



LORDS INSTITUTE OF ENGINEERING & TECHNOLOGY
(A UGC AUTONOMOUS INSTITUTION)

Approved by AICTE/Affiliated to OU/Estd.2002.Accredited
'A' grade by NAAC Accredited by NBA

PROGRAMMING LANGUAGES II LAB

B.E II /IV SEMESTER

LABORATORY OBSERVATION CUM MANUAL



Name:

Roll No:

Branch:

Semester:

Year:

Subject:

**LORDS INSTITUTE OF ENGINEERING & TECHNOLOGY****(A UGC AUTONOMOUS INSTITUTION)**

Approved by AICTE /Affiliated to OU/ Estd. 2002. Accredited 'A' grade
by NAAC-Accredited by NBA

CERTIFICATE

Certified that this is the Bonafide certificate of the observation done by Mr. / Ms.

.....Roll No.

..... of B.E II Year

..... Semester for the Academic Year 2022-2023

In.....

Laboratory.

Date:

Technical Trainer

HOD

INDEX

[illegible]

**LORDS INSTITUTE OF ENGINEERING & TECHNOLOGY [A]**

Approved by AICTE / Affiliated to Osmania University / Estd.2003.

Accredited 'A' grade by NAAC, Accredited by NBA

Vision

Lords Institute of Engineering and Technology strives for excellence in professional education through quality, innovation and teamwork and aims to emerge as a premier institute in the state and across the nation.

Mission

- To impart quality professional education that meets the needs of present and emerging technological world.
- To strive for student achievement and success, preparing them for life, career and leadership.
- To provide a scholarly and vibrant learning environment that enables faculty, staff and students to achieve personal and professional growth.
- To contribute to advancement of knowledge, in both fundamental and applied areas of engineering and technology.
- To forge mutually beneficial relationships with government organizations, industries, society and the alumni.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**PROGRAMMING LANGUAGE II LABORATORY****PREFACE**

- This manual directs the latest Python tools and techniques to help you tackle the world of data acquisition and analysis. You'll review scientific computing with NumPy, visualization with matplotlib, and machine learning with scikit-learn.
 - This revision is fully updated with new content on social media data analysis, image analysis with OpenCV, and deep learning libraries. Each chapter includes multiple examples demonstrating how to work with each library. At its heart lies the coverage of pandas, for high-performance, easy-to-use data structures and tools for data manipulation
- Manual expertly demonstrates using Python for data processing, management, and information retrieval.

Objective:

- Understand the core concepts of data analysis and the Python ecosystem
- Go in depth with pandas for reading, writing, and processing data
- Use tools and techniques for data visualization and image analysis

Prepared by,

J.M SHANTHI, Technical Trainer,

Skill Development Programme Coordinator, LIET



LORDS INSTITUTE OF ENGINEERING & TECHNOLOGY [A]

Approved by AICTE / Affiliated to Osmania University / Estd.2003.

Accredited 'A' grade by NAAC, Accredited by NBA

S.No	Program Outcomes(POs):
1.	Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering Problems.
2.	Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3.	Design/Development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental Considerations.
4.	Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5.	Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6.	The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7.	Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8.	Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9.	Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10.	Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11.	Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

Ex.No1: Write a Python Preogram to practice some basic operations on library modules

a) Numpy

b)Scipy

(a)Creating a Numpy Array

Program:

```
# Python program for
# Creation of Arrays
import numpy as np

# Creating a rank 1 Array
arr = np.array([1, 2, 3])
print("Array with Rank 1: \n",arr)

# Creating a rank 2 Array
arr = np.array([[1, 2, 3],
                [4, 5, 6]])
print("Array with Rank 2: \n", arr)

# Creating an array from tuple
arr = np.array((1, 3, 2))
print("\nArray created using "
      "passed tuple:\n", arr)
```

Output:

Array with Rank 1:

[1 2 3]

Array with Rank 2:

[[1 2 3]

[4 5 6]]

