Slip 1 :

Q.1 : Write a python program to show how to use else clause with try and except clauses.

```
# Python code to illustrate
# working of try()
def divide(x, y):
      try:
            # Floor Division : Gives only Fractional
            # Part as Answer
            result = x // y
      except ZeroDivisionError:
            print("Sorry ! You are dividing by zero ")
      else:
            print("Yeah ! Your answer is :", result)

# Look at parameters and note the working of Program
divide(3, 2)
divide(3, 0)
```

Q.2 : Write a python program to count and display even and odd numbers of a List.

```
# Python program to count Even
# and Odd numbers in a List

# list of numbers
list1 = [10, 21, 4, 45, 66, 93, 1]

even_count, odd_count = 0, 0

# iterating each number in list
for num in list1:

      # checking condition
      if num % 2 == 0:
            even_count += 1

      else:
            odd_count += 1

print("Even numbers in the list: ", even_count)
print("Odd numbers in the list: ", odd_count)
```

Q.2 : Write a python program to find sum of items of a Dictionary.

```
my_dict = {'data1':100,'data2':-54,'data3':247}
print(sum(my_dict.values()))
```

Slip 2 :

Q.1 : Write a python program to show use of finally clause of exception handling.

```
# Python code to illustrate
# working of try()
def divide(x, y):
      try:
            # Floor Division : Gives only Fractional
            # Part as Answer
            result = x // y
      except ZeroDivisionError:
            print("Sorry ! You are dividing by zero ")
      else:
            print("Yeah ! Your answer is :", result)
      finally:
            # this block is always executed
            # regardless of exception generation.
            print('This is always executed')

# Look at parameters and note the working of Program
divide(3, 2)
divide(3, 0)
```

Q.2 : Write a program that returns a list that contains only the elements that are common between the lists (without duplicates). Make sure your program works on two lists of different sizes.

```
list1=[2,5,6,77,88,5]
list2=[11,23,77,88,4]
for i in list1:
    if i in list2:
        print(i)
```

Q.2 : Write a python program which accepts file name and word to be searched in file. Display line numbers which contain given word and total occurrences of it.

Slip 3 :

Q.1 : Write a python program to show how to raise an exception in python.

```
x = "hello"

if not type(x) is int:
  raise TypeError("Only integers are allowed")
```

Q.2 : Write a python program to display ‘n’ terms of Fibonacci series using recursion.

```
# Python program to display the Fibonacci sequence

def recur_fibo(n):
   if n <= 1:
       return n
   else:
       return(recur_fibo(n-1) + recur_fibo(n-2))

nterms = 10

# check if the number of terms is valid
if nterms <= 0:
   print("Plese enter a positive integer")
else:
   print("Fibonacci sequence:")
   for i in range(nterms):
       print(recur_fibo(i))
```

Q.2 : Write a python program to reverse each word of sentence of a file and also count total
lines.

Slip 4 :

Q.1 : Write a python program to show use of assert keyword.

```
x = "hello"

#if condition returns False, AssertionError is raised:
assert x == "goodbye", "x should be 'hello'"
```

Q.2 : Write a python program to perform following task.

a. Calculate the factorial of given number.

b. Reverse the given number

```
# Python program to find the factorial of a number provided by the user.

# change the value for a different result
num = 7

# To take input from the user
#num = int(input("Enter a number: "))

factorial = 1

# check if the number is negative, positive or zero
if num < 0:
   print("Sorry, factorial does not exist for negative numbers")
elif num == 0:
   print("The factorial of 0 is 1")
else:
   for i in range(1,num + 1):
       factorial = factorial*i
   print("The factorial of",num,"is",factorial)

# reverse a number
num = 1234
reversed_num = 0

while num != 0:
    digit = num % 10
    reversed_num = reversed_num * 10 + digit
    num //= 10

print("Reversed Number: " + str(reversed_num))
```

Q.2 : Write a python program which takes file name as input and print the lines after making
only first character of each word in the sentence capitalized.

```
# Python program to read a file and capitalize
# the first letter of every word in the file.

# A file named "gfg", will be opened with the
# reading mode.
file_gfg = open('gfg.txt', 'r')

# This will traverse through every line one by one
# in the file
for line in file_gfg:

      # This will convert the content
      # of that line with capitalized
      # first letter of every word
      output = line.title()

      # This will print the output
      print(output)
```

Slip 5 :

Q.1 : Write a python program to show how to handle multiple exceptions.

```
string = input()

try:
    num = int(input())
    print(string+num)
except (TypeError, ValueError) as e:
    print(e)
```

Q.2 : Write a python program to display tables from m to n.
Example Input: m=3, n=7

```
Output: 3*1=3    4*1=4 ...... 7*1=7
3*2=6 4*2=8 ...... 7*2=14




3*10=30    4*10=40    ...... 7*10=70
```

Q.2 : Write a python program to accept directory name and print names of all files whose extension is '.txt' in the given directory.

```
# import OS
import os

for x in os.listdir():
     if x.endswith(".txt"):
          # Prints only text file present in My Folder
          print(x)
```

Slip 6 :

Q1) : Write a python program to calculate the cube of all numbers from 1 to n.

```
for i in range(2,5):
    print("the cubes of number",i,"is",i**3)
```

Q2) : Write a python program to display all prime numbers within given range.

