Slip 1 A) Write a Python program to accept n numbers in list and remove duplicates from a list.

Answer :

```
def Remove(duplicate):
    final_list = []
    for num in duplicate:
        if num not in final_list:
            final_list.append(num)
    return final_list
# Driver Code
duplicate = [2, 4, 10, 20, 5, 2, 20, 4]
print(Remove(duplicate))
```

Slip 1 B) Write Python GUI program to take accept your birthdate and output your age when a button is pressed.

```
from tkinter import *

from tkinter import messagebox

def clearAll() :

        dayField.delete(0, END)

        monthField.delete(0, END)

        yearField.delete(0, END)

        givenDayField.delete(0, END)

        givenMonthField.delete(0, END)

        givenYearField.delete(0, END)

        rsltDayField.delete(0, END)

        rsltMonthField.delete(0, END)

        rsltYearField.delete(0, END)


def checkError() :

        if (dayField.get() == "" or monthField.get() == ""
```

```
                or yearField.get() == "" or givenDayField.get() == ""
                or givenMonthField.get() == "" or givenYearField.get() == "") :
                messagebox.showerror("Input Error")
                clearAll()
                return -1
def calculateAge() :
        value = checkError()
        if value == -1 :
                return
                else :
                birth_day = int(dayField.get())
                birth_month = int(monthField.get())
                birth_year = int(yearField.get())
                given_day = int(givenDayField.get())
                given_month = int(givenMonthField.get())
                given_year = int(givenYearField.get())
                month =[31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]
                if (birth_day > given_day):
                        given_month = given_month - 1
                        given_day = given_day + month [birth_month-1]
                        if (birth_month > given_month):
                        given_year = given_year - 1
                        given_month = given_month + 12
                calculated_day = given_day - birth_day;
                calculated_month = given_month - birth_month;
```

```
            calculated_year = given_year - birth_year;

            rsltDayField.insert(10, str(calculated_day))

            rsltMonthField.insert(10, str(calculated_month))

            rsltYearField.insert(10, str(calculated_year))

    if __name__ == "__main__" :

    gui = Tk()

    gui.configure(background = "light green")

    gui.title("Age Calculator")

    gui.geometry("525x260")

    dob = Label(gui, text = "Date Of Birth", bg = "blue")

    givenDate = Label(gui, text = "Given Date", bg = "blue")

    day = Label(gui, text = "Day", bg = "light green")

    month = Label(gui, text = "Month", bg = "light green")

    year = Label(gui, text = "Year", bg = "light green")

    givenDay = Label(gui, text = "Given Day", bg = "light green")

    givenMonth = Label(gui, text = "Given Month", bg = "light green")

    givenYear = Label(gui, text = "Given Year", bg = "light green")

    rsltYear = Label(gui, text = "Years", bg = "light green")

    rsltMonth = Label(gui, text = "Months", bg = "light green")

    rsltDay = Label(gui, text = "Days", bg = "light green")

    resultantAge = Button(gui, text = "Resultant Age", fg = "Black", bg = "Red", command =
calculateAge)

    clearAllEntry = Button(gui, text = "Clear All", fg = "Black", bg = "Red", command =
clearAll)

    dayField = Entry(gui)

    monthField = Entry(gui)
```

```
yearField = Entry(gui)

givenDayField = Entry(gui)

givenMonthField = Entry(gui)

givenYearField = Entry(gui)

rsltYearField = Entry(gui)

rsltMonthField = Entry(gui)

rsltDayField = Entry(gui)

dob.grid(row = 0, column = 1)

day.grid(row = 1, column = 0)

dayField.grid(row = 1, column = 1)

month.grid(row = 2, column = 0)

monthField.grid(row = 2, column = 1)

year.grid(row = 3, column = 0)

yearField.grid(row = 3, column = 1)

givenDate.grid(row = 0, column = 4)

givenDay.grid(row = 1, column = 3)

givenDayField.grid(row = 1, column = 4)

givenMonth.grid(row = 2, column = 3)

givenMonthField.grid(row = 2, column = 4)

givenYear.grid(row = 3, column = 3)

givenYearField.grid(row = 3, column = 4)

resultantAge.grid(row = 4, column = 2)

rsltYear.grid(row = 5, column = 2)

rsltYearField.grid(row = 6, column = 2)

rsltMonth.grid(row = 7, column = 2)
```

```
        rsltMonthField.grid(row = 8, column = 2)

        rsltDay.grid(row = 9, column = 2)

        rsltDayField.grid(row = 10, column = 2)

        clearAllEntry.grid(row = 12, column = 2)

        gui.mainloop()
```

Slip 2 A) Write a Python function that accepts a string and calculate the number of upper case letters and lower case letters.

