Roll No: - _____

Sonopant Dandekar Shikshan Mandali's

Sonopant Dandekar Arts, V.S.Apte Commerce,

M.H.Mehta Science College

DEPARTMENT OF COMPUTER SCIENCE

CERTIFICATE

Certified That Mr./Miss._____

Of_____ has satisfactorily completed a course of

Necessary experiment in _____under

My supervision in the FY.BSC Computer Science in the Year 2024 – 2025

Head of Department **Subject Teacher**

Date: / /2025

INDEX

SR NO.	Title of Exercise	Practical Date	Submission Date	Remarks
1.	Write a python program to implement concept of oop's that is object oriented programming such as creating class and methods.			
2.	Write a python program to implement oop concept of inheritance and polymorphism.			
3.	Write a python program to implement python tkinter elements			
4.	Write a python program to implement python widget classes.			
5.	Aim: Write a python program to implement exception at handling.			
6.	Write a python program to implementing networking.			
7.	Write a python program to implement database in python (creating database, creating tables)			
8.	Write a python program to implement database in python (updating tables, and delete tables)			
9.	Write python program to implement datascience tool (Matplotlib)			
10.	Write a python program to Implement data science tools (scipy and pandas).			

Date: _____

Practical No: 1

Aim: Write a python program to implement concept of oop's that is object oriented programming such as creating class and methods.

1] # CREATE A CLASS.

INPUT:

```
class Myclass:
    x="Hello"
p1=Myclass()
print(p1.x)
```

OUTPUT:

```
Hello
```

2]
```
class person:
    def __init__(self,name,age):
        self.name=name
        self.age=age
p1=person("ABC",36)
print(p1.name)
print(p1.age)
```

OUTPUT:

```
ABC
36
```

3]
```
class employee:
    def __init__(self,name,id,dept,salary):
        self.name=name
        self.id=id
        self.dept=dept
        self.salary=salary
```

```
p1=employee("ABC",36,"CS",5000)
print(p1.name)
print(p1.id)
print(p1.dept)
print(p1.salary)
```

OUTPUT:

```
--------------
ABC
36
CS
5000
```

4]
```
class person:
    def __init__(self,name,age):
        self.name=name
        self.age=age
    def myfunc(self):
        print("my name is:"+self.name)
p1=person("ABC",36)
p1.myfunc()
```

OUTPUT:

```
my name is:ABC
```

5]
```
class employee:
    def __init__(self,name,id,dept,salary):
        self.name=name
        self.id=id
        self.dept=dept
        self.salary=salary
    def myfunc(self):
        print("my name ,id ,dept, and salary are:",self.name,self.id,self.dept,self.salary)
```

p1=employee("ABC",36,"CS",5000)

p1.myfunc()

OUTPUT:

```
---------- RESTART: D:/04018/pract
my name ,id ,dept, and salary are: ABC 36 CS 5000
```

6] class parrot:

name=""

age=0

parrot1=parrot()

parrot1.name="ABC"

parrot1.age=10

parrot2=parrot()

parrot2.name="XYZ"

parrot2.age=15

print(f"{parrot1.name} is {parrot1.age} years old")

print(f"{parrot2.name} is {parrot2.age} years old")

OUTPUT:

```
------------------------------
ABC is 10 years old
XYZ is 15 years old
```

Date: _____

Practical No: 2

Aim:-write a python programme implement with oop concept of the inheritance and polymorphism.

Inheritance:-

1) Single inheritance

A]

INPUT:

```
class animal:
    def eat (self):
        print("I can eat !")

    def sleep(self):
        print("I can sleep")

class dog(animal):
        def bark(self):
            print("I can bark!Woof!Woof!")

dog1=dog()
dog1.eat()
dog1.sleep()
dog1.bark()
```

OUTPUT:

```
========== RESTART: C:/Users/student/Desktop/64027/
I can eat !
I can sleep
I can bark!Woof!Woof!
>
```

B]

INPUT:

```
#employee
class emp:
    def age (self):
        print("I am 40 years")

    def name(self):
        print("I am john")

class company(emp):
        def sarlay(self):
            print(245000)

emp1=company()
emp1.name()
emp1.sarlay()
emp1.age()
```

OUTPUT:

```
========== RESTART: C:/Users/student/Desktop/64027
I am john
245000
I am 40 years
>
```

2] Multiple Inherantce

A]

INPUT:

```
class A:
    def m1(self):
        print("parent classA: m1 method ")


class B(A):
   def m2(self):
       print("parent class B :m2 method")


class C(B):
    def m3(self):
        print("parent classC: m3 method ")


obj=C()
obj.m1()
obj.m2()
obj.m3()
```

OUTPUT:

```
========== RESTART: C:/Users/student/Desktop/64027/py
parent classA: m1 method
parent class B :m2 method
parent classC: m3 method
```

B]

INPUT:

```
class student:
    def name(self):
        print("john ")
```

```
class year(student):
   def age(self):
       print(20)


class user(year):
    def id(self):
        print(64027)


std=user()
std.name()
std.age()
std.id()
```

OUTPUT:

```
================== RESTART: C:/Users/student/Desktop/64027/
john
20
64027
```

```
#different
class A:
    def m1(self):
        print("parent classA: m1 method ")


class B(A):
   def m2(self):
       print("parent class B :m2 method")
```

```
class C(B):
    def m3(self):
        print("parent classC: m3 method ")

obj=B()
obj1=C()
obj.m1()
obj.m2()

obj1.m1()
obj1.m2()
obj1.m3()
```

output:-

```
========== RESTART: C:/Users/student/Desktop/64027/
parent classA: m1 method
parent class B :m2 method
parent classA: m1 method
parent class B :m2 method
parent classC: m3 method
```

3] polymorphism

A]

INPUT:

```
class  polygon:
    def render(self):
        print("render polygon...")

class square(polygon):
    def render(self):
        print("rending square...")
```

```
class circle(polygon):
    def render(self):
        print("rending circle...")


s1=square()
s1.render()


c1=circle()
c1.render()
```

OUTPUT:

```
========== RESTART: C:/Users/student/Desktop/64027/p
rending square...
rending circle...
```

Date:_____

Practical No: 03

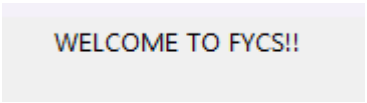
Aim: Write a python program to implement python tkinter elements.

