Sr. no	AIM	Marks
1	Write a Python program to demonstrate any 10 methods of List. Methods: Append,Insert,Index,Remove,clear,Extend,copy,reverse,pop,sort,count Code:	10 marks

```
#Creating a List
Practical = ["Vishal","Mahajan"]

#1. Append : Add element to the end of the list
Practical.append("SE-IT-A3")
print("After Appending, List is",Practical)

#2. Insert : Add element at the specified index
Practical.insert(1,"Rajesh")
print("After Inserting, List is",Practical)

#3. Index : Return the index of the first element with the
specified value
indexofVishal= Practical.index("Vishal")
print("Vishal is at",indexofVishal,"index")

#4. Remove : Remove the first item with the specified
value
Practical.remove("SE-IT-A3")
print("After Removing SE-IT-A3, List is",Practical)

#5. Copy : Returns a copy of the list
CopiedPractical = Practical.copy()
print("Copied List is",CopiedPractical)

#6. Extend : Add the elements of a list (or any iterable),
to the end of the current list
ExtendList= ["SE","IT-A","63"]
Practical.extend(ExtendList)
print("After Extending, List is",Practical)
```

	`` #7. Pop : Removes the element at the specified position Practical.pop() print("After Popping, List is",Practical) #8. Reverse : Reverses the order of the list Practical.reverse() print("After Reversing, List is",Practical) #9. Sort : Sorts the list Practical.sort() print("After Sorting, List is",Practical) #10. Count : Returns the number of elements with the specified value CountOfVishal =Practical.count("Vishal") print("Vishal occured",CountOfVishal,"times in the list") #11. Clear : Removes all the elements from the list Practical.clear() print("After Clearing, List is",Practical) ```	
2	Write a Python program to demonstrate working on tuples and any methods of Tuple. <u>Only Two Methods: Count and Index</u> Code: ``` Practical = ("Vishal","Rajesh","Mahajan") print("Original Tuple is",Practical) #Accessing a Element of a Tuple print("0th element of a tuple is",Practical[0]) ```	5 marks

```
#Methods

#1.Count
print("Number of Vishal appeared the Practical tuple
is",Practical.count("Vishal"))

#2.Index
print("Mahajan is at",Practical.index("Mahajan"),"index")
```

3	Write a Python program to demonstrate any 10 methods of Dictionary. Methods: Keys,values,update,clear,copy,pop,popitem,setdefault,items,get Code:	10 marks

```
Practical = {"Name": "Vishal", "Surname" : "Mahajan"}

#1. Keys : Return the keys of the dictionary
print("Keys in Dictionary are",Practical.keys())

#2. Values : Return the values of the dictionary
print("values in Dictionary are",Practical.values())

#3. update : Update the dictionary with the specified
key-value pairs
Practical.update({"Class" : "SE-IT-A3"})
print(Practical)

#4. get : Return the value of the specified key
print("Name is ",Practical.get("Name"))

#5. pop : Remove the element with the specified key
Practical.pop("Class")
print("After Poping Class,Dicitonary is",Practical)

#6. popitem() : Remove the last inserted key-value pair
Practical.popitem()
print("After poping item,Dictionary is",Practical)
```

```
#7. setdefault : Return the value of the specified key. If the key does not exist: insert the key, with the specified value
Practical.setdefault("Surname","Mahajan")
print("After setting default,Dictinoray is",Practical)

#8. copy : Return a copy of the dictionary
CopiedPractical = Practical.copy()
print("Copied Dictionary is",CopiedPractical)

#9. items : Return a list containing a tuple for each key value pair
print(Practical.items())

#10. clear : Remove all elements from the dictionary
print(Practical.clear())
```

4	Write a Python program to demonstrate any 10 methods of Sets. Methods: Add,remove,Union,Intersection,difference,issubset,issuperset,copy,clear,pop	10 marks

```
PracticalSet1 = {"Vishal", "Mahajan", "SE-IT-A3"}
PracticalSet2 = {"Roll no 63","SE-IT-A3"}

#1. add : Add an element to the set
PracticalSet1.add("Roll no 63")
print("After Adding, Set is",PracticalSet1)

#2. remove : Remove the specified element
PracticalSet1.remove("Roll no 63")
print("After Removing, Set is",PracticalSet1)

#3. Union : Return a set containing the union of sets
Union = PracticalSet1.union(PracticalSet2)
print("After Union",Union)
```