Sample String: 'The quick Brown Fox' Expected

Output: No. of Upper case characters: 3

No. of Lower case characters: 13

Answer:

```
def string_test(s):
   d={"UPPER_CASE":0, "LOWER_CASE":0}
   for c in s:
      if c.isupper():
         d["UPPER_CASE"]+=1
      elif c.islower():
         d["LOWER_CASE"]+=1
      else:
         pass
   print ("Original String : ", s)
   print ("No. of Upper case characters : ", d["UPPER_CASE"])
   print ("No. of Lower case Characters : ", d["LOWER_CASE"])


string_test('The quick Brown Fox')
```

Slip 2 B) Write Python GUI program to create a digital clock with Tkinter to display the time.

Answer :

```
from tkinter import *

from tkinter.ttk import *

from time import strftime

root = Tk()

root.title('Clock')

def time():

        string = strftime('%H:%M:%S %p')

        lbl.config(text = string)

        lbl.after(1000, time)

lbl = Label(root, font = ('calibri', 40, 'bold'),

                        background = 'purple',

                        foreground = 'white')

lbl.pack(anchor = 'centre')

time()

mainloop()
```

Slip 3 A) Write a Python program to check if a given key already exists in a dictionary. If key exists replace with another key/value pair.

Answer:

```
dict = {'Mon':3,'Tue':5,'Wed':6,'Thu':9}
print("The given dictionary : ",dict)
check_key = input("Enter Key to check: ")
check_value = input("Enter Value: ")
if check_key in dict:
   print(check_key,"is Present.")
   dict.pop(check_key)
   dict[check_key]=check_value
else:
   print(check_key, " is not Present.")
   dict[check_key]=check_value
   print("Updated dictionary : ",dict)
```

Slip 3 B) Write a python script to define a class student having members roll no, 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 total marks.

Answer :

```
class Student:
        def GetStudent(self):
                self.RollNo=int(input("\nEnter Student Roll No:"))
                self.Name=input("Enter Student Name:")
                self.Age=int(input("Enter Student Age:"))
                self.Gender=input("Enter Student Gender:")

        def PutStudent(self):
                print("Student Roll No:",self.RollNo)
                print("Student Name:",self.Name)
                print("Student Age:",self.Age)
                print("Student Gender:",self.Gender)

class Test(Student):
        def GetMarks(self):
                self.MarkMar=int(input("Enter Marks of Marathi Subject"))
                self.MarkHin=int(input("Enter Marks of Hindi Subject"))
                self.MarkEng=int(input("Enter Marks of Eglish Subject"))
        def PutMarks(self):
                print("Marathi Marks:", self.MarkMar)
                print("Hindi Marks:", self.MarkHin)
                print("English Marks:", self.MarkEng)
                print("Total Marks:",self.MarkMar+self.MarkHin+self.MarkEng)

n=int(input("Enter How may students"))
lst=[]
```

```
for i in range(0,n):
        obj=input("Enter Object Name:")
        lst.append(obj)
print(lst)

for j in range(0,n):
        lst[j]=Test()
        lst[j].GetStudent()
        lst[j].GetMarks()
        print("\nDisplay Details of Student",j+1)
        lst[j].PutStudent()
        lst[j].PutMarks()
```

Slip 4 A) Write Python GUI program to create background with changing colors
Answer :

```
from tkinter import *

gui = Tk(className='Python  Window Color')

# set window size

gui.geometry("400x200")

#set window color

gui.configure(bg='blue')

gui.mainloop()
```

Slip 4 B) Define a class Employee having members id, name, department, salary. Create a subclass called manager with member bonus. Define methods accept and display in both the classes. Create n objects of the manager class and display the details of the manager having the maximum total salary (salary+bonus).