1] To display the normal text using tkinter

INPUT:

```
from tkinter import*
root=Tk()
w=Label(root,text="WELCOME TO FYCS!!")
w.pack()
root.mainloop()
```

OUTPUT:

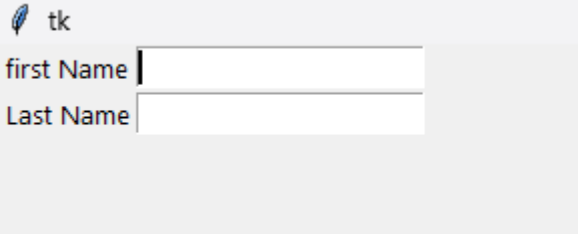


2] To display Textbox of names

INPUT:

```
from tkinter import*
master=Tk()
Label(master,text="first Name").grid(row=0)
Label(master,text="Last Name").grid(row=1)
e1=Entry(master)
e2=Entry(master)
e1.grid(row=0,column=1)
e2.grid(row=1,column=1)
mainloop()
```

OUTPUT:

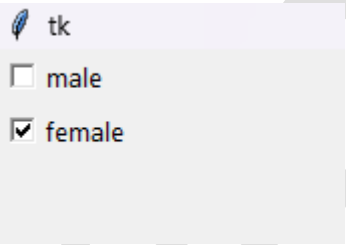


3] To display checkbox

INPUT:

```
from tkinter import*
master=Tk()
var1=IntVar()
Checkbutton(master,text="male",variable=var1).grid(row=0,sticky=W)
var2=IntVar()
Checkbutton(master,text="female",variable=var2).grid(row=1,sticky=W)
mainloop()
```

OUTPUT:

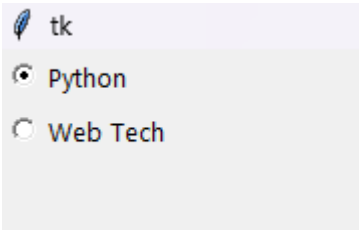


4] To display radiobutton

INPUT:

```
from tkinter import*
master=Tk()
var1=IntVar()
Radiobutton(master,text="Python",variable=var1,value=1).pack(anchor=W)
Radiobutton(master,text="Web Tech",variable=var1,value=2).pack(anchor=W)
mainloop()
```

OUTPUT:

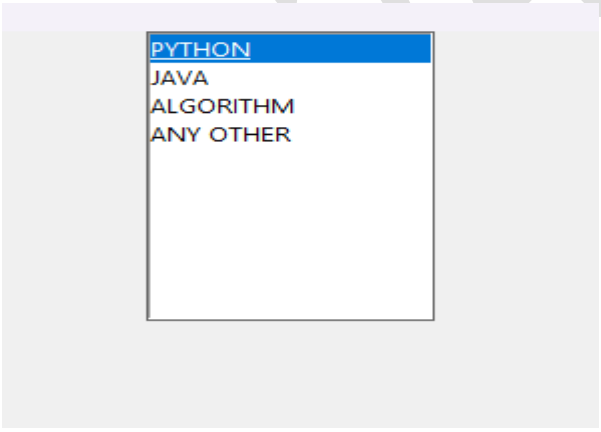


5] To display Listbox

INPUT:

```
from tkinter import*
master=Tk()
Lb=Listbox(master)
Lb.insert(1,"PYTHON")
Lb.insert(2,"JAVA")
Lb.insert(3,"ALGORITHM")
Lb.insert(4,"ANY OTHER")
Lb.pack()
master.mainloop()
```

OUTPUT:



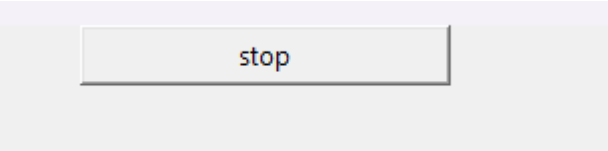
6] To display button

INPUT:

```
import tkinter as tk
m=tk.Tk()
m.title("Counting seconds")
```

```
button=tk.Button(m,text="stop",width=25,command=m.destroy)
button.pack()
m.mainloop()
```

OUTPUT:



7] To create simple Login form

INPUT:

```
from tkinter import*
def submit():
    Username=e1.get ()
    Password=e2.get()
    print(f"User name:{Username},Pass word:{Password}")
def reset():
    e1.delete(0,END)
    e2.delete(0,END)
m=Tk()
m.configure(bg="light blue")
w=Label(m,text='Username').grid(row=0)
w=Label(m,text='Password').grid(row=1)
e1=Entry(m)
e2=Entry(m)
e1.grid(row=0,column=1)
e2.grid(row=1,column=1)
m.title("login Form")
button1=Button(m,text="submit",width=25,command=submit)
button1.grid(row=2,column=0,columnspan=2)
button2=Button(m,text="reset",width=25,command=reset)
button2.grid(row=3,column=0,columnspan=2)
m.mainloop
```

