame(student,columns=["pid","roll","name"])
>>> df
   pid  roll name
0  2001    1    A
1  2002    2    B
2  2003    3    C
3  2004    4    D
4  2005    5    E
5  2006    6    F
>>> df.shape
(6, 3)
>>> df.index
RangeIndex(start=0, stop=6, step=1)
>>> df['roll'].min()
1
>>> df.columns
Index(['pid', 'roll', 'name'], dtype='object')
>>> df[2:4]
   pid  roll name
2  2003    3    C
3  2004    4    D
>>> df[df.roll>4]
   pid  roll name
4  2005    5    E
5  2006    6    F
>>> df.describe()
               pid      roll
count    6.000000  6.000000
mean    2003.500000  3.500000
std       1.870829  1.870829
min     2001.000000  1.000000
25%     2002.250000  2.250000
50%     2003.500000  3.500000
75%     2004.750000  4.750000
max     2006.000000  6.000000
>>> df.pid
0    2001
1    2002
2    2003
3    2004
4    2005
5    2006
Name: pid, dtype: int64
```