	`` #4. intersection : Return a set, that is the intersection of two other sets Intersection = PracticalSet1.intersection(PracticalSet2) print("After Intersection",Intersection) #5. copy : Return a copy of the set CopiedSet = PracticalSet1.copy() print("Copied Set is", CopiedSet) #6. difference : Return a set containing the difference between two or more sets Difference = PracticalSet1.difference(PracticalSet2) print("After Difference",Difference) #7. issubset : Returns whether another set contains this set or not subset = {'Vishal','Mahajan'} print("Is subset?", subset.issubset(PracticalSet1)) #8. superset : Returns whether this set contains another set or not print("Is superset?",PracticalSet1.issuperset(subset)) #9. pop : Remove an element from the set PracticalSet1.pop() print("After poping",PracticalSet1) #10.clear : Remove all elements from the set PracticalSet1.clear() print(PracticalSet1) ```	
5	Write a Python program to implement three different syntaxes of range function ``` for i in range (10): print(i,end= " ") ```	5 Marks

```
for i in range(0,10):
    print(i,end= " ")

for i in range(0,10,2):
    print(i,end= " ")
```

6 | Write a Python program to demonstrate any 10 inbuilt Math methods. | 10 marks

```
import math
#1. Degree to Radian
print("90 degree to Radian is",math.radians(90))

#2. Radian to Degree
print("pie/2 to degree is",math.degrees((math.pi/2)))

#3. sin of a radian
print("Sin of pi/2 is",math.sin((math.pi/2)))

#4.cos of a radian
print("Cos of pi/2 is",math.cos((math.pi/2)))

#5.tan of a radian
print("tan of pi/2 is",math.tan((math.pi/2)))

#6. sin of a radian
print("Asin of pi/2 is",math.asin((1)))

#7.cos of a radian
print("Acos of pi/2 is",math.acos((0)))

#8.tan of a radian
print("Atan of pi/2 is",math.atan((1)))

#9. Pow
print("10 raised to 2 is",math.pow(10,2))

#10.log
print("log of 10 is",math.log(10))
```

	`` #11.log print("log10 of 10 is",math.log10(10)) #12.factorial print("factorial of 5 is",math.factorial(5)) #13.GCD print("GCD of 2 and 4 is",math.gcd(2,4)) ```	
7	Write a Python program to demonstrate any 10 inbuilt String methods ``` string = "vishal MAHAJAN" # Convert all characters in the string to uppercase print("Uppercase string: ", string.upper()) # Convert all characters in the string to lowercase print("Lowercase string: ", string.lower()) # Capitalize the first character of the string print("Capitalized string: ", string.capitalize()) # Split the string into a list of words print("Split string: ", string.split(" ")) # Replace all occurrences of 'v' with 'V' in the string print("Replaced 'v' with 'V': ", string.replace("v","V")) # Check if all characters in the string are uppercase print("Is all uppercase? ", string.isupper()) # Check if all characters in the string are lowercase print("Is all lowercase? ", string.islower()) # Check if the string is a decimal string1 ="10" print("Is decimal? ", string1.isdecimal()) ```	10 marks

```
# Count the number of occurrences of 'A' in the string
print("Count of 'A': ", string.count("A"))

# Find the index of the first occurrence of 'v' in the
string
print("Index of 'v': ", string.index("v"))
```

8 Write a Python program to demonstrate types of operators. 10 marks

```
a=int(input("Enter the First Number on which  Operation is
to be Performed : "))
b=int(input("Enter the Second Number on which Operation is
to be Performed : "))

#1. Arithmetic Operators
print("\nArithmetic Operators")
print("Addition of ",a," and ",b," is ",a+b)
print("Subtraction of ",a," and ",b," is ",a-b)
print("Multiplication of ",a," and ",b," is ",a*b)
print("Division of ",a," and ",b," is ",a/b)
print("Modulus of ",a," and ",b," is ",a%b)
print("Exponent of ",a," and ",b," is ",a**b)
print("Floor Division of ",a," and ",b," is ",a//b)

#2. Comparison Operators
print("\nComparison Operators")
print("Is ",a," greater than ",b," : ",a>b)
print("Is ",a," less than ",b," : ",a<b)
print("Is ",a," equal to ",b," : ",a==b)
print("Is ",a," not equal to ",b," : ",a!=b)
print("Is ",a," greater than or equal to ",b," : ",a>=b)
print("Is ",a," less than or equal to ",b," : ",a<=b)

#3. Bitwise Operators
print("\nBitwise Operators")
print("Bitwise AND of ",a," and ",b," is ",a&b)
```

```
print("Bitwise OR of ",a," and ",b," is ",a|b)
print("Bitwise XOR of ",a," and ",b," is ",a^b)
print("Bitwise NOT of ",a," is ",~a)
print("Bitwise Left Shift of ",a," by 2 is ",a<<2)
print("Bitwise Right Shift of ",a," by 2 is ",a>>2)

#4. Assignment Operators
print("\nAssignment Operators")
c=b
print("The Value of c is ",c)
c+=b
print("The Value of c+=b is ",c)
c-=b
print("The Value of c-=b is ",c)
c*=b
print("The Value of c*=b is ",c)
c/=b
print("The Value of c/=b is ",c)
c%=b
print("The Value of c%=b is ",c)
c**=b
print("The Value of c**=b is ",c)
c//=b
print("The Value of c//=b is ",c)



#5. Logical Operators
print("\nLogical Operators")
print("True and True is ",True and True)
print("True or False is ",True or False)
print("not True is ",not True)
```