Date:_____

Practical No: 04

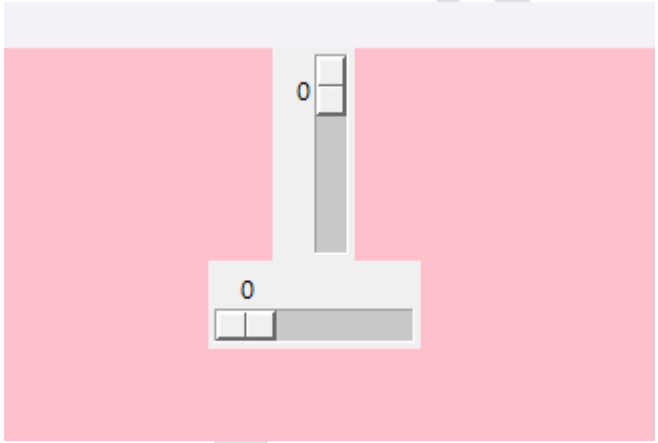
Aim: Write a python program to implement python widget classes.

1] Scale

INPUT:

```
from tkinter import*
master=Tk()
master.configure(bg="pink")
w=Scale(master,from_=0,to=42)
w.pack()
w=Scale(master,from_=0,to=200,orient=HORIZONTAL)
w.pack()
mainloop()
```

OUTPUT:



2] MenuButton

INPUT:

```
from tkinter import*
top=Tk()
top.configure(bg="light yellow")
mb=Menubutton(top,text="info",bg="light blue")
```

```
mb.grid()
mb.menu=Menu(mb,tearoff=0)
mb["menu"]=mb.menu
cVar=IntVar()
aVar=IntVar()
mb.menu.add_checkbutton(label='contact',variable=cVar)
mb.menu.add_checkbutton(label='About',variable=aVar)
mainloop()
```

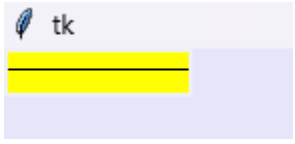
OUTPUT:

3] Canvas line

INPUT:

```
from tkinter import*
master=Tk()
master.configure(bg="lavender")
w=Canvas(master,width=40,height=60,bg="yellow")
w.grid()
canvas_height=20
canvas_width=200
y=int(canvas_height/2)
w.create_line(0,y,canvas_width,y)
master.mainloop()
```

OUTPUT:



4] Canvas square

INPUT:

```
import tkinter as tk
root=tk.Tk()
root.geometry('800x600')
root.title('Canvas Demo-Rectangle')
canvas=tk.Canvas(root,width=600,height=400,bg="white")
canvas.pack(anchor=tk.CENTER,expand=True)
canvas.create_rectangle((100,100),(300,300),fill="light blue",outline="black")
root.mainloop()
```

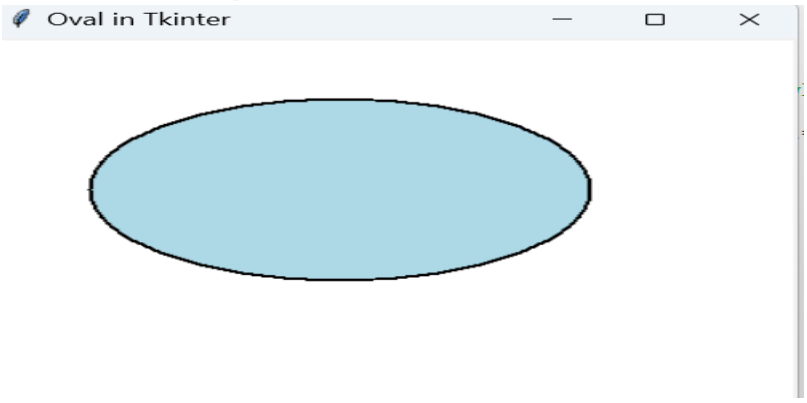
OUTPUT:

5] Canvas

INPUT:

```
import tkinter as tk
root=tk.Tk()
root.geometry('800x600')
root.title('Canvas Demo-Rectangle')
canvas=tk.Canvas(root,width=600,height=400,bg="white")
canvas.pack(anchor=tk.CENTER,expand=True)
canvas.create_rectangle((100,100),(300,300),fill="light blue",outline="black")
root.mainloop()
```

OUTPUT:



Date: _____

Practical No: 05

Aim: Write a python program to implement exception at handling.

---

1)

INPUT:

```
##n1=int(input("Enter First number"))
##n2=int(input("Enter second number"))
##c=n1+n2
##print(c)
try:
    n1=int(input("Enter First number"))
    n2=int(input("Enter second number"))
    c=n1+n2
    print(c)
except:
      print("Enter only numbers.")
```

OUTPUT:

```
---------- RESTART: E:/64018/prac5_py
Enter First number8
Enter second numbero
Enter only numbers. font-color=RED
```

2)

INPUT:

```
x=int(input("Enter first Number"))
y=int(input("Enter second Number"))
try:
    z=x/y
    print(x,'/',y,'=',z)
except:
    print("Alert:Second number cannot be zero")
```

OUTPUT:

```
Enter first Number90
Enter second Number0
Alert:Second number cannot be zero
```

3)

INPUT:

```
try:
    num=int(input("Enter the number"))
    if(num%2)==0:
        print(f"{num} is even")
    else:
        print(f"{num} is odd")
except:
    print(f"Enterd number is float")
```

OUTPUT:

```
========== RESTART: E:/64018/
Enter the number1.4
Enterd number is float
```

4)

INPUT:

```
try:
    fact=1
    num=int(input("Enter the Number"))
    for i in range(1,num+1):
        fact=fact*i
        print('The factorial is:',fact)
except:
    print("Alert:Enter a number value only")
```

OUTPUT:

```
Enter the Numberrt
Alert:Enter a number value only
```

5)INPUT:

```
try:
   a=5
   b=6
   if(a==b):
      print("a is equal to b")
   else:
      print("a is not equal to b")
except:
   print("There is an Error!!!")
```

OUTPUT:

```
a is not equal to b
```

OR

```
try:
   a=5
   b=hghg
   if(a==b):
      print("a is equal to b")
   else:
      print("a is not equal to b")
except:
   print("There is an Error!!!")
```

OUTPUT:

```
There is an Error
```

Date: _____

Practical No: 06

Aim: write a python program to implementing networking.

1] Server

INPUT:

```
import socket

s=socket.socket()

print("socket sucessfully created")

port=40674

s.bind(('',port))

print("socket binded to %s" %(port))

s.listen(5)

print("socket is listening")

while True:

   c,addr=s.accept()

   print('Got connection from',addr)

   c.send(b'Hii Thanks for connecting')

   c.close()
```

OUTPUT:

```
64018/server_py.py ==========
======
socket sucessfully created
socket binded to 40674
socket is listening
```

2] Client

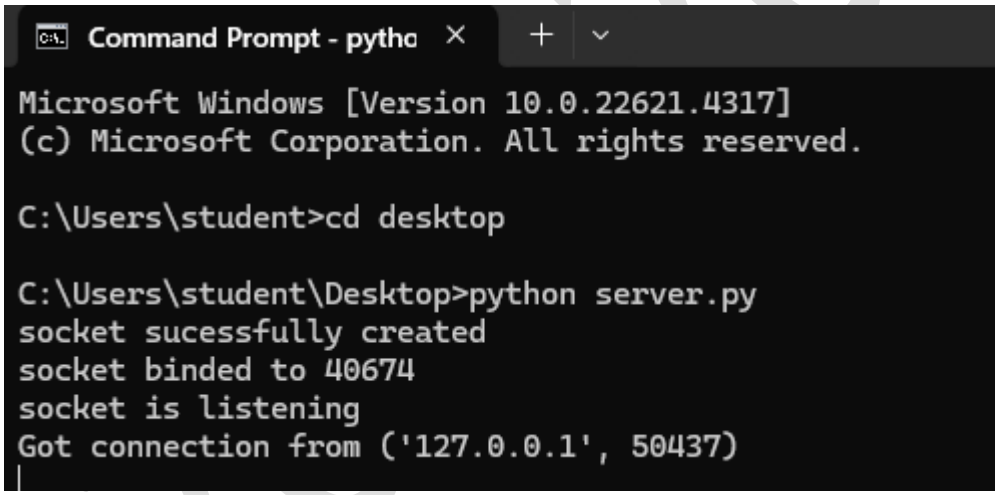
INPUT:

```
import socket
s=socket.socket()
port=40674
s.connect(('127.0.0.1',port))
print(s.recv(40674))
s.close()
```

OUTPUT:

```
===== RESTART: C:/Users/studen
b'Hii Thanks for connecting'
```

If not showing output of client go in command prompt



Date:_____

Practical No: 7

Aim: write a python program to implement database in python (creating database, creating tables).

1] Download and install MySql Connector (<u>give following 2 commands in command prompt</u>)

A] python –m pip install mysql-connector-python

B] python.exe -m pip install --upgrade pip

Command Prompt

```
Microsoft Windows [Version 10.0.17134.1246]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\student>python-m pip install mysql-connector-python
'python-m' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\student>python -m pip install mysql-connector-python
Collecting mysql-connector-python
  Downloading mysql_connector_python-9.2.0-cp311-cp311-win_amd64.whl.metadata (6.2 kB)
Downloading mysql_connector_python-9.2.0-cp311-cp311-win_amd64.whl (16.1 MB)
   ---------------------------------------- 16.1/16.1 MB 7.0 MB/s eta 0:00:00
Installing collected packages: mysql-connector-python
Successfully installed mysql-connector-python-9.2.0

C:\Users\student>python.exe -m pip install --upgrade pip
Requirement already satisfied: pip in c:\users\student\appdata\local\programs\python\python311\lib\site-packages (25.0.1)

C:\Users\student>
```

2] Create database in mysql

```
mysql> create database employee24;
Query OK, 1 row affected (0.00 sec)

mysql> use employee24;
Database changed
mysql>
```

3] create table in python idle

INPUT:

```
import mysql.connector as mysql

db=mysql.connect(user="root",passwd='',host='localhost',database="employee24",charset='utf8')

cur=db.cursor()
```

cur.execute("create table student_info24(Roll_No int primary key,Ename varchar(20))")

print("Sucessfully created a table")

db.close()

OUTPUT:

```
Sucessfully created a table
```

```
mysql> use employee24;
Database changed
mysql> select*from student_info24;
Empty set (0.00 sec)
```

4] insert values in table

import mysql.connector as mysql

db=mysql.connect(user="root",passwd='',host='localhost',database="employee24",charset='utf8')

cur=db.cursor()

cur.execute("insert student_info24 values(2,'XYZ')")

print("value inserted sucessfully")

db.close()

```
value inserted sucessfully
```

```
mysql> select*from student_info24;
+---------+-------+
| Roll_No | Ename |
+---------+-------+
|       1 | ABC   |
|       2 | XYZ   |
|       3 | LMN   |
|       4 | KPA   |
+---------+-------+
4 rows in set (0.00 sec)
```

Date:_____

Practical No: 8

Aim: write a python program to implement database in python (updating tables, and delete tables).