Q2)OR

```
import os

count = 0
for root_dir, cur_dir, files in os.walk(r'E:\account'):
    count += len(files)
print('file count:', count)
```

Slip 7 :

Q.1 : Write a python program to find square of given number using list comprehension.

```
def square(a,b):
    square=[value**2 for value in range(a,b+1)]
    return square
print(square(5,6))
```

Q.2 : Write a python program which will find all such numbers which are divisible by 3 and not by 7 within given range m to n.

```
n1=[]
for x in range(10,40):
    if(x%3==0) and (x%7!=0):
        n1.append(str(x))
print(','.join(n1))
```

Slip 8 :

Q1) : Write a python program to find the repeated items of a tuple.

```
var=int(input())
tup=(10, 8, 5, 2, 10, 15, 10, 8, 5, 8, 8, 2)
a=list(tup)
for i in range(len(a)):
  a[i]=int(a[i])
count=a.count(var)
print(var,'appears',count,'times in the tuple')
```

Q.2 : Write a python program with user defined function which accept long string containing multiple words and it return same string with the words in backwards order.
Example: Input= “I am Msc student” then output = “student Msc am I”.

```
string="I am Msc Student"
revstr=string.split()[::-1]
print(revstr)
```

Slip 9 :

Q1) : Write a python program to find the length of a set. (Don’t use built in function len).

```
L={1,2,3,4,5,"abc"}
count=0
for x in L:
    count += 1
print("Length of list is : ",count)
```

Q2) : Write a python program that accepts a sentence and calculate the number of uppercase letters and lowercase letters.

```
upper=0
lower=0
string1=input("enter a string:")
for i in string1:
    if(i.isupper()):
        upper=upper+1
    elif(i.islower()):
        lower=lower+1
    else:
        pass
print("Number of UCL letters:",upper)
print("Number of LCL of letters:",lower)
```

Slip 10 :

Q1) : Write a python program to accept n elements in a set and find the length of a set, maximum, minimum value and the sum of values in a set.

```
a = {23,45,17,8,56,10}
print("Set A :",a)
print("\nMaximum of A :",max(a))
print("Minimum of A :",min(a))
print("Length of  set is : ",len(a))
print("Sum of the values in the set : ",sum(a))
```

Q2) : Write a python program that accepts a sentence and calculate the number of letters and digits in it.

```
alpha=0
digit=0
string1=input("Enter a string:")
for i in string1:
    if(i.isalpha()):
        alpha=alpha+1
    elif(i.isdigit()):
        digit=digit+1
    else:
        pass
print("Number of alphabets="+str(alpha))
print("Number of digits="+str(digit))
```

Slip 11 :

Q.1 : Write a program which checks whether given element exists within a tuple

```
tuplex = ("w", 3, "r", "e", "s", "o", "u", "r", "c", "e")
print("r" in tuplex)
print(5 in tuplex)
```

Q.2 : Write a Python program to find the greatest common divisor (gcd) of two integers.

```
def gcd(a,b):
    if(b==0):
        return a
    else:
        return gcd(b,a%b)
a=60
b=48
print("the gcd of 60 and 48 is : ",end="")
print(gcd(60,48))
```

Q.2 : Define a class Student having members – rollno, name, age, gender. Create a subclass called –Test with member marks of 3 subjects. Create three objects of the Test class and display all the details of the student with percentage.

```
class Student():
    def __init__(self,roll_no,name,age,gender):
        self.roll_no=roll_no
        self.name=name
        self.age=age
        self.gender=gender

class Test(Student):
    def
__init__(self,roll_no,name,age,gender,sub1mark,sub2mark,sub3mark,):
        super().__init__(roll_no,name,age,gender)
        self.mark1=sub1mark
        self.mark2=sub2mark
        self.mark3=sub3mark

    def get_marks(self):
        self.total=self.mark1+self.mark2+self.mark3
        print(self.name , "\b's marks:", self.total)
        print("sub1 marks :",self.mark1)
        print("sub2 marks :",self.mark2)
        print("sub3 marks :",self.mark3)

p1=Test(1,"amar",19,'male',82,89,76)
p2=Test(2,'priya',20,'female',94,91,84)
p1.get_marks()
p2.get_marks()
```

Slip 12 :

Q.1 : Write a python program to find the repeated items of a tuple.

```
var=int(input())
tup=(10, 8, 5, 2, 10, 15, 10, 8, 5, 8, 8, 2)
a=list(tup)
for i in range(len(a)):
  a[i]=int(a[i])
count=a.count(var)
print(var,'appears',count,'times in the tuple')
```