9	Write a Python program using if else statement to check if number inputted by user is even or odd `number = int(input("Enter a number: "))`	5 marks

	`` # Check if the number is even if number % 2 == 0: print("The number is even") else: print("The number is odd") ```	
10	Write a Python program using if else statement to demonstrate use of all comparison and logical operators in conjunction with if statement ``` maths=int(input("Enter the marks of Maths: ")) #Using IF-ELIF-ELSE if((maths>=65) and(maths<=75)): print("Student with",maths," marks have Grade B") elif ((maths>=76) and (maths<=85)): print("Student with",maths," marks have Grade A") elif (maths>86): print("Student with",maths," marks have Grade O") else: print("Student with",maths," marks have Grade C") #WAP to determine if the Character enteed is a vowel or not print("\nWAP to determine if the Character enteed is a vowel or not") char=input("Enter any character:") if (char=='A' or char=='E' or char=='I' or char=='O' or char=='U'): print("Entered Char",char,"is vowel") elif (char=='a' or char=='e' or char=='i' or char=='o' or char=='u'): print("Entered Char",char,"is vowel") else: print("Entered Char",char,"is consonant") ```	5 Marks
11	Write a Python program using if elif else statement to demonstrate if number inputted by user is positive, negative or zero	5 Marks

	`` print("\nWAP to test whether a number entered by the user is negative,positive or equal to zero") num=int(input("Enter any number between Positive or Negative: ")) if(num>0): print("Entered Number",num,"is Posititve") elif(num<0): print("Entered Number",num,"is Negative") else: print("Entered Number",num,"is Zero") ```	
12	Write a Python program to read marks of 3 subjects of a student and check if the average marks are above 50 then print that student is passed in exam ``` Sub1=int(input("Enter the marks of Subject 1: ")) Sub2=int(input("Enter the marks of Subject 2: ")) Sub3=int(input("Enter the marks of Subject 3: ")) avg=(Sub1+Sub2+Sub3)/3 if(avg>50): print("Student is Passed with average ",avg) else: print("Student is Failed with average ",avg) ```	5 Marks
13	Write a program to check whether the input year is a leap year or not. ``` year=int(input("Enter the year to be checked: ")) if (year%4==0): if(year%100==0): if(year%400==0): print(year,"is a Leap Year") else: print(year,"is not a Leap Year") else: print(year,"is a Leap Year") ```	5 Marks

14	Write a program to calculate electricity bill according to following criteria i. first 10 units then no charge ii. next 100 units - 5 rs per unit iii. next 200 units - 10 rs per unit Iv. Above this - 15 rs per unit `units=int(input("Enter the units consumed: "))` `if(units<=10):` `  print("No Charge")` `elif(units<=110):` `  print("Bill is ",(units-10)," units * 5 i.e.",(units-10)*5)` `elif(units<=310):` `  print("Bill is (100 units *5) +",(units-110),"units * 10 i.e.", (100*5)+(units-110)*10)` `else:` `  print("Bill is (100 units *5) + (200 units * 10) +",(units-310),"units * 15 i.e.", (100*5)+(200*10)+(units-310)*15)`	5 Marks
15	Write a program to find the lowest number out of the three numbers `num1=int(input("Enter the first number: "))` `num2=int(input("Enter the second number: "))` `num3=int(input("Enter the third number: "))` `if(num1<num2 and num1<num3):` `  print(num1,"is the lowest number")` `elif(num2<num1 and num2<num3):` `  print(num2,"is the lowest number")` `else:` `  print(num3,"is the lowest number")` Or	5 Marks

	`print(min(int(input("Enter the first number: ")), int(input("Enter the second number: ")), int(input("Enter the third number: "))), "is the lowest number")`	
16	Write a program to print all natural numbers between 1 to 100 using looping statements `#Using for Loop` `print("\nUsing for Loop")` `for i in range(1,101):` `  print(i, end=" ")` `#Using while Loop` `print("\n\nUsing while Loop")` `i=1` `while i<=100:` `  print(i, end=" ")` `  i+=1` `print("\n")`	5 Marks
17	Write a program to find the sum of natural numbers up to n, where n is provided by the user. `print("WAP to find the sum of natural numbers up to n,where n is provided by the user.")` `num=int(input("Enter number upto which sum is to be Calculated: "))` `print("\nUsing While Loop")` `if num<0:` `  print("Enter a Positive Integer")` `else:` `  temp=num` `  sum=0` `  while temp >0:` `  sum=sum+temp` `  temp=temp-1`	5 Marks

```
    print("Sum of Number from 1 to",num,"is",sum)

print("\nUsing For Loop")
sum=0
if num<0:
    print("Enter a Positive Integer")
else:
    for i in range(1,num+1):
        sum=sum+i
    print("Sum of Number from 1 to",num,"is",sum)
```

