Name:- Keshav Santosh Chaudhari	**Div:-B**
Subject:-Python Programming	
RollNo:-21110019	**Class:-TYBCA**

1.Write a Program related to functions and modules.

```
def add(x,y):
    return x+y
def sub(x,y):
    return x-y
def prod(x,y):
    return x*y
def div(x,y):
    return x/y


import calc
a=10
b=34

addition=calc.add(a,b)
print(addition)

subtraction=calc.sub(a,b)
print(subtraction)

production=calc.prod(a,b)
print(production)

division=calc.div(a,b)
print(division)
```

Output:

```
44

-24

340

0.29411764705882354
```

2.Write a program to demonstrate the use of Dictionary and related functions.

```
# creating a dictionary
country_capitals = {
  "Germany": "Berlin",
  "Canada": "Ottawa",
  "England": "London"
}

# printing the dictionary
print(country_capitals)

#Access dictionary items

country_capitals = {
  "Germany": "Berlin",
  "Canada": "Ottawa",
  "England": "London"
}

# access the value of keys
print(country_capitals["Germany"])
print(country_capitals["England"])

#Add elements to a dictionary
country_capitals = {
  "Germany": "Berlin",
  "Canada": "Ottawa",
}

# add an item with "Italy" as key and "Rome" as its value
country_capitals["Italy"] = "Rome"

print(country_capitals)

#remove dictionary items

country_capitals = {
  "Germany": "Berlin",
  "Canada": "Ottawa",
}

# delete item having "Germany" key
del country_capitals["Germany"]

print(country_capitals)
```

Output:

```
# printing the dictionary

{'Germany': 'Berlin', 'Canada': 'Ottawa', 'England': 'London'}

# access the value of keys

Berlin

London

#Add elements to a dictionary

{'Germany': 'Berlin', 'Canada': 'Ottawa', 'Italy': 'Rome'}

# delete item having "Germany" key

{'Canada': 'Ottawa'}
```

3.Write a program to demonstrate the working of classes and objects.

#Class

```
class Company:
    # attributes
    name = "XYZ Bank"
    turnover = 5000
    revenue = 1000
    no_of_employees = 100

    # method
    def productivity(self):
        return Company.revenue / Company.no_of_employees


comp = Company()
print(comp.name)
print(comp.turnover)
print(comp.revenue)
print(comp.no_of_employees)
print(Company().productivity())
```

Output:

```
XYZ Bank

5000
```

```
1000

100

10.0
```

#Object

```
class Student:
  def __init__(self, rollno, name):
    self.rollno = rollno
    self.name = name
  def displayStudent(self):
    print ("rollno : ", self.rollno, ", name: ", self.name)
emp1 = Student(121, "Priya")
emp2 = Student(122, "Sakshi")
emp1.displayStudent()
emp2.displayStudent()
```

Output:

```
rollno :  121 , name:  Priya

rollno :  122 , name:  Sakshi
```

4.Write a program to demonstrate the working of Inheritance.

1. Single Inheritance

```
class Vehicle:
    def Vehicle_info(self):
        print("Inside Vehicle class")
class Car(Vehicle):
        def car_info(self):
            print("Inside Car class")
car=Car()#Create object of car
car.Vehicle_info()
car.car_info()
```

Output:

```
Inside Vehicle class

Inside Car class
```

2. Multiple Inheritance

```
#Parent class 1
class Person:
   def person_info(self,name,age):
      print("Inside Person Class")
      print('Name:',name,'Age:',age)

#Parent class2
class Company:
   def company_info(self,company_name,location):
      print("Inside Company Class")
      print('Name:',company_name,'location:',location)

#Child class
class Employee(Person, Company):
   def Employee_info(self,salary,skill):
      print("Inside Employee Class")
      print('Salary:',salary,'Skill:',skill)
emp=Employee()
emp.person_info("Keshav",22)
emp.company_info("Google","Pune")
emp.Employee_info(60000,'Meachine Learning')
```

Output:

```
Inside Person Class

Name: Keshav Age: 22

Inside Company Class

Name: Google location: Pune

Inside Employee Class

Salary: 60000 Skill: Meachine Learning
```

3. Multilevel Inheritance

```
class Vehicle:
   def Vehicle_info(self):
      print("Inside Vehicle Class")
```

```
#Child class
class Car(Vehicle):
    def car_info(self):
        print("Inside Car class")

#Child class
class SportsCar(Car):
    def sports_car_info(self):
        print("Inside SportsCar class")

#Create object of SportsCar
s_car=SportsCar()
s_car.Vehicle_info()
s_car.car_info()
s_car.sports_car_info()
```