Answer :

```
class Employee:
    def AcceptEmp(self):
        self.Id=int(input("Enter emp id:"))
        self.Name=input("Enter emp name:")
        self.Dept=input("Enter emp Dept:")
        self.Sal=int(input("Enter emp Salary:"))

    def DisplayEmp(self):
        print("Emp id:",self.Id)
        print("Emp Name:",self.Name)
        print("Emp Dept:",self.Dept)
        print("Emp Salary:",self.Sal)

class Manager(Employee):
    def AcceptMgr(self):
        self.bonus=int(input("Enter Manager Bonus"))
    def DisplayMgr(self):
        print("Manger Bonus is:",self.bonus)
        self.TotalSal=self.Sal+self.bonus
        print("Total Salary: ", self.TotalSal)

n=int(input("Enter How may Managers:"))
lst=[]
for i in range(0,n):
    obj=input("Enter Object Name:")
    lst.append(obj)
```

```
print(lst)

for j in range(0,n):
        lst[j]=Manager()
        lst[j].AcceptEmp()
        lst[j].AcceptMgr()
        print("\nDisplay Details of Manager",j+1)
        lst[j].DisplayEmp()
        lst[j].DisplayMgr()

#maximum logic
maxTotalSal= lst[0].TotalSal
maxIndex=0
for j in range(1,n):
        if lst[j].TotalSal > maxTotalSal:
                maxTotalSal= lst[j].TotalSal
                maxIndex=j
print("\nDisplay Details of Manager Having Maximum Salary(Salary+Bonus)")
lst[maxIndex].DisplayEmp()
lst[maxIndex].DisplayMgr()
```

Slip 5 A) Write a Python script using class, which has two methods get_String and print_String. get_String accept a string from the user and print_String print the string in upper case.

Answer :

```
class MyClass:
        def Get_String(self):
                self.MyStr=input("Enter any String: ")
        def Print_String(self):
                s=self.MyStr
                print("String in Upper Case: " , s.upper())
# main body
Obj=MyClass()
Obj.Get_String()
Obj.Print_String()
```

Slip 5 B) Write a python script to generate Fibonacci terms using generator function.

Answer :

```
def Fibo(terms2):
   f1=0
   yield f1
   f2=1
   yield f2
for i in range(0,terms2-2):
   f3=f1+f2
   yield f3
   f1=f2
   f2=f3
#mainbody
terms1=int(input("How many terms:"))
gen=Fibo(terms1)
while True:
   try:
      print(next(gen))
   except StopIteration:
      break
```

Slip 6 A) Write python script to calculate area and volume of cube and sphere

Answer:

```
pi=22/7

radian = float(input('Radius of sphere: '))

sur_area = 4 * pi * radian **2

volume = (4/3) * (pi * radian ** 3)

print("Surface Area is: ", sur_area)

print("Volume is: ", volume)

# Python3 code to find area

# and total surface area of cube


def areaCube( a ):

        return (a * a * a)

def surfaceCube( a ):

        return (6 * a * a)

a = 5

print("Area =", areaCube(a))

print("Total surface area =", surfaceCube(a))
```

Slip 6 B) Write a Python GUI program to create a label and change the label font style (font name, bold, size). Specify separate check button for each style.

Answer :

```
import tkinter as tk

parent = tk.Tk()

parent.title("Welcome to Tybbaca Student in SMBST College,Sangamner")

my_label = tk.Label(parent, text="Hello", font=("Arial Bold", 70))

my_label.grid(column=0, row=0)

parent.mainloop()
```

Slip 8 A) Write a python script to find the repeated items of a tuple

Answer :

```
#Initialize array
t = (1, 2, 3, 4, 2, 7, 8, 8, 3, 2)
print(t)
lst=[]
print("Repeated elements in given tuple ")
#Searches for repeated element
for i in range(0, len(t)):
    if t.count(t[i])>1 :
        if t[i] not in lst:
            lst.append(t[i])
            print(t[i])
```

Slip 8 B) Write a Python class which has two methods get_String and print_String. get_String accept a string from the user and print_String print the string in upper case. Further modify the program to reverse a string word by word and print it in lower case.

Answer :

```
class MyClass:
        def Get_String(self):
                self.MyStr=input("Enter any String: ")
        def Print_String(self):
                s=self.MyStr
                print("String in Upper Case: " , s.upper())
                #String Reverse logic
                cnt=len(s)
                i=cnt-1
                RevStr=""
                while(i >= 0):
                        RevStr=RevStr + s[i]
                        i=i-1
                print("String in Reverse & Lower case:" , RevStr.lower())
# main body
Obj=MyClass()
Obj.Get_String()
Obj.Print_String()
```

Slip 9 A) Write a Python script using class to reverse a string word by word

Answer :

```
class MyClass:
   def Get_String(self):
      self.myStr=input("Enter any String: ")
   def Reverse_String(self):
      s=self.myStr
      cnt=len(s)
      i=cnt-1
      revStr=""
      while(i >= 0):
         revStr=revStr + s[i]
         i=i-1
         print("String in Reverse:" , revStr)

# main body
Obj=MyClass()
Obj.Get_String()
Obj.Reverse_String()
```

Slip 9 B) Write Python GUI program to accept a number n and check whether it is Prime, Perfect or Armstrong number or not. Specify three radio buttons.