18

Write a python program to read marks of 3 subjects of 10 students and print total marks and average of each student. Also print the message if average is greater than 50 they are "pass"

```
print("\nUsing For Loop")
for i in range(0,10):
    print("Student",i+1)
    total=0

    for j in range(0,3):
        marks=int(input("Enter the marks of Subject
"+str(j+1)+": "))
        total=total+marks
    average=total/3

    print("Total Marks of Student",i+1,"is",total)
    print("Average Marks of Student",i+1,"is",average)
    if average>50:
        print("Student is Pass")
    else:
        print("Student is Fail")

print("\nUsing While Loop")
i=0
while i<10:
    print("Student",i+1)
    total=0
```

5 Marks

```
    j=0
    while j<3:
        marks=int(input("Enter the marks of Subject
"+str(j+1)+": "))
        total=total+marks
        j+=1
    average=total/3

    print("Total Marks of Student",i+1,"is",total)
    print("Average Marks of Student",i+1,"is",average)
    if average>50:
        print("Student is Pass")
    else:
        print("Student is Fail")
    i+=1
```

19

Print Pattern

5 Marks

A				
A	C			
A	C	E		
A	C	E	G	
A	C	E	G	I

```
pattern = ("A","C","E","G","I")

for i in range(6):
    for j in range(i):
        print(pattern[j],end=" ")
    print("\n")
```

<table>
<tr>
<td>20</td>
<td>Print Pattern
<table>
<tr><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr>
<tr><td>4</td><td>3</td><td>2</td><td>1</td><td></td></tr>
<tr><td>3</td><td>2</td><td>1</td><td></td><td></td></tr>
<tr><td>2</td><td>1</td><td></td><td></td><td></td></tr>
<tr><td>1</td><td></td><td></td><td></td><td></td></tr>
</table>

```
for i in range(5,0,-1):
    for j in range(i,0,-1):
        print(j,end=" ")
    print("\n")
```
</td>
<td>5 Marks</td>
</tr>
<tr>
<td>21</td>
<td>Write a Python program to print Fibonacci series of 10 numbers using a while/for loop.

```
a=0
b=1

for i in range(10):
    print(a)
    temp=a
    a=b
    b=b+temp

n=10
a=0
b=1
while(n!=0):
    print(a)
    temp=a
    a=b
    b=b+temp
    n=n-1
```
</td>
<td>5 Marks</td>
</tr>
<tr>
<td>22</td>
<td>Write a program to calculate factorial of a given number using a for loop.</td>
<td>5 Marks</td>
</tr>
</table>

	`` num = int(input("Enter the Number: ")) fact=1 for i in range(num,0,-1): fact=fact*i print(fact) ```	
23	Write a program to print a number in reverse order. Also show if the number is palindrome. ``` num = input("Enter the Number: ") reverse = num[::-1] print("Reverse of a number is",reverse) if num == reverse: print("Entered Number is Palindrome") else: print("Entered Number is not a Palindrome") ```	5 Marks
24	Write a python program to print prime numbers between 1 to 100 ``` for i in range(2,101): for j in range(2,i//2): if i%j == 0: break else: print(i,end=" ") ```	5 Marks
25	Write a Python program to create an array of *double* data type. a. Using this array, create another array whose elements are three times that of the elements of the first array. b. perform slicing and indexing of arrays. c. Demonstrate various methods of array class ``` import array arr = array.array('i', [1, 2, 3, 4, 5]) print("Original Array is",arr) threetimearr = array.array('i',[3*i for i in arr]) ```	10 Marks

```
print("Three Time Array is",threetimearr)

print("First three element of",arr,"is",arr[0:3])
print("First element in the ",arr,"is",arr[0])

arr.append(63)
print("After Adding element 63 , array becomes",arr)

print("Index of element 63
in",arr,"is",arr.index(63))

arr.remove(63)
print("After Removing element 63 , array
becomes",arr)

print("Count of element 63 in array
is",arr.count(63))

arr.extend([7,8,9])
print("Extended Array is",arr)

print("Poping the element from array",arr.pop())

arr.reverse()
print("Reversed array is",arr)
```