Array created using passed tuple:

```
[1 3 2]
```

(ii) SciPy

Program:

```
import numpy as np
from scipy import io as sio
array = np.ones((4, 4))
sio.savemat('example.mat', {'ar': array})
data = sio.loadmat('example.mat', struct_as_record=True)
data['ar']
```

Output:

```
array([[ 1.,  1.,  1.,  1.],
       [ 1.,  1.,  1.,  1.],
       [ 1.,  1.,  1.,  1.],
       [ 1.,  1.,  1.,  1.]])
```


Ex.No 2:

Write a Python Program to demonstrate array creation techniques using

a)list

b)tuple

c)resized array

Program:

```
# Python program to demonstrate
# Creation of Array
# importing "array" for array creations
import array as arr

# creating an array with integer type
a = arr.array('i', [1, 2, 3])

# printing original array
print ("The new created array is : ", end = " ")
for i in range(0, 3):
    print (a[i], end = " ")
print()

# creating an array with double type
b = arr.array('d', [2.5, 3.2, 3.3])

# printing original array
print ("The new created array is : ", end = " ")
for i in range(0, 3):
    print (b[i], end = " ")
```

Output:

The new created array is : 1 2 3

The new created array is : 2.5 3.2 3.3

Ex. No 3a: Write a Pandas program to select the 'name' and 'score' columns from the following DataFrame.

Program:

```
import pandas as pd
import numpy as np
exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily',
'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'],
'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

df = pd.DataFrame(exam_data , index=labels)
print("Select specific columns and rows:")
print(df.iloc[[1, 3, 5, 6], [1, 3]])
```

Output:

Select specific columns and rows:

score qualify

b 9.0 no

d NaN no

f 20.0 yes

g 14.5 yes

Ex.No 3b: Write a Pandas program to sort the Data Frame first by;name;in decscenting order,then by in ascending order

Program:

```
import pandas as pd

import numpy as np

exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily',
'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'],

'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],

'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],

'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}

labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

df = pd.DataFrame(exam_data , index=labels)

print("Orginal rows:")

print(df)

df = df.sort_values(by=['name', 'score'], ascending=[False, True])

print("Sort the data frame first by 'name' in descending order, then by 'score' in ascending order:")

print(df)
```

Output:

```
Orginal rows:
   name  score  attempts  qualify
a Anastasia  12.5         1     yes
b    Dima    9.0         3      no
c Katherine  16.5         2     yes
d   James   NaN         3      no
e   Emily    9.0         2      no
f Michael  20.0         3     yes
g Matthew  14.5         1     yes
h   Laura   NaN         1      no
i   Kevin    8.0         2      no
```


U21CS4L3 /PROGRAMMING LANGUAGE II

```
j    Jonas  19.0      1    yes
```

Sort the data frame first by 'name' in descending order, then by 'score' in ascending order:

```
      name score attempts qualify
f  Michael  20.0        3     yes
g  Matthew  14.5        1     yes
h   Laura   NaN        1     no
i   Kevin   8.0         2     no
c Katherine  16.5        2     yes
j   Jonas  19.0        1     yes
d   James   NaN        3     no
e   Emily   9.0         2     no
b   Dima    9.0         3     no
a Anastasia 12.5        1     yes
```


Ex.No.4 a) Write a Pandas program to replace the 'qualify' column contains the values 'yes' and 'no' with True and False.

Program:

```
import pandas as pd
import numpy as np
exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily',
'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'],
'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
df = pd.DataFrame(exam_data , index=labels)
print("Original rows:")
print(df)
print("\nReplace the 'qualify' column contains the values 'yes' and 'no' with
True and False:")
df['qualify'] = df['qualify'].map({'yes': True, 'no': False})
print(df)
```

Output:

Original rows:

	name	score	attempts	qualify
a	Anastasia	12.5	1	yes
b	Dima	9.0	3	no
c	Katherine	16.5	2	yes
d	James	NaN	3	no
e	Emily	9.0	2	no
f	Michael	20.0	3	yes
g	Matthew	14.5	1	yes
h	Laura	NaN	1	no
i	Kevin	8.0	2	no
j	Jonas	19.0	1	yes

U21CS4L3 /PROGRAMMING LANGUAGE II

Replace the 'qualify' column contains the values 'yes' and 'no' with True and False:

	name	score	attempts	qualify
a	Anastasia	12.5	1	True
b	Dima	9.0	3	False
c	Katherine	16.5	2	True
d	James	NaN	3	False
e	Emily	9.0	2	False
f	Michael	20.0	3	True
g	Matthew	14.5	1	True
h	Laura	NaN	1	False
i	Kevin	8.0	2	False
j	Jonas	19.0	1	True

4b) Write a Pandas program to change the name 'James' to 'Suresh' in name column of the data frame.