Answer :

```
def palindrome(n):
    temp = n
    rev = 0
    while n > 0:
        dig = n % 10
        rev = rev * 10 + dig
        n = n // 10
    if temp == rev:
        print(temp, "The number is a palindrome!")
    else:
        print(temp, "The number isn't a palindrome!")

def armstrong(n):
    count = 0
    temp = n
    while temp > 0:
        digit = temp % 10
        count += digit ** 3
        temp //= 10
    if n == count:
        print(n, "is an Armstrong number")
    else:
        print(n, "is not an Armstrong number")

def perfect(n):
    count = 0
    for i in range(1, n):
        if n % i == 0:
            count = count + i
    if count == n:
        print(n, "The number is a Perfect number!")
    else:
        print(n, "The number is not a Perfect number!")

if __name__ == '__main__':
    n = int(input("Enter number:"))
    palindrome(n)
    armstrong(n)
    perfect(n)
```

Slip 10 A) Write Python GUI program to display an alert message when a button is pressed.

Answer :

```
from tkinter import *

import tkinter.messagebox

root = tkinter.Tk()

root.title("When you press a button the message will pop up")

root.geometry('500x300')

def onClick():

tkinter.messagebox.showinfo("Welcome to TYBBACA Student", "Hi I'm  Ramesh")

button = Button(root, text="Click Me", command=onClick, height=5, width=10)

button.pack(side='bottom')

root.mainloop()
```

Slip 10 B) Write a Python class to find validity of a string of parentheses, '(', ')', '{', '}', '[' ']'. These brackets must be close in the correct order. for example "()" and "()[]{}" are valid but "[)", "({[)]" and "{{{" are invalid.

Answer :

```
s="{()[}("
lst=[]
if len(s) % 2 != 0:
    print("Invalid Sequence")
else:
    for b in s:
        if b=="(" or b=="{" or b=="[":
            lst.append(b)
        elif b==")" or b=="}" or b=="]":
            cnt=len(lst)-1
            if b==")":
                if lst[cnt]=="(":
                    lst.pop()
                else:
                    print("Invalid Sequence")
                    break
            if b=="}":
                if lst[cnt]=="{":
                    lst.pop()
```

```
            else:
                print("Invalid Sequence")
                break
        if b=="]":
            if lst[cnt]=="[":
                lst.pop()
            else:
                print("Invalid Sequence")
                break
    if len(lst)==0:
        print("valid Sequence")
```

Slip 11 A) Write a Python program to compute element-wise sum of given tuples. Original lists: (1, 2, 3, 4) (3, 5, 2, 1) (2, 2, 3, 1) Element-wise sum of the said tuples: (6, 9, 8, 6)

Answer :

```
x = (1,2,3,4)

y = (3,5,2,1)

z = (2,2,3,1)

print("Original lists:")

print(x)

print(y)

print(z)

print("\nElement-wise sum of the said tuples:")

result = tuple(map(sum, zip(x, y, z)))

print(result)

from numpy import array

lst1=[1,5,7]

lst2=[3,2,1]

a = array(lst1)

b = array(lst2)

print(a + b)
```

Slip 11 B) Write Python GUI program to add menu bar with name of colors as options to change the background color as per selection from menu option.

Answer :

```
from tkinter import *

app = Tk()

app.title("Hi !Welcome to TYBBACA Students")

app.geometry("800x500")

menubar = Menu(app, background='blue', fg='white')

file = Menu(menubar, tearoff=False, background='yellow')
```

```
edit = Menu(menubar, tearoff=False, background='pink')

file.add_command(label="New")

file.add_command(label="Exit", command=app.quit)

edit.add_command(label="Cut")

edit.add_command(label="Copy")

edit.add_command(label="Paste")

menubar.add_cascade(label="File", menu=file)

menubar.add_cascade(label="Edit", menu=edit)

app.config(menu=menubar)

app.mainloop()
```

Slip 12 A) Write a Python GUI program to create a label and change the label font style (font name, bold, size) using tkinter module.