26	Write a Python program to implement a stack using an array.	5 Marks

```
import array

class Stack:
    def __init__(self):
        self.stack = array.array("i")

    def push(self,element):
        self.stack.append(element)
        print("After Pushing element,",element,"in
the stack,stack is",self.stack)
```

```
    def pop(self):
        if len(self.stack) == 0:
            print("Stack is Empty")
        else:
            print("After Poping
element,",self.stack.pop(),"from the stack,stack
is",self.stack)
        pass

    def top(self):
        if len(self.stack) == 0:
            print("Stack is Empty")
        else:
            print("Top Element of the Stack
is",self.stack[-1])
        pass

stack = Stack()

while(True):

    choice =
int(input("\n1.Push\n2.Pop\n3.Top\n4.Exit\nEnter Your
Choice:"))

    if choice == 1:
        element = int(input("Enter the Element to be
Pushed: "))
        stack.push(element)
    elif choice == 2:
        stack.pop()
    elif choice == 3:
        stack.top()
    elif choice == 4:
        break
    else:
        print("Enter Right Choice")
```

27	Write a python program to implement Linear Search Algorithm `num = int(input("Enter the Number of Elements: "))` `arr = []` `for i in range(num):` `  arr.append(int(input("Enter the Element: ")))` `element = int(input("Enter the Element to be Searched: "))` `for i in range(len(arr)):` `  if arr[i] == element:` `  print("Element found at",i,"index")` `  break` `else:` `  print("Element Not Found")`	5 Marks
28	Write a python program to implement binary search algorithm `def binarysearch(arr,element):` `  start = 0` `  end = len(arr) - 1` `  for _ in range(len(arr)):` `  mid = start + (end - start) // 2` `  if arr[mid] == element:` `  print("Element Found in the Given array")` `  break` `  elif arr[mid] < element:` `  start = mid + 1` `  else:` `  end = mid - 1` `  else:` `  print("Element Not Found")`	5 Marks

```
num = int(input("Enter the Number of Elements: "))
arr = []
for i in range(num):
    arr.append(int(input("Enter the Element: ")))

element = int(input("Enter the Element to be Searched: "))

arr.sort()
binarysearch(arr,element)
```

29	To implement a python program to define functions to handle multiple exceptions. `print("\nDivided by Zero Error Handling:")` `def divide(num,dem):` `    try:` `        result = num/dem` `    except ZeroDivisionError:` `        print("Division by Zero is not Possible")` `    else:` `        print("Result is",result)` `    finally:` `        print("Execution Completed")` `num = int(input("Enter the Numerator: "))` `dem = int(input("Enter the Denominator: "))` `divide(num,dem)` `print("\nValue Error Handling:")` `def ExceptionHandling():` `    try:` `        num = int(input("Enter the Number: "))` `    except ValueError:`	10-Marks

```
        print("Please Enter a Valid Number")
    else:
        print("Entered Number is",num)
    finally:
        print("Execution Completed")

ExceptionHandling()

print("\nIndex Error Handling:")
def IndexErrorHandling():
    try:
        arr = [1,2,3,4,5]
        index = int(input("Enter the Index: "))
        print("Element at the Given Index is",arr[index])
    except IndexError:
        print("Index Out of Range")
    else:
        print("Element Found")
    finally:
        print("Execution Completed")

IndexErrorHandling()

print("\nKey Error Handling:")
def KeyErrorHandling():
    try:
        dict = {"Name":"Vishal","Surname":"Mahajan"}
        key = input("Enter the Key: ")
        print("Value at the Given Key is",dict[key])
    except KeyError:
        print("Key Not Found")
    else:
        print("Key Found")
    finally:
        print("Execution Completed")

KeyErrorHandling()
```

30	To implement a python program to demonstrate anonymous functions (lambda,map,reduce,filter) `print("\nSquaring using Lambda Function")` `lam = lambda x:x**2` `num = int(input("Enter the Number to be Squared: "))` `print("Square using lambda Function is",lam(num))` `print("\nSquaring a entire List using Map function")` `def squarefunction(num):` `  return num**2` `squarelist = [1,2,3,4,5,6,7,8,9,10]` `result=map(squarefunction,squarelist)` `print("Squared List of",squarelist,"is",list(result))` `print("\nCalculating Addition using Reduce")` `from functools import reduce` `def add(x,y):` `  return x+y` `reducelist = [1,2,3,4,5,6,7,8,9,10]` `print("Sum of elements in` `list",reducelist,"is",reduce(add,reducelist))` `print("\nFiltering Even number in the List using Filter")` `def iseven(num):` `  return num %2 == 0` `filterlist =[1,2,3,4,5,6,7,8,9,10]` `print("Filtered Even Number from the` `list",filterlist,"is",list(filter(iseven,filterlist)))`	10 Marks
31	Write a Python program to demonstrate the use of iterator and generator functions. `Practical = ["Vishal", "Rajesh","Mahajan"]` `print("\nIterating Using Iter")` `iterator = iter(Practical)` `print(next(iterator))` `print(next(iterator))`	10 Marks

```
print(next(iterator))

print("\nIterating thorugh Generator")
def generator(Practical):
    for i in Practical:
        yield i

generator_prac = generator(Practical)
print(next(generator_prac))
print(next(generator_prac))
print(next(generator_prac))
```