1] Update command

```
import mysql.connector as mysql

db=mysql.connect(user="root",passwd='',host='localhost',database="employee24",charset='utf8')

cur=db.cursor()

cur.execute("UPDATE student_info24 SET Ename='ABC' WHERE ROll_No=3")

print("update sucessfully")

db.close()
```

```
mysql> select*from student_info24;
+---------+-------+
| Roll_No | Ename |
+---------+-------+
|       1 | ABC   |
|       2 | XYZ   |
|       3 | ABC   |
+---------+-------+
3 rows in set (0.00 sec)
```

2] Delete command

```
import mysql.connector as mysql

db=mysql.connect(user="root",passwd='',host='localhost',database="employee24",charset='utf8')

cur=db.cursor()

cur.execute("DELETE from student_info24 WHERE Roll_No=2")

print("delete sucessfully")

db.close()
```

```
mysql> select*from student_info24;
+---------+-------+
| Roll_No | Ename |
+---------+-------+
|       1 | ABC   |
|       3 | ABC   |
+---------+-------+
2 rows in set (0.00 sec)
```

Date:_____

Practical No: 9

Aim: Write python program to implement data science tool (Matplotlib)

NOTE: Install latest version of python like idle 3.13.2

1. **IN Command prompt**
   A] pip install matplotlib

2. **IN Python script**

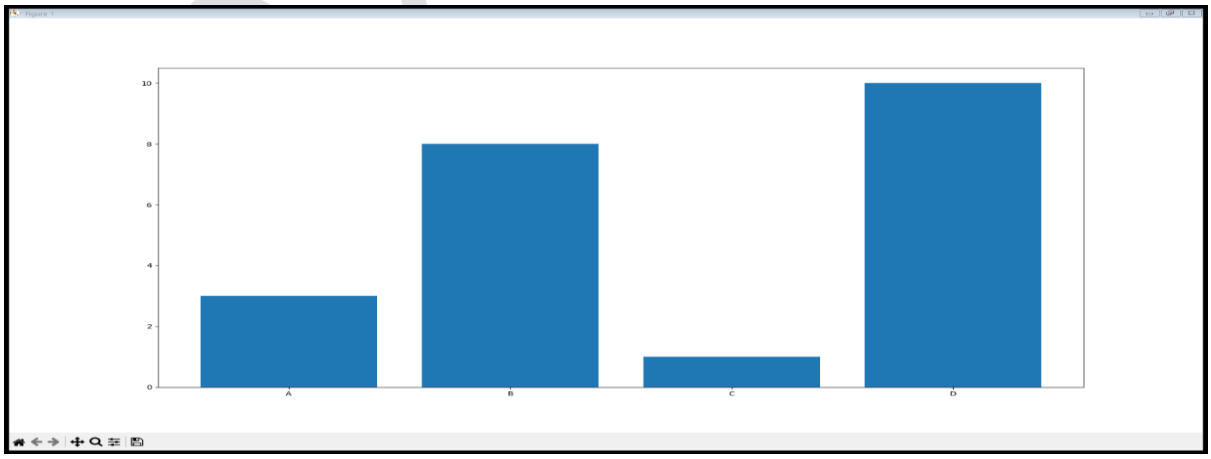
   A] import matplotlib.pyplot as plt

1) Creating Bars

INPUT:

```
import matplotlib.pyplot as plt
import numpy as np
x=np.array(["A","B","C","D"])
y=np.array([3,8,1,10])
plt.bar(x,y)
plt.show()
```

OUTPUT:



2) draw the line in a diagram from position (0, 0) to position (6, 250):

INPUT:

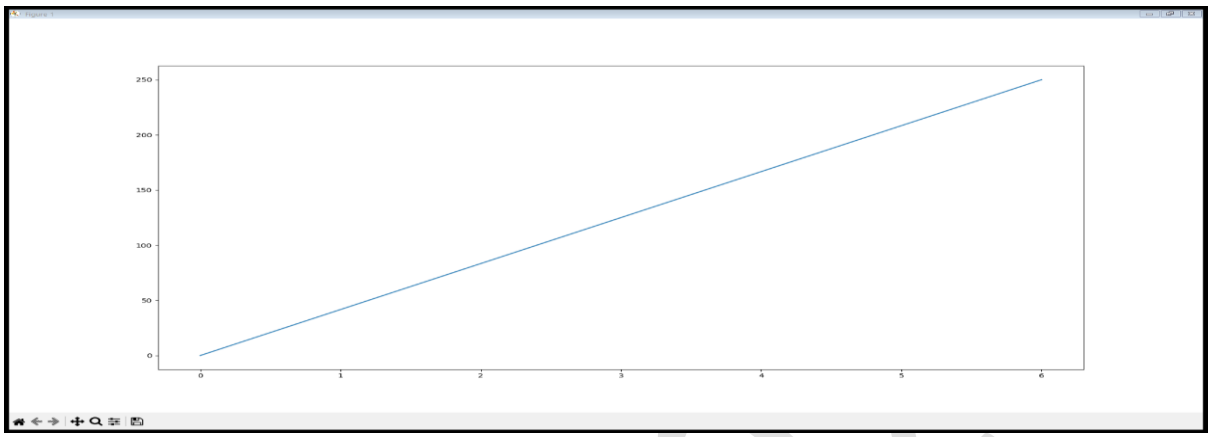
```
import matplotlib.pyplot as plt
import numpy as np
xpoints=np.array([0,6])
```

ypoints=np.array([0,250])

plt.plot(xpoints,ypoints)

plt.show()