Answer :

```
import tkinter as tk

parent = tk.Tk()

parent.title("Welcome to SMBST College Student")

my_label = tk.Label(parent, text="Hello", font=("Arial Bold", 70))

my_label.grid(column=0, row=0)

parent.mainloop()
```

Slip 12 B) Write a python program to count repeated characters in a string. Sample string: 'thequickbrownfoxjumpsoverthelazydog' Expected output: o-4, e-3, u-2, h-2, r-2, t-2

Answer :

```
check_string="MalegaonBaramatiPune"
dict = {}
for ch in check_string:
    if ch in dict:
        dict[ch] += 1
    else:
        dict[ch] = 1

for key in dict:
    print(key, "-" , dict[key])
```

Slip 13 A) Write a Python program to input a positive integer. Display correct message for correct and incorrect input. (Use Exception Handling)

Answer :

```
num = int (input("Enter Any Positive number:"))
try:
        if num >= 0:
                raise ValueError("Positive Number-Input Number is Correct")
        else:
                raise ValueError("Negative Number-Input Number is InCorrect")
except ValueError as e:
print(e)
```

Slip 13 B) Write a program to implement the concept of queue using list.

Answer :

```
q=[]
def Insert():
   if len(q)==size: # check wether the stack is full or not
      print("Queue is Full!!!!")
   else:
      element=input("Enter the element:")
      q.append(element)
      print(element,"is added to the Queue!")
def Delete():
   if len(q)==0:
      print("Queue is Empty!!!")
   else:
      e=q.pop(0)
      print("element removed!!:",e)
def display():
   print(q)
#Main body
size=int(input("Enter the size of Queue:"))
while True:
   print("\nSelect the Operation: 1.Insert 2.Delete 3.Display 4.Quit")
   choice=int(input())
   if choice==1:
      Insert()
   elif choice==2:
      Delete()
   elif choice==3:
      display()
```

```
elif choice==4:
    break
else:
    print("Invalid Option!!!")
```

Slip 14 A) Write a Python GUI program to accept dimensions of a cylinder and display the surface area and volume of cylinder.

Answer :

```
pi=22/7

height = float(input('Height of cylinder: '))

radian = float(input('Radius of cylinder: '))

volume = pi * radian * radian * height

sur_area = ((2*pi*radian) * height) + ((pi*radian**2)*2)

print("Volume is: ", volume)

print("Surface Area is: ", sur_area)
```

Slip 14 B) Write a Python program to display plain text and cipher text using a Caesar encryption.

Answer :

```
def encrypt(text,s):

result = ""

  for i in range(len(text)):

    char = text[i]

       if (char.isupper()):

      result += chr((ord(char) + s-65) % 26 + 65)

    else:

      result += chr((ord(char) + s - 97) % 26 + 97)

    return result

text = "CEASER CIPHER DEMO"

s = 4

print "Plain Text : " + text

print "Shift pattern : " + str(s)

print "Cipher: " + encrypt(text,s)
```

Slip 15 A) Write a Python class named Student with two attributes student_name, marks. Modify the attribute values of the said class and print the original and modified values of the said attributes.

Answer :

```
class Student:
        def Accept(self):
        self.name=input("Enter Student Name:")
        self.mark=int(input("Enter Student Total Marks:"))

        def Modify(self):
            self.oldmark=self.mark
            self.mark=int(input("Enter Student New Total Marks:"))
            print("Student Name:",self.name)
            print("Old Total Mark:",self.oldmark)
            print("New Total Mark:",self.mark)
#main body
Stud1=Student()
Stud1.Accept()
Stud1.Modify()
```

Slip 15 B) Write a python program to accept string and remove the characters which have odd index values of given string using user defined function.

Answer :

```
def MyStr():
   Str=input("Enter any String: ")
   cnt=len(Str)
   newStr=""
   for i in range(0,cnt):
      if i%2 ==0:
         newStr=newStr + Str[i]
   print("New String with removed odd Index Character: ",newStr)
# Mainbody
MyStr()
```

Slip 16 A) Write a python script to create a class Rectangle with data member's length, width and methods area, perimeter which can compute the area and perimeter of rectangle.

Answer :

```
class Rect:
        def __init__(self,l2,w2):
                self.l=l2
                self.w=w2
        def RectArea(self):
                self.a=self.l * self.w
                print("Area of Rectangle:", self.a)
        def RectPer(self):
                self.p=2*(self.l + self.w)
                print("Perimeter of Rectangle:", self.p)
#main body
l1=int(input("Enter Length:"))
w1=int(input("Enter Width:"))

Obj=Rect(l1,w1)
Obj.RectArea()
Obj.RectPer()
```

Slip 16 B) Write Python GUI program to add items in listbox widget and to print and delete the selected items from listbox on button click. Provide three separate buttons to add, print and delete.