Output:

```
Inside Vehicle Class

Inside Car class

Inside SportsCar class
```

4. Hierarchical Inheritance

```
class Vehcile:
    def info(self):
        print("This is vehicle")

class Car(Vehcile):
    def car_info(self,name):
        print("Car name is:",name)

class Truck(Vehcile):
    def truck_info(self,name):
        print("Truck name is:",name)

obj1=Car()
obj1.info()
obj1.car_info('BMW')

obj2=Truck()
```

```
obj2.info()
obj2.truck_info('Ford')
```

Output:

```
This is vehicle

Car name is: BMW

This is vehicle

Truck name is: Ford
```

5. Hybrid Inheritance

```
class Vehicle:
    def vehicle_info(self):
        print("Inside Vehicle Class")

class Car(Vehicle):
    def car_info(self):
        print("Inside Car class")

class Truck(Vehicle):
    def truck_info(self):
        print("Inside Truck Class")

class SportsCar(Car,Vehicle):
    def sports_car_info(self):
        print("Inside SportsCar class")

s_car=SportsCar()
s_car.vehicle_info()
s_car.car_info()
s_car.sports_car_info()
```

Output:

```
Inside Vehicle Class

Inside Car class

Inside SportsCar class
```

5.Write a program to demonstrate the working of Overloading Methods and Operator.

1.Method Overloading

```
class Mathematics:
 def add(self,*args):
   sum = 0
   for a in args:
     sum = sum + a
     print(sum)
obj = Mathematics()
obj.add(8, 9, 12)
obj.add(8, 9)
```

Output:

```
8

17

29

8

17
```

2.Operator

```
class A:
    def __init__(self,a):
        self.a=a

    def __add__(self, o):
        return self.a+o.a
ob1=A(1)
ob2=A(2)
ob3=A("Hello ")
ob4=A("Priyanka")

print(ob1+ob2)
print(ob3+ob4)
```

Output:

```
3

Hello Priyanka
```

6.Write a program to demonstrate Exception Handing mechanism.

```
a=10
b=0
try:
    print(a/b)
except Exception:
    print("number can not be divide by zero  ")
    print("Bye")
```

Output:

```
number can not be divide by zero

Bye
```

7. Write a program to demonstrate Regular expression in python.

```
import re

txt="The rain is Spain"

x=re.search("^The.*Spain$",txt)

if x:

    print("YES!We have a match!")

else:

    print("No match")
```

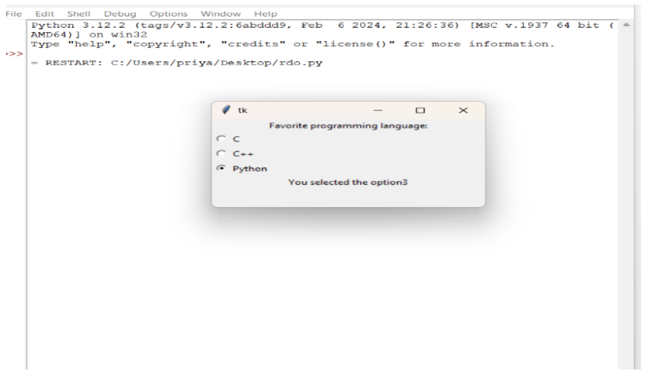
Output:

```
YES! We have a match!
```

8.Write a program to demonstrate Radio button, checkbox, Dialog Boxes using python Tkinter.

1.Radio button

```
from tkinter import*
def selection():
    selection="You selected the option"+str(radio.get())
    label.config(text=selection)
top=Tk()
top.geometry("300x150")
radio=IntVar()
lbl=Label(text="Favorite programming language:")
lbl.pack()
R1=Radiobutton(top,text="C",variable=radio,value=1,command=selection)
R1.pack(anchor=W)
R2=Radiobutton(top,text="C++",variable=radio,value=2,command=selection)
R2.pack(anchor=W)
R3=Radiobutton(top,text="Python",variable=radio,value=3,command=selection)
R3.pack(anchor=W)

label=Label(top)
label.pack()
top.mainloop()
```



```
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb  6 2024, 21:26:36) [MSC v.1937 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/priya/Desktop/rdo.py
```

Favorite programming language:

C

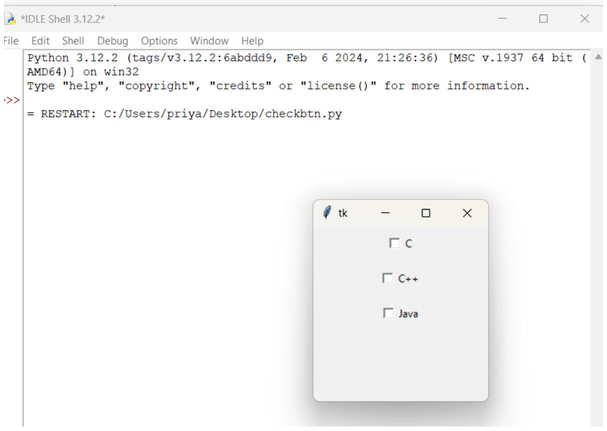
C++

[✓ Python]