OUTPUT:



3) Use a dotted line:

INPUT:

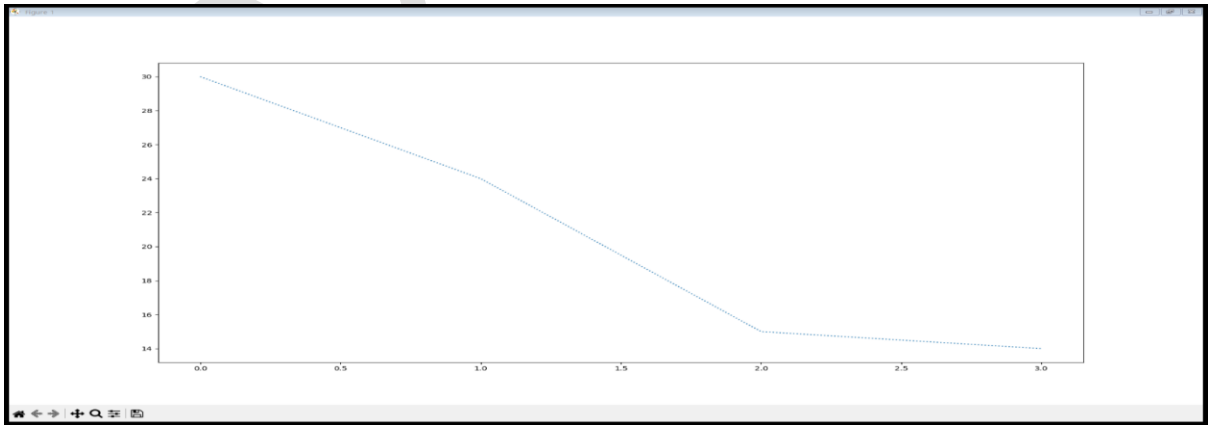
import matplotlib.pyplot as plt

import numpy as np

ypoints=np.array([30,24,15,14])

plt.plot(ypoints, linestyle="dotted")

plt.show()



4) Add labels to the X- and Y- axis:

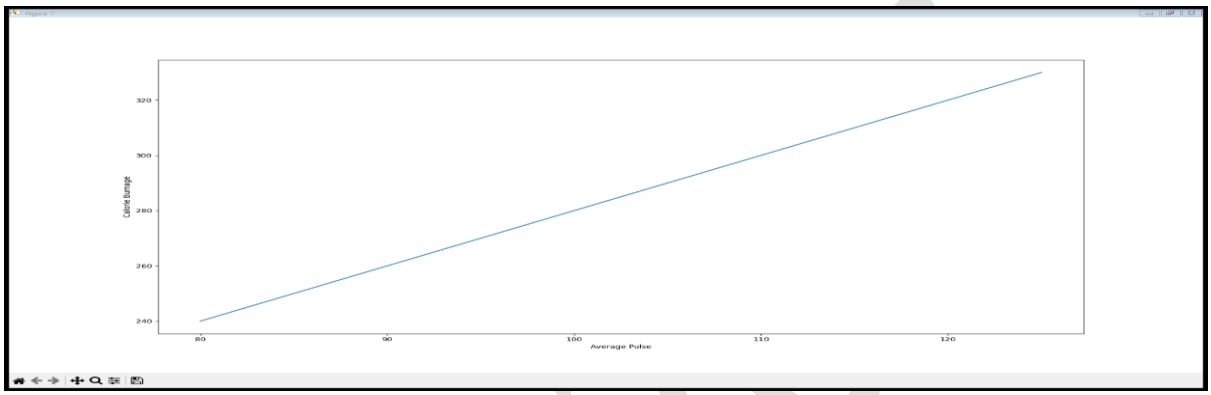
INPUT:

import matplotlib.pyplot as plt

import numpy as np

```
x=np.array([80,85,90,95,100,105,110,115,120,125])
y=np.array([240,250,260,270,280,290,300,310,320,330])
plt.plot(x,y)
plt.xlabel("Average Pulse")
plt.ylabel("Calorie Burnage")
plt.show()
```

OUTPUT:

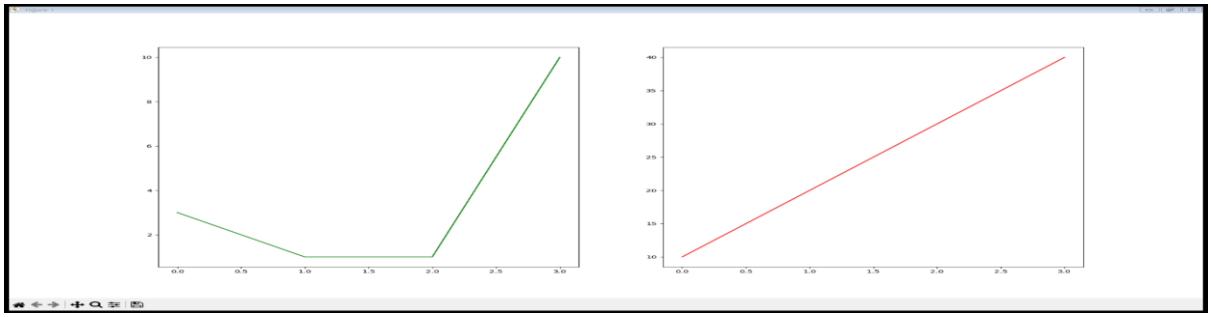


5) Subplot

INPUT:

```
import matplotlib.pyplot as plt
import numpy as np
#plot1:
x=np.array([0,1,2,3])
y=np.array([3,1,1,10])
plt.subplot(1,2,1)
plt.plot(x,y,color="green")
#plot 2:
x=np.array([0,1,2,3])
y=np.array([10,20,30,40])
plt.subplot(1,2,2)
plt.plot(x,y,color="red")
plt.show()
```