32	Write a Python program to calculate sum of first 5 natural numbers/ factorial/ gcd-lcm/ n raise to x using recursion (any one)	5 Marks

```
def sumoffirst5(num):
    if num == 0 :
        return 0
    else :
        return num + sumoffirst5(num-1)

print("\nSum of First 5 natural number is",sumoffirst5(5))


fact_num = int(input("Enter the Number : "))
def factorial(num):
    if num == 0:
        return 1
    else:
        return num * factorial(num-1)

print("Factorial of",fact_num,"is",factorial(fact_num))

num1 = int(input("Enter first number: "))
num2 = int(input("Enter second number: "))

def gcd(num1,num2):
```

```
    if num2== 0:
        return num1
    else:
        return gcd(num1,num1 % num2)

def lcm(num1,num2):
    return (num1*num2)//gcd(num1,num2)

print("Gcd of",num1,num2,"is",gcd(num1,num2))
print("LCM of",num1,num2,"is",lcm(num1,num2))


def nraisedtox(n,x):
    if x == 0:
        return 1
    else:
        return n * nraisedtox(n,x-1)

print(nraisedtox(3,2))
```

33	Declare a Class with class-name Student which accepts the Student details, creates an inner class of Student Marks with a constructor that takes marks as arguments and returns the total and percentage of marks along with the student details	10 Marks

```
class Student:
    def __init__(self,name,rollno):
        self.name = name
        self.rollno = rollno

    class StudentMarks:
        def __init__(self,student,marks,totalmarks):
            self.student = student
            self.marks = marks
            self.totalmarks = totalmarks

        def total(self):
            return sum(self.marks)
```

```
        def percentage(self):
            return (sum(self.marks)/self.totalmarks)*100

        def display(self):
            print("Name of Student is",self.student.name)
            print("Roll no of Student
is",self.student.rollno)
            print("Marks:", self.marks)
            print("Total:", self.total())
            print("Percentage:", self.percentage())


name = input("\nEnter the name of the student:")
rollno = int(input("Enter the rollno of the student:"))

Student_obj = Student(name, rollno)
limit = int(input("\nEnter the number of subjects:"))


marks = []
for i in range(limit):
    marks.append(int(input("Enter the marks:")))

totalmarks = int(input("Enter the Total Marks: "))
StudentMarks_obj =
Student_obj.StudentMarks(Student_obj,marks,totalmarks)
StudentMarks_obj.display()
```

34	Write a Python class named Circle constructed by a radius and two methods which will compute the area and the perimeter of a circle. `import math` `class Circle:` `  def __init__(self,radius):` `  self.radius = radius`	10 Marks

```
    def area(self):
        return math.pi*(self.radius**2)

    def circumference(self):
        return 2*math.pi*self.radius

    def display(self):
        print("\nCircle Details:")
        print("Radius:", self.radius)
        print("Area:", self.area())
        print("Perimeter:", self.circumference())

radius = int(input("Enter the Radius: "))
Circle_obj = Circle(radius)
Circle_obj.display()
```

35

Write a python program to create a Bank class where deposits and withdrawals can be handled by using instance methods.

10 Marks

```
class Bank:
    def __init__(self, balance):
        self.balance = balance

    def deposit(self, amount):
        self.balance += amount
        return self.balance

    def withdraw(self, amount):
        if amount > self.balance:
            return "Insufficient balance"
        else:
            self.balance -= amount
            return self.balance

    def display(self):
        print("\nBank Details:")
        print("Balance:", self.balance)
```

```
balance = int(input("\nEnter the initial balance: "))
Bank_obj = Bank(balance)

amount = int(input("\nEnter the amount to be deposited:
"))
print("Balance after deposit:", Bank_obj.deposit(amount))

amount = int(input("\nEnter the amount to be withdrawn:
"))
print("Balance after withdrawal:",
Bank_obj.withdraw(amount))

Bank_obj.display()
```