Slip 17 A) Write Python GUI program that takes input string and change letter to upper case when a button is pressed.

Answer:

Slip 17 B) Define a class Date (Day, Month, Year) with functions to accept and display it. Accept date from user. Throw user defined exception "invalid Date Exception" if the date is invalid.

Answer :

```
class MyDate:
    def accept(self):
        self.d=int(input("Enter Day:"))
        self.m=int(input("Enter Month:"))
        self.y=int(input("Enter Year:"))
    def display(self):
        try:
            if self.d>31:
                raise ValueError("Day value is greater than 31")
            if self.m>12:
                raise ValueError("Month Value is Greater than 12")
                print("Date is: ", self.d, "-" ,self.m , "-",self.y )
        except ValueError as e:
        print(e)
#main body
Obj= MyDate()
Obj.accept()
Obj.display()
```

Slip 18 A) Create a list a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89] and write a python program that prints out all the elements of the list that are less than 5.

Answer :

```
lst=[1,1,2,3,5,8,13,21,34,55]
cnt=len(lst)
print("Total number of Element in list is:",cnt)
for i in range(0,cnt):
        if lst[i]<5:
                print(lst[i])
```

Slip 18 B) Write a python script to define the class person having members name, address. Create a subclass called Employee with members staffed salary. Create 'n' objects of the Employee class and display all the details of the employee.

Answer :

Slip 19 A) Write a Python GUI program to accept a number form user and display its multiplication table on button click.

Answer :

```
from tkinter import *

def show_table():

        num = int(entry.get())

        str1=' Table of ' + str(num) + '\n----------------\n'

        for i in range(1,11):

                str1 = str1 + " " + str(num) + " x " + str(i) + " = " + str(num*i) + "\n"


        output_label.configure(text = str1, justify=LEFT)


main_window = Tk()

main_window.title("Multiplication Table")

message_label = Label(text= ' Enter a number to \n display its Table ' ,

font=( ' Verdana ' , 12))

output_label = Label(font=( ' Verdana ' , 12))

entry = Entry(font=( ' Verdana ' , 12), width=6)

calc_button = Button(text= ' Show Multiplication Table ' , font=( ' Verdana ', 12),

command=show_table)

message_label.grid(row=0, column=0,padx=10, pady=10)

entry.grid(row=0, column=1,padx=10, pady=10, ipady=10)

calc_button.grid(row=0, column=2,padx=10, pady=10)

output_label.grid(row=1, column=0, columnspan=3,padx=10, pady=10)

mainloop()
```

Slip 19 B) Define a class named Shape and its subclass(Square/ Circle). The subclass has an init function which takes an argument (Lenght/redious). Both classes should have methods to calculate area and volume of a given shape.

Answer :

```
class Shape:
        pass

class Square(Shape):
        def __init__(self,l2):
                self.l=l2
        def SArea(self):
                a=self.l * self.l
                print("Area of Square:", a)
        def SPerimeter(self):
                p=4 * self.l
                print("Perimeter of Square:",p)
class Circle(Shape):
        def __init__(self,r2):
                self.r=r2
        def CArea(self):
                a=3.14 * self.r * self.r
                print("Area of Circle:", a)
        def SCircumference(self):
                c=2 * 3.14 * self.r
                print("Circumference of Circle:",c)
#main body
l1=int(input("Enter Length of Square: "))
obj=Square(l1)
obj.SArea()
obj.SPerimeter()

r1=int(input("Enter Radius of Circle: "))
obj=Circle(r1)
obj.CArea()
obj.SCircumference()
```

Slip 20 A) Write a python program to create a class Circle and Compute the Area and the circumferences of the circle.

Answer :

```
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())
```

Slip 20 B) Write a Python script to generate and print a dictionary which contains a number (between 1 and n) in the form(x,x*x). Sample Dictionary (n=5) Expected Output: {1:1, 2:4, 3:9, 4:16, 5:25}

Answer :

```
dict={}
n=int(input("How many numbers do you want to add in Dictionar:"))
for x in range(1,n+1):
    dict[x]=x*x
print(dict)
```

Slip 21 A) Define a class named Rectangle which can be constructed by a length and width. The Rectangle class has a method which can compute the area and Perimeter.

Answer :

```
class Rectangle:

    def __init__(self, l, w):

        self.length = l

        self.width  = w

       def rectangle_area(self):

        return self.length*self.width

newRectangle = Rectangle(12, 10)

print(newRectangle.rectangle_area())
```

Slip 21 B) Write a Python program to convert a tuple of string values to a tuple of integer values.