You selected the option3

2.Checkbox

```
from tkinter import*
top=Tk()
top.geometry("200x200")
checkvar1=IntVar()
checkvar2=IntVar()
checkvar3=IntVar()
chkbtn1=Checkbutton(top,text="C",variable=checkvar1,onvalue=1,offvalue=0,
                height=2,width=10)
chkbtn2=Checkbutton(top,text="C++",variable=checkvar2,onvalue=1,offvalue=0,
                height=2,width=10)
chkbtn3=Checkbutton(top,text="Java",variable=checkvar3,onvalue=1,offvalue=0,
                height=2,width=10)
chkbtn1.pack()
chkbtn2.pack()
chkbtn3.pack()
top.mainloop()
```

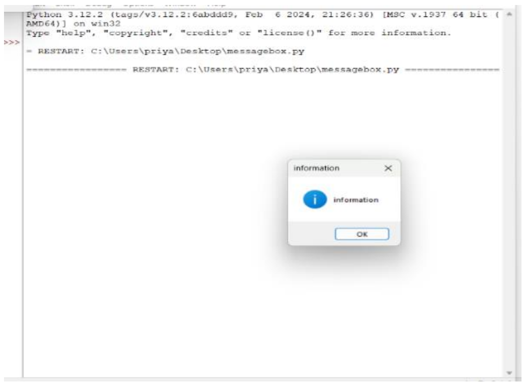


```
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb  6 2024, 21:26:36) [MSC v.1937 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/priya/Desktop/checkbtn.py
```

3.Dialog Boxes

Message box:

```
from tkinter import*
from tkinter import messagebox
top = Tk()
top.geometry("100x100")
messagebox.showinfo("information","information")
top.mainloop()
```

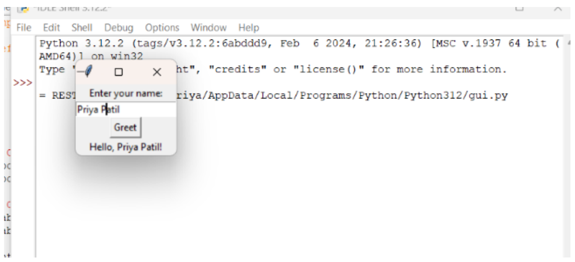


```
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb  6 2024, 21:26:36) [MSC v.1937 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\priya\Desktop\messagebox.py

================= RESTART: C:\Users\priya\Desktop\messagebox.py ================
```

9.Write a program to demonstrate to learn GUI programming using Tkinter.

```
import tkinter as tk
def greet():
    name = entry.get()
    if name:
        greeting.config(text=f"Hello, {name}!")
    else:
        greeting.config(text="Hello!")

# Create the main window
root = tk.Tk()
root.title("Greetings")

# Create and add widgets
label = tk.Label(root, text="Enter your name:")
label.pack()

entry = tk.Entry(root)
entry.pack()

button = tk.Button(root, text="Greet", command=greet)
button.pack()

greeting = tk.Label(root, text="")
greeting.pack()

# Run the application
```

```
root.mainloop()
```



```
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb  6 2024, 21:26:36) [MSC v.1937 64 bit (
AMD64)] on win32
Type "             ht", "credits" or "license()" for more information.
>>>
= REST             riya/AppData/Local/Programs/Python/Python312/gui.py
```

```
Priya Patil
Hello, Priya Patil!
```

10) Program to create a database for insert , update, and delete in SQL.

```
import mysql.connector


try:
   mydb = mysql.connector.connect(
   host="localhost",

   user="root",

   password="Priya@2715",

  auth_plugin = "mysql_native_password"

    )
  mycursor = mydb.cursor()
  mycursor.execute("CREATE DATABASE PRIYA")

  mycursor.execute("USE PRIYA")

    mycursor.execute("CREATE TABLE Employee (name VARCHAR(255),
 profession VARCHAR(255))")
 sql = ("INSERT INTO Employee (name, profession) VALUES (%s,%s)")

 val = ("Priya Patil","Web Developer")

mycursor.execute(sql, val)

 update = "UPDATE Employee SET name = 'Priyanka Patil ' WHERE name
= Priya Patil'"

 mydb.cManishamit()

 delete = "DELETE FRMANISHA Employee WHERE name = Priyanka
Patil'"

mydb.cManishamit() except Exception

as e:     print("An error occurred:", e)
```