Q.2 : Write a python program to accept string and remove the characters which have odd index values of a given string using user defined function

```
def removeodd(string):
    str2=''
    for x in range(len(string)):
        if x%2==0:
            str2=str2+string[x]
    return str2
str1=input('enter a string : ')
print('String after removing char : ',removeodd(str1))
```

Q.2 : Define a class Person having members — name, address. Create a subclass called
—Employee with member staffed, salary. Create ‘n’ objects of the Employee class and display all the details of the Employee.

```
class person:
    def __init__(self,name,address):
        self.empname=name
        self.address=address

    def display(self):
        print('name : {}\taddress : {}\tsalary : {}'.format(self.empname,
        self.address,a.getsalary()))

class employee(person):
    def __init__(self, name, address,salary):
        super().__init__(name, address)
        self.salary=salary

    def getsalary(self):
        return self.salary
name1=input('enter name : ')
address=input('enter address : ')
salary=int(input('enter salary : '))
a=employee(name1,address,salary)
a.display()
```

Slip 13 :

Q.1 : Write a python program that prints out all the elements of the list that are less than 25.

```
list=[13,6,77,90,5,8]
newlist=[x for x in list if x<25]
print(newlist)
```

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

```
class Circle():
    def __init__(self, r):
        self.radius = r

    def area(self):
        return self.radius**2*3.14

    def perimeter(self):
        return 2*self.radius*3.14

NewCircle = Circle(8)
print(NewCircle.area())
print(NewCircle.perimeter())
```

Q.2 : For given a .txt file that has a list of a bunch of names, count how many of each name there are in the file and print count.

```
counter_dict = {}
with open('nameslist.txt') as f:
      line = f.readline()
      while line:
            line = line.strip()
            if line in counter_dict:
                  counter_dict[line] += 1
            else:
                  counter_dict[line] = 1
            line = f.readline()

print(counter_dict)
```

Slip 14 :

Q.1 : Write a python program which reverse given string and displays both original and reversed string. (Don’t use built-in function)

```
original_string = ['ritika','kirti']
print("List before reverse : ",original_string)
print("List after reverse : ", original_string[::-1])
```

Q.2 : Write a python program to implement binary search to search the given element using
function.

```
def binary_search(arr,a,low,high):
    while low<=high:
        mid=low+(high-low)//2
        if arr[mid]==a:
            return mid
        elif array[mid]<a:
            low=mid+1
        else:
            high=mid-1
        return -1
arr=[1,2,3,4,5,6,7]
a=4
print("The given array is : ",arr)
print("Element to be found is ",a)
index=binary_search(arr,a,0,len(arr)-1)
if index!=-1:
    print("The index of the element is "+str(index))
else:
    print("Element not found...")
```

Q.2 : Write a python program that returns a list that contains only the elements that are common between the lists (without duplicates). Make sure your program works on two lists of different sizes.

```
list1=[2,5,6,77,88,5]
list2=[11,23,77,88,4]
for i in list1:
    if i in list2:
        print(i)
```

Slip 15 :

Q.1 : Write a python program to count the number of characters in a string without using any built-in function.

```
word="python"
length=0
for i in word:
    length=length+1
print(length)
```

Q.2 : Write a python program to check if a given key already exists in a dictionary. If key exists replace with another key/value pair.

```
d={'a':1,'b':2,'c':3}
key=input("Enter key to check : ")
if key in d.keys():
    print("key is present and value of the key is : ")
    print(d[key])
else:
    print("key isn't present!")
new_key="A"
old_key="a"
d[new_key]=d.pop(old_key)
print(d)
```

Slip 16 :

Q.1 : Write a python program to accept and convert string in uppercase or vice versa.

```
c = input()
i = 0
j = []
for i in range(len(c)):
    if c[i]==c[i].lower():
     j.append(c[i].upper())
    else :
        j.append(c[i].lower())
f = ''.join([str(elem) for elem in j])
print(f)
```

Q.2 : Write a python program to create a class Calculator with basic calculator operations (addition,subtraction,division,multiplication,remainder).

```
def add(x, y):
    return x + y

def subtract(x, y):
    return x - y

def multiply(x, y):
    return x * y

def divide(x, y):
    return x / y

print("Select operation.")
print("1.Add")
print("2.Subtract")
print("3.Multiply")
print("4.Divide")

while True:
    choice = input("Enter choice(1/2/3/4): ")
    if choice in ('1', '2', '3', '4'):
        num1 = float(input("Enter first number: "))
        num2 = float(input("Enter second number: "))

        if choice == '1':
            print(num1, "+", num2, "=", add(num1, num2))

        elif choice == '2':
            print(num1, "-", num2, "=", subtract(num1, num2))

        elif choice == '3':
            print(num1, "*", num2, "=", multiply(num1, num2))

        elif choice == '4':
            print(num1, "/", num2, "=", divide(num1, num2))

        next_calculation = input("Let's do next calculation? (yes/no): ")
        if next_calculation == "no":
          break

    else:
        print("Invalid Input")
```