Original tuple values: (('333', '33'), ('1416', '55'))

New tuple values: ((333, 33), (1416, 55))

Answer :

```
def Convert_Fun(tuple_str):
    result = tuple((int(x[0]), int(x[1])) for x in tuple_str)
    return result
tuple_str = (('333', '33'), ('1416', '55'))
print("Original tuple values:")
print(tuple_str)
print("\nNew tuple values:")
print(Convert_Fun(tuple_str))
```

Slip 22 A) Write a python class to accept a string and number n from user and display n repetition of strings by overloading * operator.

Answer :

```
list_of_colors = ['Red', 'White', 'Black']

colors = '-'.join(list_of_colors)  # *

print()

print("All Colors: "+colors)

print()
```

Slip 22 B) Write a python script to implement bubble sort using list

Answer :

```
lst=[12,10,17,9,1]
cnt=len(lst)
for i in range(0,cnt-1):
    for j in range(0,cnt-1):
        if lst[j]>lst[j+1]:
            temp=lst[j]
            lst[j]=lst[j+1]
            lst[j+1]=temp
print(lst)
```

Slip 23 A) Write a Python GUI program to create a label and change the label font style (font name, bold, size) using tkinter module.

Answer :

```
import tkinter as tk

parent = tk.Tk()

parent.title("Welcome to India")

my_label = tk.Label(parent, text="Hello", font=("Arial Bold", 90))

my_label.grid(column=0, row=0)

parent.mainloop()
```

Slip 23 B) Create a class circles having members radius. Use operator overloading to add the radius of two circle objects. Also display the area of circle.

Answer :

Slip 24 A) Write a Python Program to Check if given number is prime or not. Also find factorial of the given no using user defined function.

Answer :

```
def Prime(num):
        flag=0
        for i in range(2,num):
                if num%i==0 :
                        flag=1
                        break
                if flag==0:
                        print("Number is Prime")
                else:
                        print("Number is Not Prime")
        def Fact(num):
                f=1
                for i in range(1,num+1):
                        f=f*i
                print("Factorial of Given number is:",f)
#main body
n=int(input("Enter any number to Check:"))
Prime(n)
Fact(n)
```

Slip 24 B) Write Python GUI program which accepts a number n to displays each digit of number in words.

Answer :

```
def printValue(digit):

    if digit == '0':

        print("Zero ", end = " ")

    elif digit == '1':

        print("One ", end = " ")

    elif digit == '2':

        print("Two ", end = " ")

    elif digit=='3':

        print("Three",end=" ")

    elif digit == '4':
```

```
            print("Four ", end = " ")
        elif digit == '5':
            print("Five ", end = " ")
        elif digit == '6':
            print("Six ", end = " ")
        elif digit == '7':
            print("Seven", end = " ")
        elif digit == '8':
            print("Eight", end = " ")
        elif digit == '9':
            print("Nine ", end = " ")
def printWord(N):
    i = 0
    length = len(N)
    while i < length:
        printValue(N[i])
        i += 1
N = "123"
printWord(N)
```

Slip 25 A) Write a Python function that accepts a string and calculate the number of upper case letters and lower case letters. Sample String : 'The quick Brow Fox' Expected

Output : No. of Upper case characters : 3

No. of Lower case Characters : 12

Answer :

```
def string_test(s):
    d={"UPPER_CASE":0, "LOWER_CASE":0}
    for c in s:
        if c.isupper():
            d["UPPER_CASE"]+=1
        elif c.islower():
            d["LOWER_CASE"]+=1
        else:
            pass
    print ("Original String : ", s)
    print ("No. of Upper case characters : ", d["UPPER_CASE"])
    print ("No. of Lower case Characters : ", d["LOWER_CASE"])
string_test('The quick Brown Fox')
```

Slip 25 B) Write a Python script to Create a Class which Performs Basic Calculator Operations.