Program:

```
import pandas as pd
import numpy as np
exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily',
'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'],
'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
df = pd.DataFrame(exam_data , index=labels)
print("Original rows:")
print(df)
print("\nChange the name 'James' to 'Suresh':")
df['name'] = df['name'].replace('James', 'Suresh')
print(df)
```

Output:

Original rows:

	name	score	attempts	qualify
a	Anastasia	12.5	1	yes
b	Dima	9.0	3	no
c	Katherine	16.5	2	yes
d	James	NaN	3	no
e	Emily	9.0	2	no
f	Michael	20.0	3	yes
g	Matthew	14.5	1	yes
h	Laura	NaN	1	no
i	Kevin	8.0	2	no
j	Jonas	19.0	1	yes

Change the name 'James' to 'Suresh':

	name	score	attempts	qualify
a	Anastasia	12.5	1	yes
b	Dima	9.0	3	no

U21CS4L3 /PROGRAMMING LANGUAGE II

c	Katherine	16.5	2	yes
d	Suresh	NaN	3	no
e	Emily	9.0	2	no
f	Michael	20.0	3	yes
g	Matthew	14.5	1	yes
h	Laura	NaN	1	no
i	Kevin	8.0	2	no
j	Jonas	19.0	1	yes

Ex.No 5 a) Write a Pandas program to insert a new column in existing DataFrame.

Program:

```
import pandas as pd
import numpy as np
exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily',
'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'],
'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
df = pd.DataFrame(exam_data , index=labels)
print("Original rows:")
print(df)
color = ['Red','Blue','Orange','Red','White','White','Blue','Green','Green','Red']
df['color'] = color
print("\nNew DataFrame after inserting the 'color' column")
print(df)
```

Output:

Original rows:

	name	score	attempts	qualify
a	Anastasia	12.5	1	yes
b	Dima	9.0	3	no
c	Katherine	16.5	2	yes
d	James	NaN	3	no
e	Emily	9.0	2	no
f	Michael	20.0	3	yes
g	Matthew	14.5	1	yes
h	Laura	NaN	1	no
i	Kevin	8.0	2	no

U21CS4L3 /PROGRAMMING LANGUAGE II

```
j    Jonas  19.0      1    yes
```

New DataFrame after inserting the 'color' column

```
      name  score  attempts  qualify  color
a  Anastasia  12.5        1     yes    Red
b     Dima    9.0         3     no    Blue
c  Katherine  16.5        2     yes  Orange
d     James   NaN         3     no    Red
e     Emily   9.0         2     no   White
f  Michael  20.0         3     yes   White
g  Matthew  14.5         1     yes   Blue
h     Laura   NaN         1     no   Green
i     Kevin   8.0         2     no   Green
j     Jonas  19.0         1     yes    Red
```


5b) Write a Pandas program to iterate over rows in a DataFrame.

Program:

```
import pandas as pd

import numpy as np

exam_data = [{'name':'Anastasia', 'score':12.5}, {'name':'Dima','score':9},
{'name':'Katherine','score':16.5}]

df = pd.DataFrame(exam_data)

for index, row in df.iterrows():

    print(row['name'], row['score'])
```