OUTPUT:

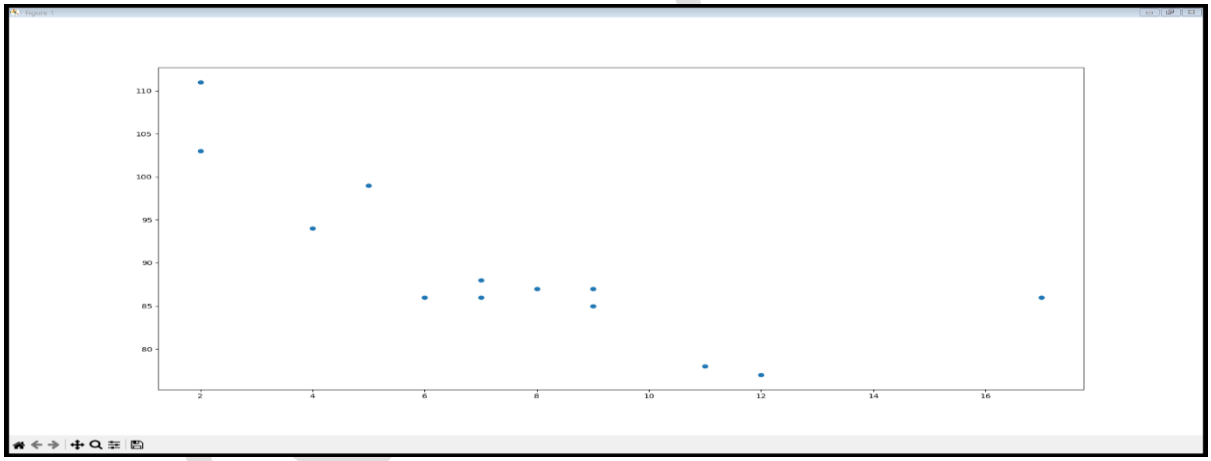


6) Creating Scatter plot

INPUT:

```
import matplotlib.pyplot as plt
import numpy as np
x=np.array([5,7,8,7,2,17,2,9,4,11,12,9,6])
y=np.array([99,86,87,88,111,86,103,87,94,78,77,85,86])
plt.scatter(x,y)
plt.show()
```

OUTPUT:

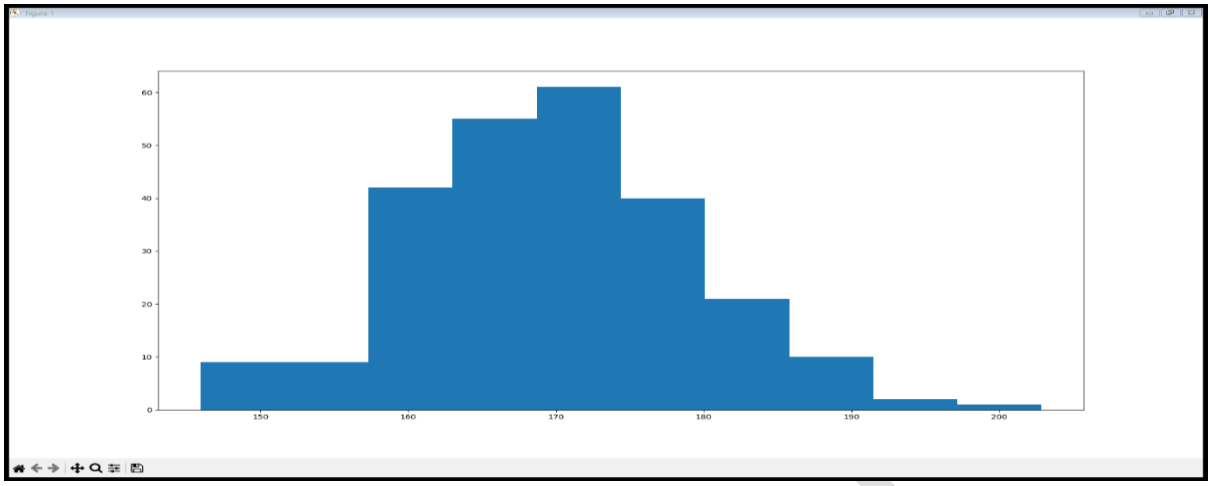


7) Create Histogram

INPUT:

```
import matplotlib.pyplot as plt
import numpy as np
x=np.random.normal(170,10,250)
plt.hist(x)
plt.show()
```

OUTPUT:

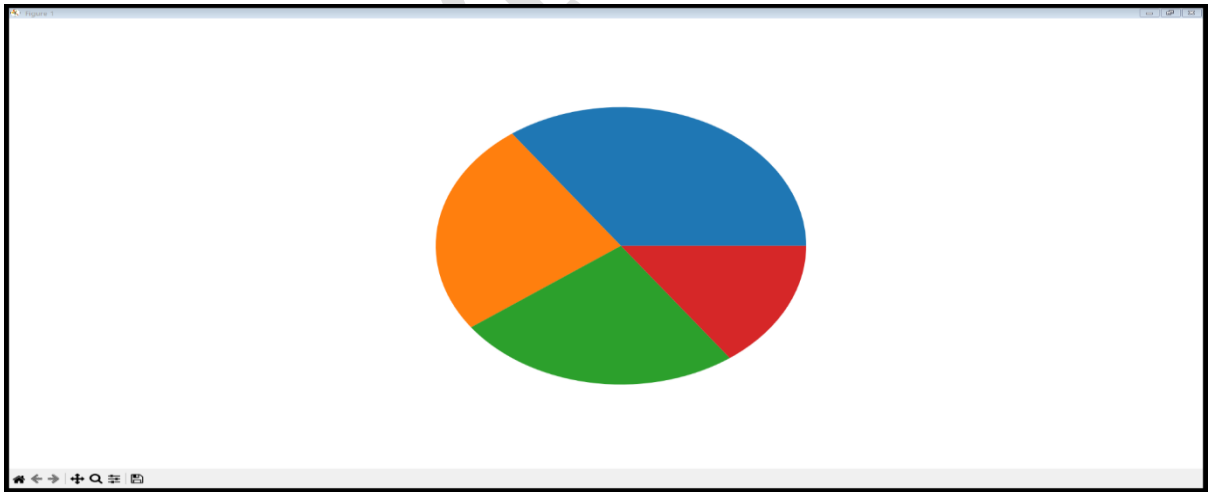


8) Create Pie charts

INPUT:

```
import matplotlib.pyplot as plt

import numpy as np

y=np.array([35,25,25,15])

plt.pie(y)

plt.show()
```

OUTPUT:



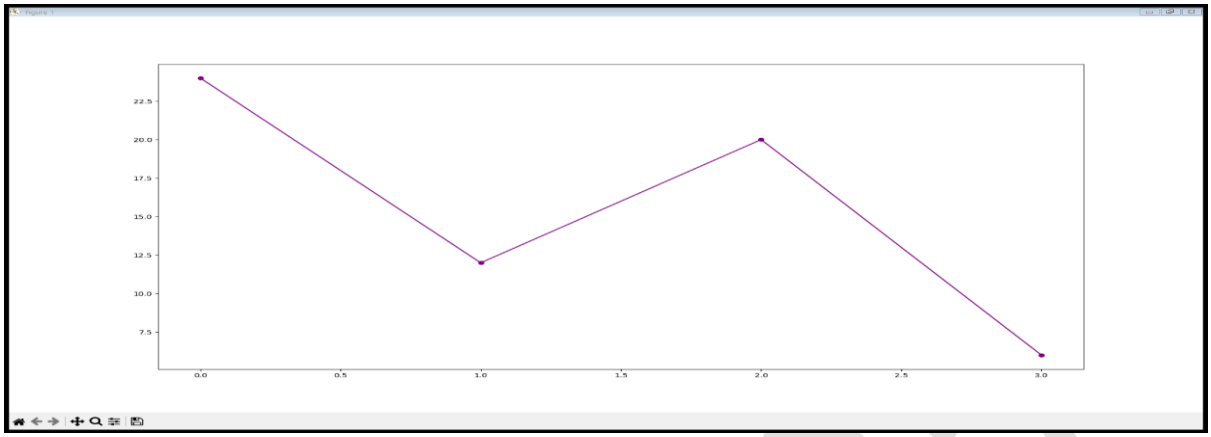
9) Markers

INPUT:

```
import matplotlib.pyplot as plt

import numpy as np

ypoints=np.array([24,12,20,6])

plt.plot(ypoints,marker='o', color=”purple”)
```

plt.show()

OUTPUT:



Date: _____

Practical No: 10

Aim: Write a python program to implement data science tools (spicy and pandas).

******* first download scipy in command prompt******

```
C:\Users\student>pip install scipy
Collecting scipy
  Downloading scipy-1.15.2-cp313-cp313-win_amd64.whl.metadata (60 kB)
Requirement already satisfied: numpy<2.5,>=1.23.5 in c:\users\student\appdata\local\programs\python\python313\lib\site-p
ackages (from scipy) (2.2.2)
Downloading scipy-1.15.2-cp313-cp313-win_amd64.whl (41.0 MB)
   ---------------------------------------- 41.0/41.0 MB 10.6 MB/s eta 0:00:00
Installing collected packages: scipy
Successfully installed scipy-1.15.2
```

***** do practical in python(3.13 version)*****

A] scipy

1] to print pi value

INPUT:

```
from scipy import constants

print(constants.pi)
```

OUTPUT:

```
========== RESTART: E:/64018
3.141592653589793
```

2] to print different operations

INPUT:

```
from scipy import constants

print(constants.minute)

print(constants.hour)

print(constants.day)

print(constants.week)

print(constants.year)

print(constants.Julian_year)
```

OUTPUT:

```
========== RE
60.0
3600.0
86400.0
604800.0
31536000.0
31557600.0
```

B] pandas

1] To display DataFrame

INPUT:

```
import pandas as pd
data={
   "class":["FYCS","SYCS","TYCS"],
   "students":[50,40,45]
   }
#load data into a DataFrame object:
df=pd.DataFrame(data)
print(df)
```

OUTPUT:

```
========== RESTART: E
  class  students
0  FYCS        50
1  SYCS        40
2  TYCS        45
```

2] To read CSV file

INPUT:

```
import pandas as pd
#reading csv file
df=pd.read_csv("E:/64018/student.csv")
```

print(df)

OUTPUT:

```
---------- RESTART: E:/64
      Name  Roll no
0   khanak        1
1    aarya        2
2  pratiksha      3
3    mansi        4
4  hindavi        5
5   riddhi        6
6    disha        7
```