Program 1 : String slicing

Ans :

```
String = input("Enter String :")
print("After applying Slicing on String: \n")
print(String[:3])
print(String[1:5:2])
print(String[-1:-12:-2])
print(String[1::2])
print(String[-1:-6])
print("Upper case the first letter of the string")
cap = String.capitalize()
print(cap,"\n")
print("returns True if the string ends with the specified value, otherwise False.")
end = String.endswith("!")
print(end,"\n")
print("returns True if all the characters are alphabet letters (a-z).")
alpha = String.isalpha()
print(alpha,"\n")
print("returns a string where all characters are lower case.")
low = String.lower()
print(low,"\n")
print("Converts a string into upper case")
upp = String.upper()
print(upp,"\n")
```

Program 2 : Addition of two numbers

Ans :

```
num1=int(input("Enter number 1 :"))
num2=int(input("Enter number 2 :"))
Addition = num1+num2
print("Addition of num1 and num2:-",Addition)

x=10
y=5
if x>y:
   print ("True")
else:
   print("False")
```

```
a=1
b=2
c=1

d=(b**2)-(4*a*c)
if (d<0):
   sol1=(-b+cmath.sqrt(d)/(2*a))
   sol2=(-b-cmath.sqrt(d)/(2*a))
   print("The roots are ",sol1,sol2)
elif (d>0):
   sol1=(-b+math.sqrt(d)/(2*a))
   sol2=(-b-math.sqrt(d)/(2*a))
   print("The roots are {:6.2f} {}".format(sol1,sol2))
elif (d==0):
   sol1=-b/(2*a)
   sol2=sol1
   print("The roots are {} and {}".format(sol1,sol2))
```

Program 3 : Quadratic Equation

Ans :

```
import math
import cmath
```

Program 4 :
Ans :

```
print("Program to write content into file :")
poem = "Hai Good Morning,How are you"
filein = open("C://Users\Shahid\Desktop\python program\Shahid.txt","wt")
filein.write(poem)
filein.close()
print("Program to read content from file :")
fileout = open("C://Users\Shahid\Desktop\python program\Shahid.txt","rt")
poem = fileout.read()
print(poem)
fileout.close()
print("\n")
print("Program to read the content of a file with read chunk")
poe=""
fout=open("C://Users\Shahid\Desktop\python program\Shahid.txt","rt")
chunk=10
while True:
   fragment = fout.read(chunk)
   print(fragment)
   if not fragment:
      break
   poe+=fragment
print(poe)
fout.close()
print("\n")
print("Program to read the content using readlines()")
fiout=open("C://Users\Shahid\Desktop\python program\Shahid.txt","rt")
lines=fiout.readlines()
print(lines)
print(len(lines),"Lines read")
fout.close()
```

Program 5 : Find the largest of five numbers
Ans :

```
num1=int(input("Enter first number:-"))
num2=int(input("Enter second number:-"))
if(num1>num2):
   print(num1,"is greatest")
elif(num2>num1):
   print(num2,"is greatest")
else:
   print("num1 and num2 are equal")
```

Program 6 :
Ans :

```
def largeoftwo():
   a = int(input("Enter First Number :"))
```

```
    b = int(input("Enter Second Number :"))
    if(a > b):
        print("Firstnumber is Greater than Secondnumber")
    elif(b > a):
        print("Seconnumber is Greater than Firstnumber")
    else:
        print("Both First and Second are equal")

def bigofthree():
    num1 = int(input("Enter First Number :"))
    num2 = int(input("Enter Second Number :"))
    num3 = int(input("Enter Third Number :"))

    if(num1 >= num2) and (num1 >= num3):
        lnum = num1
    elif(num2 >= num1) and (num2 >= num3):
        lnum = num2
    else:
        lnum = num3
    print("The largest number among",num1,",",num2,"and",num3,"is: ",lnum)

while(True):
    choice = int(input("1.Largest of two number\n2.Biggest of three number\n3.Exit\nEnter Choice:"))
    if(choice == 1):
        largeoftwo()
    elif(choice == 2):
        bigofthree()
    elif(choice == 4):
        print("Thank you!")
        break
    else:
        print("Enter valid choice!")
```

Program 7 : Perfect number , armstrong number , Palindrom number

Ans :

```
def armstrong():
    num = input("Enter Number:")
    order=len(num)
    num=int(num)
    sum = 0
    temp = num
    while temp > 0:
        digit = temp % 10
        sum += digit ** order
        temp //= 10

    if num == sum:
        print(num,"is an Armstrong number")
    else:
        print(num,"is not an Armstrong number")
```

```
def palindrome():
    num=int(input("Enter a number:"))
    temp=num
    rev=0
    while(num>0):
        dig=num%10
        print("digit=",dig)
        rev=rev*10+dig
        print("rev=",rev)
        num=num//10
        print("num is ",num)
    if(temp==rev):
        print("The number is palindrome!")
    else:
        print("Not a palindrome!")

def perfectno():
    Number = int(input(" Please Enter any Number: "))
    Sum = 0
    for i in range(1, Number):
        print("i=",i)
        if(Number % i == 0):
            Sum = Sum + i
            print("sum is ",Sum)
    if (Sum == Number):
        print(" %d is a Perfect Number" %Number)
    else:
        print(" %d is not a Perfect Number" %Number)

while(True):
    choice = int(input("1.Armstrong number\n2.Palindrom number\n3.Perfect Number\n4.Exit\nEnter Choice:"))
    if(choice == 1):
        armstrong()
    elif(choice == 2):
        palindrome
    elif(choice == 3):
        perfectno()
    elif(choice == 4):
        print("Thank you!")
        break
    else:
        print("Enter valid choice!")
```

Program 8 : Different Operation on list,tuple,dictiontory

Ans :

```
def ListOpe():
    list = []
    print("empty list",list)
    My_List=['A','B','c','d','e']
    print(My_List)
    print(My_List[1])
    print(My_List[0:4])
```

```
    My_List.append(['f'])
    My_List.append(['g'])
    print("After Addition of two elements :",My_List)

    List2 = ['Hello' , 'there!']
    My_List.append(List2)
    print("Combined two different list",My_List)
    if "B" in My_List:
        print("yes, 'B' is in the List")

    print("Display index",My_List.index('c'))

    My_List.pop(4)
    print(My_List)
    print(My_List.reverse())
    print(My_List.count('e'))
    print(My_List.count('A'))


def TupleOpe():
    tup1 = ('Hello','Hiii','How','Hee')
    print(tup1)
    tup2 = ('A','B','C','D','E')
    print(tup2)

    print("Concatenating tow tuple :-",tup1+tup2)

    print("Apply Slicing on tuple")
    print(tup1[1:])
    print(tup2[2:5])

    print("Length of a tuple2")
    print(len(tup2))

    print("returns the count total number of time word is repetaed in list")
    print(tup2.count("C"))

    my_tuple = ("mouse", [8, 4, 6], (1, 2, 3))
    print(my_tuple)

    tuple = ("hello")
    print(type(tuple))

    print(tup1[0])

    print('How' in tup1)

def DicOpe():
    My_Dict = {'name': 'Shahid','Srn':'R19CA866','Course':'MCA','Section':'B'}
    print("Print Directorie :-",My_Dict)

    print("Print Name :-",My_Dict['name'])
```

```
    print("Update MyCourse")
    My_Dict['Course'] = 'MS'


    print("Removing elements from dictionary")

    print(My_Dict.pop('Section'))
    print(My_Dict)

    print(My_Dict.keys())

    print(My_Dict.get("name","Shahid"))

    print(My_Dict.popitem())

    Dict = My_Dict.copy()
    print(Dict)

    My_Dict.update({"srn":"r19ca866"})
    print(My_Dict)

while(True):
    choice = int(input("1. List Operation\n2. Tuple Operation\n3. Dictionary Operation\n4. Exit\nEnter Choice: "))
    if(choice == 1):
        ListOpe()
    elif(choice == 2):
        TupleOpe()
    elif(choice == 3):
        DicOpe()
    elif(choice == 4):
        print("Thank you!")
        break
    else:
        print("Enter Valid Choice!")
```

Program 9 : Factorial program with function and without function

Ans :

```
def Factusingfunction():
    def factorial(n):
        if n == 0:
            return 1
        else:
            return n * factorial(n-1)
    n=int(input("Input a number to compute the factiorial : "))
    print(factorial(n))


def Factwithoutfunction():
    num=int(input("Enter number :"))
    fact = 1
    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):
            fact = fact*i
            print("The factorial of",num,"is",fact)

while(True):
    choice = int(input("1. Factorialofnumber using function\n2. Factorialofnumber without function\n3.Exit\nEnter
Choice:"))
    if (choice == 1):
        Factusingfunction()
    elif (choice == 2):
        Factwithoutfunction()
    elif (choice == 3):
        print("Thank you!")
        break
    else:
        print("Enter Valid Choice!")
```

Program 10 : program on class

Ans :

```
'''
class professional:
    def __init__(self,n,o):
        self.name=n
        self.occupation=o
    def do_work(self):
        print(self.name,"is a/an",self.occupation)
    def speaks(self):
        print(self.name,"says how are you??")

tom=professional("tom cruise","actor")
tom.do_work()
tom.speaks()

steffi=professional("steffi graff","tennis player")
steffi.do_work()
steffi.speaks()

class customer:
    def __init__(self,name,balance=0):
        self.name=name
        self.balance=balance
        print("The initial balance is",self.balance)
    def withdraw(self,amount):
        if (amount>self.balance):
            raise Exception()
        self.balance-=amount
        return self.balance
```

```
    def deposit(self,amount):
        self.balance+=amount
        return self.balance
steve=customer("steve jobs",500000)
print("The balance in the account after withdrawal is ",steve.withdraw(300000))
print("The balance in the account after deposit is",steve.deposit(400000))
'''
class CSStudent:
    count = 0               # Class Variable
    def __init__(self,name,roll):
        self.name = name          # Instance Variable
        self.__roll = roll          # Instance Variable
        CSStudent.count+=1
    @staticmethod
    def status():
        print("The total number of students are ",CSStudent.count)
    def __private_method(self):
        print("this is a private method")
    def public_method(self):
        self.__private_method()

# Objects of CSStudent class
a = CSStudent('Geek', 1)
print("The value of the static variable when one object got created is ",CSStudent.count)
b = CSStudent('Nerd', 2)
print("private attribute is",a._CSStudent__roll)
a.public_method()

print("The value of the static variable referred through object \"a \" is ",a.count)  # prints "cse"
print("The value of the static variable referred through object \"b \" is ",b.count)  # prints "cse"
print(a.name)    # prints "Geek"
print(b.name)    # prints "Nerd"
print(a._CSStudent__roll)    # prints "1"
print(b._CSStudent__roll)    # prints "2"

# Class variables can be accessed using class
# name also
print(CSStudent.count) # prints "cse"
CSStudent.status()
```

Part B

Program 1 : Design the code illustrating click counert , that binds an events with an event handler.

Ans :

```
#Creating a GUI using a class
# Click Counter
# Demonstrates binding an event with an event handler
from tkinter import *
class Application(Frame):   #class Application extending Frame(Application object
                    #that holds all the buttons)
  """ GUI application which counts button clicks. """
  def __init__(self, master):
```

```
    """ Initialize the frame. """
    super(Application, self).__init__(master)
    self.grid()
    self.bttn_clicks = 0 # the number of button clicks
    self.create_widget()

  def create_widget(self):
    """ Create button which displays number of clicks. """
    self.bttn = Button(self)
    self.bttn["text"]= "Total Clicks: 0"
    self.bttn["command"] = self.update_count
    self.bttn.grid()
  def update_count(self):
    """ Increase click count and display new total. """
    self.bttn_clicks += 1
    self.bttn["text"] = "Total Clicks: " + str(self.bttn_clicks)

# main
root = Tk()
root.title("Click Counter")
root.geometry("200x50")
app = Application(root)
root.mainloop()
```

Program 2 : Demonstrate capturing of input from mouse and move the user defined sprite to the location pointed by the mouse.

Ans :

```
# Moving marbles object
# Demonstrates mouse input
from superwires import games
games.init(screen_width = 640, screen_height = 480, fps = 50)
class Pan(games.Sprite):
  """ A pan controlled by the mouse. """
  def update(self):
    """ Move to mouse coordinates. """
    self.x = games.mouse.x
    self.y = games.mouse.y
def main():
  wall_image = games.load_image("C://Users/Shahid/Desktop/butterfly.jpg transparent = False)
  games.screen.background = wall_image
  pan_image = games.load_image("C://Users/Shahid/Desktop/myimage.bmp")
  the_pan = Pan(image = pan_image,
  x = games.mouse.x,
  y = games.mouse.y)
  games.screen.add(the_pan)
  games.mouse.is_visible = True  #The mouse pointer will not be visible.
  games.screen.event_grab = False  #All imput will be focused on the graphics screen.
  games.screen.mainloop()
# kick it off!
main()
```

Program 3 : Demonstrate the bouncing pizza game.

Ans :

```
# Bouncing Pizza
# Demonstrates dealing with screen boundaries
#Chapter 11 • Graphics: The Pizza Panic Game 343
from superwires import games
games.init(screen_width = 640, screen_height = 480, fps = 50)
class Pizza(games.Sprite):
   """ A bouncing pizza. """

   def update(self):
      """ Reverse a velocity component if edge of screen reached. """
      if self.right > games.screen.width or self.left < 0:
         self.dx = -self.dx
      if self.bottom > games.screen.height or self.top < 0:
         self.dy = -self.dy
def main():
 wall_image = games.load_image("C://Users/Shahid/Desktop/download.jpg", transparent = False)
 games.screen.background = wall_image
 pizza_image = games.load_image("F://pizza.bmp")
 the_pizza = Pizza(image = pizza_image,
 x = games.screen.width/2,
 y = games.screen.height/2,
 dx = 1,
 dy = 1)
 games.screen.add(the_pizza)
 games.screen.mainloop()
# kick it off!
main()
```

Program 4 : Animation adds life to the screen, demonstrate creating an animation with a program

Ans :

```
# Explosion
# Demonstrates creating an animation
from superwires import games
games.init(screen_width = 640, screen_height = 480, fps = 50)

nebula_image = games.load_image("F://nebula.jpg", transparent = 0)
games.screen.background = nebula_image
explosion_files = ["F://pizza.bmp", "F://marbles.bmp"]
explosion = games.Animation(images = explosion_files,
 x = games.screen.width/2,
 y = games.screen.height/2,
 n_repeats = 0,
 repeat_interval = 20)
games.screen.add(explosion)
games.screen.mainloop()
```

Program 5 : Adding sound and music to an application is made simple in python.

Ans :

```
# Sound and Music
# Demonstrates playing sound and music files
from superwires import games
games.init(screen_width = 640, screen_height = 480, fps = 50)
# load a sound file
missile_sound = games.load_sound("missile.wav") #load_sound method takes only wav files
# load the music file
games.music.load("theme.mid")
#load method takes any of these kinds of files WAV, MP3, OGG, and MIDI
choice = None
while choice != "0":
 print("""
 Sound and Music
 0 - Quit
 1 - Play missile sound
 2 - Loop missile sound
 3 - Stop missile sound
 4 - Play theme music
 5 - Loop theme music
 6 - Stop theme music
 """
 )
 choice = input("Choice: ")
 print()
 # exit
 if choice == "0":
 print("Good-bye.")
 # play missile sound
 elif choice == "1":
 missile_sound.play()
 print("Playing missile sound.")
 # loop missile sound
 elif choice == "2":
 loop = int(input("Loop how many extra times? (-1 = forever): "))
 missile_sound.play(loop)
 print("Looping missile sound.")
 # stop missile sound
 elif choice == "3":
 missile_sound.stop()
 print("Stopping missile sound.")
 # play theme music
 elif choice == "4":
 games.music.play()
 print("Playing theme music.")
 # loop theme music
 elif choice == "5":
 loop = int(input("Loop how many extra times? (-1 = forever): "))
```

```
 games.music.play(loop)
 print("Looping theme music.")
 # stop theme music
 elif choice == "6":
 games.music.stop()
 print("Stopping theme music.")
 # some unknown choice
 else:
 print("\nSorry, but", choice, "isn't a valid choice.")
input("\n\nPress the enter key to exit.")
```