Output:

```
Anastasia 12.5
Dima 9.0
Katherine 16.5
```


Ex.No.6 Write a three lines of code, you can generate a basic graph using python matplotlib.

```
# importing the required module
import matplotlib.pyplot as plt

# x axis values
x = [1,2,3]
# corresponding y axis values
y = [2,4,1]

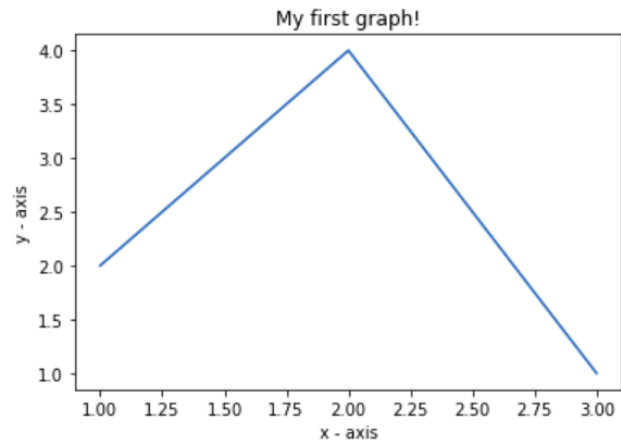
# plotting the points
plt.plot(x, y)

# naming the x axis
plt.xlabel('x - axis')
# naming the y axis
plt.ylabel('y - axis')

# giving a title to my graph
plt.title('My first graph!')

# function to show the plot
plt.show()
```


Output:



Ex.No 6 b) Write a code how to add style to a graph using python matplotlib.

```
# importing all the necessary packages
import numpy as np
import matplotlib.pyplot as plt

# importing the style package
from matplotlib import style

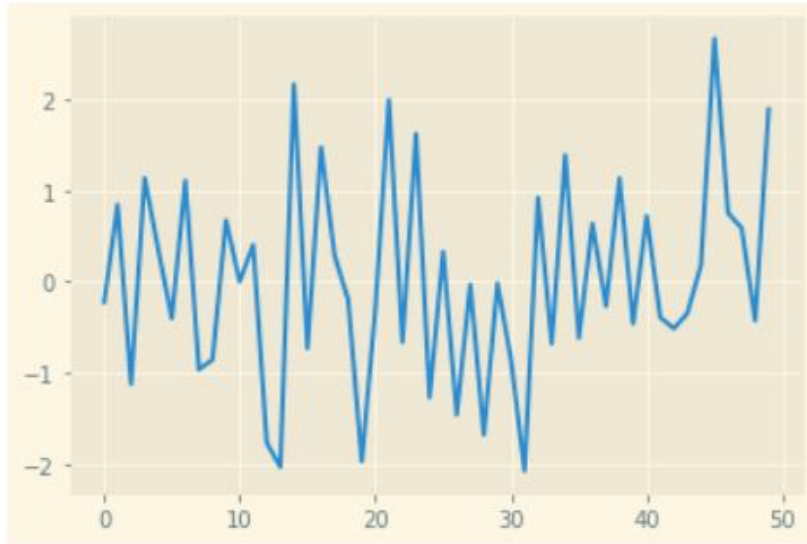
# creating an array of data for plot
data = np.random.randn(50)

# using the style for the plot
plt.style.use('Solarize_Light2')

# creating a plot
plt.plot(data)

# show plot
plt.show()
```


Output:



Ex.No.7 a) Write a code to represent the data in bar graph using python matplotlib.

```
import numpy as np
import matplotlib.pyplot as plt

# creating the dataset
data = {'C':20, 'C++':15, 'Java':30,
        'Python':35}
courses = list(data.keys())
values = list(data.values())

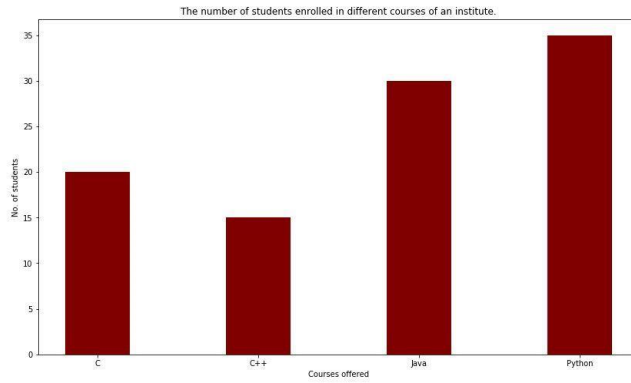
fig = plt.figure(figsize = (10, 5))

# creating the bar plot
plt.bar(courses, values, color ='maroon',
        width = 0.4)

plt.xlabel("Courses offered")
plt.ylabel("No. of students enrolled")
plt.title("Students enrolled in different courses")
plt.show()
```


U21CS4L3 /PROGRAMMING LANGUAGE II

Output:



b) Write a code to represent the data in Histogram using python matplotlib.