36	Write a python program to Create a class 'Employee' with following attribute as 'EmpID', 'name', 'dept', and 'salary'. Print 'name: ', %s and 'salary:', %10.2f when an object is printed. Create a function to update the salary of a given employee. Print the total number of employees. Create two derived classes "Manager" and "Staff" from base class "Employee" and display their details.	10 Marks

```
class Employee:
    total_employees = 0
    def __init__(self, EmpID, name, dept, salary,role):
        self.EmpID = EmpID
        self.name = name
        self.dept = dept
        self.salary = salary
        self.role = role
        Employee.total_employees += 1

    def __str__(self):
        return "Name: %s, Salary: %10.2f" % (self.name,
self.salary)

    def update_salary(self, salary):
        self.salary = salary
```

```
    def display(self):
        print("\nEmployee Details:")
        print("EmpID:", self.EmpID)
        print("Name:", self.name)
        print("Department:", self.dept)
        print("Salary:", self.salary)
        print("Role:", self.role)

    def total(self):
        print("\nTotal number of employees:",
Employee.total_employees)


class Manager(Employee):
    def __init__(self, EmpID, name, dept, salary,role):
        super().__init__(EmpID, name, dept, salary,role)

    def display(self):
        super().display()


class Staff(Employee):
    def __init__(self, EmpID, name, dept, salary,role):
        super().__init__(EmpID, name, dept, salary,role)

    def display(self):
        super().display()




def main():
    employees = []

    while True:
        print("\n1. Add Staff")
        print("2. Add Manager")
        print("3. Display Staff/Manager Details")
```

```
        print("4. Update Salary")
        print("5. Display Total Employees")
        print("6. Exit")

        choice = int(input("Enter your choice: "))

        if choice == 1:
            EmpID = input("Enter Staff ID: ")
            name = input("Enter Staff Name: ")
            dept = input("Enter Department: ")
            salary = float(input("Enter Salary: "))
            employees.append(Staff(EmpID, name, dept,
salary,role="Staff"))

        elif choice == 2:
            EmpID = input("Enter Manager ID: ")
            name = input("Enter Manager Name: ")
            dept = input("Enter Department: ")
            salary = float(input("Enter Salary: "))
            employees.append(Manager(EmpID, name, dept,
salary,role="Manager"))

        elif choice == 3:
            for employee in employees:
                employee.display()

        elif choice == 4:
            EmpID = input("Enter Employee ID to Update
Salary: ")
            salary = float(input("Enter New Salary: "))
            for employee in employees:
                if employee.EmpID == EmpID:
                    employee.update_salary(salary)
                    print("Salary updated successfully.")
                    break
            else:
                print("Employee ID not found.")
```

```
        elif choice == 5:
            print("\nTotal number of employees:",
Employee.total_employees)

        elif choice == 6:
            break

        else:
            print("Invalid choice. Please choose a valid
option.")

if __name__ == "__main__":
    main()
```

37

Write a Python program to declare a base class College having two derived classes student and faculty and display their details.

10 Marks

```
class College:
    def __init__(self, name, dept, role):
        self.name = name
        self.dept = dept
        self.role = role

    def display(self):
        print("\nDetails:")
        print("Name:", self.name)
        print("Department:", self.dept)
        print("Role:", self.role)


class Student(College):
    def __init__(self, name, dept, role, rollno):
        super().__init__(name, dept, role)
        self.rollno = rollno

    def display(self):
        super().display()
        print("Roll No:", self.rollno)
```

```python
class Faculty(College):
    def __init__(self, name, dept, role, empid):
        super().__init__(name, dept, role)
        self.empid = empid

    def display(self):
        super().display()
        print("EmpID:", self.empid)


Studentobj = Student("Vishal", "IT", "Student", 63)
Studentobj.display()

Facultyobj = Faculty("Teacher", "IT", "Faculty", 221068)
Facultyobj.display()
```

38	Write a Python program to declare a class Calculate which calculates the Area of Circle, Triangle and Rectangle (Use Method Overloading).	10 Marks

```python
import math
class Calculate:
    def area(self,**kwargs):
        if "radius"  in kwargs:
            return self.circle(**kwargs)
        elif "length" in kwargs and "breadth" in kwargs:
            return self.rectangle(**kwargs)
        elif "base" in kwargs and "height" in kwargs:
            return self.triangle(**kwargs)

    def circle(self,**kwargs):
        radius = kwargs["radius"]
        return math.pi*(radius**2)

    def rectangle(self, **kwargs):
        length = kwargs["length"]
        breadth = kwargs["breadth"]
        return length * breadth
```

```
    def triangle(self, **kwargs):
        base = kwargs["base"]
        height = kwargs["height"]
        return 0.5 * base * height

Calculate_obj = Calculate()

circle_area = Calculate_obj.area(radius=5)
print("\nArea of circle:", circle_area)

triangle_area = Calculate_obj.area(base=3, height=4)
print("\nArea of triangle:", triangle_area)

rectangle_area = Calculate_obj.area(length=4, breadth=6)
print("\nArea of rectangle:", rectangle_area)
```