Answer :

```
class MathOp:
    def AddOp(self):
        self.a=int(input("Enter first no:"))
        self.b=int(input("Enter Second no:"))
        self.c= self.a + self.b
        print("Addition is:",self.c)

    def SubOp(self):
        self.a=int(input("Enter first no:"))
        self.b=int(input("Enter Second no:"))
        self.c= self.a - self.b
```

```
                print("Sub is:",self.c)
        def MulOp(self):
                self.a=int(input("Enter first no:"))
                self.b=int(input("Enter Second no:"))
                self.c= self.a * self.b
                print("Addition is:",self.c)
                print("Multiplication is:",self.c)
#main body
obj=MathOp()
while True:
        print("\n1. Addtion")
        print("2. Substraction")
        print("3. Multiplication")
        print("4. Exit")

        ch=int(input("Enter choice to perform any opertaion"))
        if ch==1:
                obj.AddOp()
        elif ch==2:
                obj.SubOp()
        elif ch==3:
                obj.MulOp()
        elif ch==4:
                print("\nProgram Stop")
                break
        else:
                print("Wrong Choice")
```

Slip 26 A) Write an anonymous function to find area of square and rectangle

Answer :

```
areaSquare=lambda length : length * length
areaRect=lambda length,width : length * width
l=int(input("Enter Length Value to calcualte area of Square: "))
print("Area of Square:",areaSquare(l))
l=int(input("\n Enter Length Value to calcualte area of Rectangle:"))
w=int(input("Enter Width Value to calcualte area of Rectangle: "))
print("Area of Rectangle:",areaRect(l,w))
```

Slip 26 B) Write Python GUI program which accepts a sentence from the user and alters it when a button is pressed. Every space should be replaced by *, case of all alphabets should be reversed, digits are replaced by?.

Answer :

Slip 27 A) Write a Python program to unzip a list of tuples into individual lists.

Answer :

```
l = [(1,2), (3,4), (8,9)]
print(list(zip(*l)))
```

Slip 27 B) Write Python GUI program to accept a decimal number and convert and display it to binary, octal and hexadecimal number.

Answer :

```
dec = 344

print("The decimal value of", dec, "is:")

print(bin(dec), "in binary.")

print(oct(dec), "in octal.")

print(hex(dec), "in hexadecimal.")
```

Slip 28 A) Write a Python GUI program to create a list of Computer Science Courses using Tkinter module (use Listbox).

Answer :

```
import tkinter as tk

parent = tk.Tk()

parent.geometry("250x200")

label1 = tk.Label(parent,text = "A list of Computer Science Courses ")

listbox = tk.Listbox(parent)

listbox.insert(1,"PHP")

listbox.insert(2, "Python")

listbox.insert(3, "Java")

listbox.insert(4, "C#")

label1.pack()

listbox.pack()

parent.mainloop()
```

Slip 28 B) Write a Python program to accept two lists and merge the two lists into list of tuple.

Answer :

```
def merge(list1, list2):

        merged_list = [(list1[i], list2[i]) for i in range(0, len(list1))]
        return merged_list
list1 = [1, 2, 3]
list2 = ['a', 'b', 'c']

print(merge(list1, list2))
                lst1=[1,2,3,5]
                lst2=["SMBST","Sangamner"]
                t1=tuple(lst1)
                t2=tuple(lst2)
                t3=t1 + t2
                print(t3)
```

Slip 29 A) Write a Python GUI program to calculate volume of Sphere by accepting radius as input.

Answer:

```
pi = 3.1415926535897931

r= 6.0

V= 4.0/3.0*pi* r**3

print('The volume of the sphere is: ',V)
```

Slip 29 B) Write a Python script to sort (ascending and descending) a dictionary by key and value.

Answer :

```
names = {1:'Sun' ,2:'Mon' ,4:'Wed' ,3:'Tue' ,6:'Fri' ,5:'Thur' }
#print a sorted list of the keys
print(sorted(names.keys()))
#print the sorted list with items.
print(sorted(names.items()))
```

Slip 30 A) Write a Python GUI program to accept a string and a character from user and count the occurrences of a character in a string.

Answer :

```
s = "The TYBBACA Student is clever."

print("Original string:")

print(s)

print("Number of occurrence of 'o' in the said string:")

print(s.count("o"))
```

Slip 30 B) Python Program to Create a Class in which One Method Accepts a String from the User and Another method Prints it. Define a class named Country which has a method called print Nationality. Define subclass named state from Country which has a mehtod called printState. Write a method to print state, country and nationality.

Answer :

```
class Country:
    def AcceptCountry(self):
        self.cname=input("Enter Country Name: ")
    def DisplayCountry(self):
        print("Country Name is:", self.cname)

class State(Country):
    def AcceptState(self):
        self.sname=input("Enter State Name: ")
    def DisplayState(self):
        print("State Name is:", self.sname)
#main body
Obj=State()
Obj.AcceptCountry()
Obj.AcceptState()
Obj.DisplayCountry()
Obj.DisplayState()
```