Program:

```
from matplotlib import pyplot as plt
import numpy as np
```

```
# Creating dataset
```

```
a = np.array([22, 87, 5, 43, 56,
              73, 55, 54, 11,
              20, 51, 5, 79, 31,
              27])
```

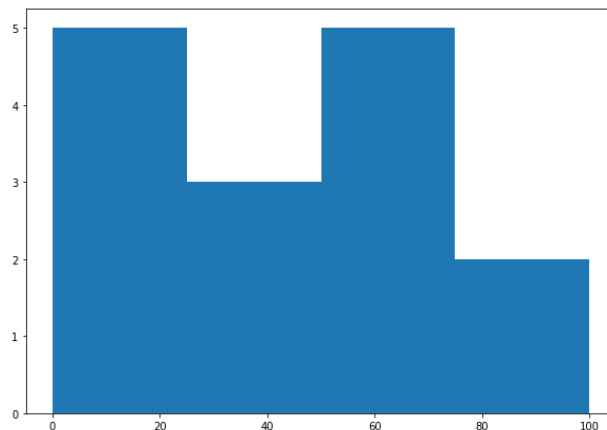
```
# Creating histogram
```

```
fig, ax = plt.subplots(figsize=(10, 7))
ax.hist(a, bins = [0, 25, 50, 75, 100])
```

```
# Show plot
```

```
plt.show()
```

Output:



c) Write a code to represent the data in Scatter plot using python matplotlib.

Program:

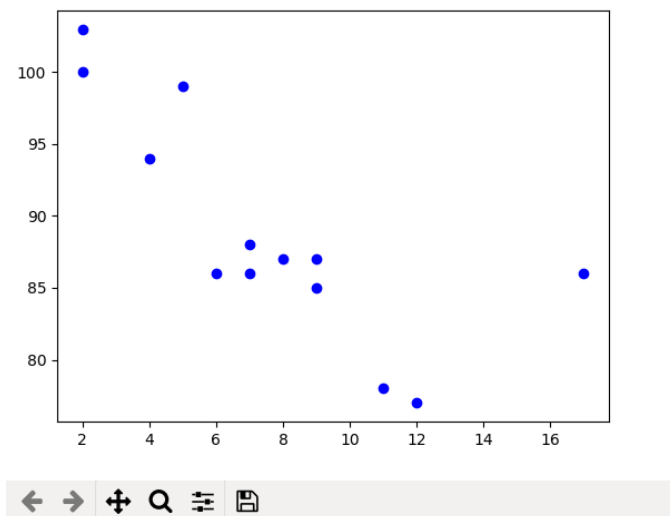
```
import matplotlib.pyplot as plt
x =[5, 7, 8, 7, 2, 17, 2, 9,
    4, 11, 12, 9, 6]

y =[99, 86, 87, 88, 100, 86,
    103, 87, 94, 78, 77, 85, 86]

plt.scatter(x, y, c ="blue")

# To show the plot
plt.show()
```

Output:



Ex.No.8 a) Write a code to represent the data in Area plot using python matplotlib.