39	Write a Python program to Create a user defined module to implement a data structure queue. The module should perform the following functions: **1**. Enqueue element from the rear side **2.** Dequeue element from the front side **3.** Rotate the queue **4.** Extend queue	10 Marks

```
class Queue:
    def __init__(self):
        self.queue = []

    def enqueue(self,element):
        self.queue.append(element)
        print("After Enqueue, Queue is",self.queue)

    def dequeue(self):
        if len(self.queue) != 0:
            element = self.queue.pop(0)
            print("After Dequeue, Queue is",self.queue)
            return element
        else:
            print("Queue is Empty")

    def rotate(self):
```

```
        if len(self.queue) != 0:
            element = self.queue.pop(0)
            self.queue.append(element)
            print("After Rotate, Queue is",self.queue)
        else:
            print("Queue is Empty")

    def extend(self,Listofelements):
        self.queue.extend(Listofelements)
        print("After Extend, Queue is",self.queue)

    def printqueue(self):
        print("Queue is",self.queue)

queue = Queue()

while(True):

print("\nMenu:\n1.Enqueue\n2.Dequeue\n3.Rotate\n4.Extend\n
5.Print\n6.Exit")
    choice = int(input("Enter your choice: "))
    if choice == 1:
        element = input("Enter the element to add into
queue: ")
        queue.enqueue(element)
    elif choice == 2:
        queue.dequeue()
    elif choice == 3:
        queue.rotate()
    elif choice == 4:
        limit = int(input("Enter total number to extend:
"))
        items = []
        print("Enter the elements: ")
        for i in range(limit):
            i = input()
            items.append(i)
        queue.extend(items)
```

```
    elif choice == 5:
        queue.printqueue()
    elif choice == 6:
        break
    else:
        print("Invalid choice")
```

40

Write a Python program to Implement queue using deque (deck) and show insertion of element from the rear side, deletion of element from the front side, rotate and extend queue.

10 Marks

```
import collections
List = list()
de = collections.deque(List)
def Enqueue(de):
     a = int(input("Enter the element to add into queue:
"))
    de.append(a)

def Dequeue(de):
    de.popleft()

def rotate(de):
    de.rotate(-1)

def Extend(de):
    n = int(input("Enter total number to extend: "))
    List1 = []
    print("Enter the elements:")
    for a in range(n):
        a = int(input())
        List1.append(a)
    de.extend(List1)

def Print(de):
    print("\n",de)

while(True):
```

```
                                i=int(input("\nMenu        of
Deque\n1.Enqueue\n2.Dequeue\n3.Rotate\n4.Extend\n5.Print\n
6.Exit\nEnter your choice: "))
    if i == 1:
        Enqueue(de)
    elif i == 2:
        Dequeue(de)
    elif i == 3:
        rotate(de)
    elif i == 4:
        Extend(de)
    elif i == 5:
        Print(de)
    elif i == 6:
        break
```

41	Write a Python program to create, write, read, append and close a file using File manipulating methods.	10 Marks

```
file = open("Vishal.txt", "w")
file.write("Hello, I am Vishal Mahajan SE-IT-A 63")
file.close()


file = open("Vishal.txt", "r")
print(file.read())
file.close()

file = open("Vishal.txt", "a")
file.write("\nI am learning Python")
file.close()

file = open("Vishal.txt", "r")
print(file.read())
file.close()
```

42	Create a class Student to input data members roll number, name, age with a display method to print their details,using pickle module. Show the details for min 5 students. `import pickle` `class Student:` `  def __init__(self):` `  self.name = ""` `  self.rollno = 0` `  self.age = 0` `  def getdata(self):` `  self.name = input("Enter the Name of the Student: ")` `  self.rollno = int(input("Enter the Roll No of the Student: "))` `  self.age = int(input("Enter the Age of the Student: "))` `  def display(self):` `  print("Name: ", self.name)` `  print("Roll number: ", self.rollno)` `  print("Age: ", self.age)` `student = Student()` `student.getdata()` `file = open("VishalPractical.pkl","wb")` `pickle.dump(student,file)` `file = open("VishalPractical.pkl","rb")` `student = pickle.load(file)` `student.display()` `file.close()`	10 Marks

43	Create a GUI with frame and widgets: First name and Last name (Text), Gender (Radiobutton), Subject (dropdown menu). Add 3 buttons the GUI window created for displaying the given input, reset and exit the window. Connect the form to a database and enter values from the form into the database.	10 Marks
44	Write a python program to design canvas and create a shapes using different `import tkinter as tk` `def oval(event):` `  canvas.create_oval(event.x, event.y, event.x+50, event.y+50, fill="green")` `def rectangle(event):` `  canvas.create_rectangle(event.x, event.y, event.x+50, event.y+30, fill="blue")` `root = tk.Tk()` `canvas = tk.Canvas(root)` `canvas.pack()` `canvas.bind("<Button-1>", oval)` `canvas.bind("<Button-3>", rectangle)` `root.mainloop()`	10 Marks
45	Write a Python program to implement Data Frames and different techniques to create Data Frames. Write a Python program to implement any 5 operations on Data Frame.	10 Marks
46	Write a Python program to Visualize the dataframe using Bar chart,Histogram,Pie chart,Line Graph.	10 Marks

47	To implement a python program to demonstrate the following: a. Client-Server Chat Application using TCP. b. Client-Server Chat Application using UDP.	10 Marks
48	Write a python program to To build a REST API using flask. Also mention all the steps to install flask to create web applications. And use of @app.route to display information on web page	10 Marks