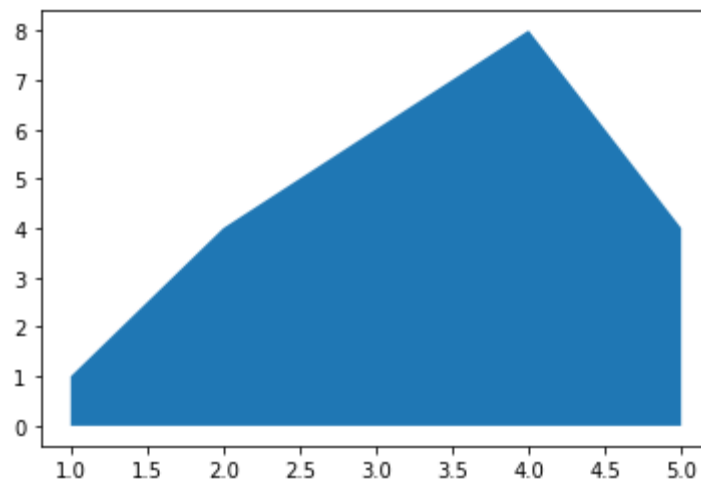
Program:

```
# library
import numpy as np
import matplotlib.pyplot as plt

# Create data
x=range(1,6)
y=[1,4,6,8,4]

# Area plot
plt.fill_between(x, y)
plt.show()
```

Output:



b)Write a code to represent the data in pie chart using python matplotlib.

Program:

```
# Import libraries
from matplotlib import pyplot as plt
import numpy as np

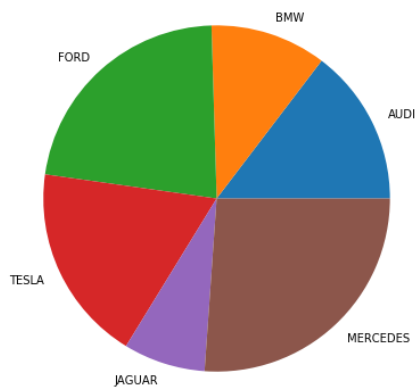
# Creating dataset
cars = ['AUDI', 'BMW', 'FORD',
        'TESLA', 'JAGUAR', 'MERCEDES']

data = [23, 17, 35, 29, 12, 41]

# Creating plot
fig = plt.figure(figsize =(10, 7))
plt.pie(data, labels = cars)

# show plot
plt.show()
```

Output:



c)Write a code to represent the data working with multiple plots using python matplotlib.

Program:

```
# importing libraries
import matplotlib.pyplot as plt
import numpy as np
import math

# Get the angles from 0 to 2 pie (360 degree) in narray object
X = np.arange(0, math.pi*2, 0.05)

# Using built-in trigonometric function we can directly plot
# the given cosine wave for the given angles
Y1 = np.sin(X)
Y2 = np.cos(X)
Y3 = np.tan(X)
Y4 = np.tanh(X)

# Initialise the subplot function using number of rows and columns
figure, axis = plt.subplots(2, 2)

# For Sine Function
axis[0, 0].plot(X, Y1)
axis[0, 0].set_title("Sine Function")

# For Cosine Function
axis[0, 1].plot(X, Y2)
axis[0, 1].set_title("Cosine Function")

# For Tangent Function
```

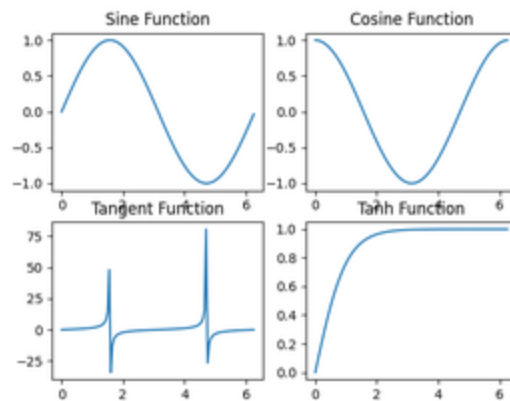


```
axis[1, 0].plot(X, Y3)
axis[1, 0].set_title("Tangent Function")

# For Tanh Function
axis[1, 1].plot(X, Y4)
axis[1, 1].set_title("Tanh Function")

# Combine all the operations and display
plt.show()
```

Output



Ex.No.9a) Write a program to give basic introduction about CV library?

About Open CV:

Open CV is an open source computer vision library. It has been widely adopted all over the world for its ability of processing real-time computer vision. It has multiple language interfaces and thus can be used in Windows, Linux, Mac OSX, Android and iOS.

Why this application note is helpful?

Since Open CV supports almost all mainstream operating systems, steps for setting up the development environment are different from platform to platform. The files included in the Open CV library can be different for each version of the library as well. Therefore, having a correct setup guide for Open CV is crucial to the success of software development. This application note will provide a step-by-step guide to help the reader setup the Open CV 2.4.3 development environment in Microsoft Visual Studio 2010 under 64-bit Windows 8.

Program:

```
#include<opencv\cv.h>
#include<opencv\highgui.h>

using namespace
cv;int main() {
    Mat image;

    VideoCapture
    cap;cap.open(0);
```


U21CS4L3 /PROGRAMMING LANGUAGE II

```
namedWindow("window",CV_WINDOW_AUT
OSIZE); while(1){
    cap >> image;
    imshow("window",
    image);waitKey(33);
}
return 0;
}
```

**9b)Write a python program to draw a circleand ellipse using open cv
import cv2**

Program:

```
import numpy as np
# Load image
image = cv2.imread('C://gfg//images//blobs.jpg', 0)

# Set our filtering parameters
# Initialize parameter setting using cv2.SimpleBlobDetector
params = cv2.SimpleBlobDetector_Params()

# Set Area filtering parameters
params.filterByArea = True
params.minArea = 100

# Set Circularity filtering parameters
params.filterByCircularity = True
params.minCircularity = 0.9

# Set Convexity filtering parameters
params.filterByConvexity = True
params.minConvexity = 0.2

# Set inertia filtering parameters
params.filterByInertia = True
params.minInertiaRatio = 0.01

# Create a detector with the parameters
detector = cv2.SimpleBlobDetector_create(params)
```



```
# Detect blobs
keypoints = detector.detect(image)

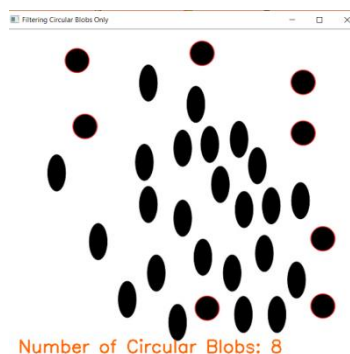
# Draw blobs on our image as red circles
blank = np.zeros((1, 1))
blobs = cv2.drawKeypoints(image, keypoints, blank, (0, 0, 255),

                           cv2.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS)

number_of_blobs = len(keypoints)
text = "Number of Circular Blobs: " + str(len(keypoints))
cv2.putText(blobs, text, (20, 550),
            cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 100, 255), 2)

# Show blobs
cv2.imshow("Filtering Circular Blobs Only", blobs)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Output:



9c) Write a python program to see how we can resize the image using cv

Program:

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

image = cv2.imread(r"D:\sims\eb\sim21\EB-ML-06-10-2022-Test-Output-15\PERFORATION\Overkill\Fail\Blister 1 2022-03-12 12-59-43.859 TO M0 G0 3 PERFORATION Mono.bmp", 1)
# Loading the image

half = cv2.resize(image, (0, 0), fx = 0.1, fy = 0.1)
bigger = cv2.resize(image, (1050, 1610))

stretch_near = cv2.resize(image, (780, 540),
                           interpolation = cv2.INTER_LINEAR)

Titles = ["Original", "Half", "Bigger", "Interpolation Nearest"]
images = [image, half, bigger, stretch_near]
count = 4

for i in range(count):
    plt.subplot(2, 2, i + 1)
    plt.title(Titles[i])
    plt.imshow(images[i])

plt.show()
```


10) Write a Python Program Add or Blend Two images with different weights using open cv

Program:

```
import cv2

# read two images
src1 = cv2.imread('image1.png', cv2.IMREAD_COLOR)
src2 = cv2.imread('image2.png', cv2.IMREAD_COLOR)

# add or blend the images
dst = cv2.addWeighted(src1, 1, src2, 1, 0.0)

# save the output image
cv2.imwrite('image.png', dst)
```

Output:

