M. No	Date	IDX	PROGRAM	Sign & Marks
		Python Programs on Numpy, Pandas		
		1	Write a python program to create, display, append, insert and reverse the order of the items in an array.	
		2	Write a program to add, transpose and multiply two matrices	
		3	Write a python program to demonstrate Numpy arrays creation using array() function	
		4	Demonstrate the use of ndim, shape, size, dtype	
		5	Python program to demonstrate basic slicing, integer and Boolean indexing	
		6	Python program to find min, max, sum, cumulative sum of array	
		7	Explore data through pandas data frame	
		8	Apply head() function to pandas data frame	
		9	Perform various data selection operations on data frame	
		10	Write a Python programming to create a pie chart with a title	
		11	Select any two columns from data frame and observe change in one attribute with respect to other attribute with scatter plot in matplotlib.	

Program 1

AIM : Write a python program to create, display, append, insert and reverse the order of the items in an array.

PROGRAM :

```
import numpy as np
arr=np.array([10,20,30])
print("Array created with the elements : ", arr)
ele=int(input(" Enter the element to insert :  "))
pos=int(input(" Enter the position to insert :  "))
arr = np.insert(arr, pos, ele)
print(f"Array after inserting {ele} at index {pos}:", arr)
ele=int(input(" Enter the element to append :  "))
arr=np.append(arr, ele)
print(f"Array after appending {ele} is :", arr)
print("Reverse of array is : ",arr[::-1])
```

OUTPUT :

```
Array created with the elements :  [10 20 30]
 Enter the element to insert :  50
 Enter the position to insert :  2
Array after inserting 50 at index 2: [10 20 50 30]
 Enter the element to append :  70
Array after appending 70 is : [10 20 50 30 70]
Reverse of array is :  [70 30 50 20 10]
```

Program 2

AIM : Write a program to add, transpose and multiply two matrices

PROGRAM :

```
def display(arr):
    for i in range(3):
        for j in range(3):
            print(arr[i][j],end=' ')
        print()

import numpy as np
arr=np.arange(1,10)
#arr1 =np.random.randint(1,10,(3,3))
#arr2 =np.random.randint(1,10,(3,3))
arr1=arr.reshape(3,3)
print("Matrix A is : ")
```

```
display(arr1)
arr=np.arange(10,19)
arr2=arr.reshape(3,3)
print("\nMatrix B is : ")
display(arr2)
print(" \nAddition of two matrices is ")
display(arr1+arr2)
print("\nMultiplication of two matrices is ")
display(arr1@arr2)
print("\nTranspose of Matrix A is ")
display(arr1.transpose())
```

OUTPUT

```
Matrix A is :
1 2 3
4 5 6
7 8 9

Matrix B is :
10 11 12
13 14 15
16 17 18

Addition of two matrices is
11 13 15
17 19 21
23 25 27

Multiplication of two matrices is
84 90 96
201 216 231
318 342 366

Transpose of Matrix A is
1 4 7
2 5 8
3 6 9
```

Program 3

AIM : Write a python program to demonstrate different types of Numpy arrays creation using array() function

PROGRAM :

```
import numpy as np
n=int(input("Enter number of elements : "))
a = np.array(np.random.randint(1,100,n))
print("array created with ",n," elements is ");
for i in a:
    print(i,end=' ')
r,c=map(int,input("\nEnter the size of Matrix :").split())
x=np.array(np.random.randint(1,100,(r,c)))
print("The matrix of size %d X %d is "%(r,c))
print(x)
```

OUTPUT :

```
Enter number of elements : 5
array created with  5  elements is
44 32 81 78 15
Enter the size of Matrix :4 3
The matrix of size 4 X 3 is
[[40 93 47]
 [ 2 93 86]
 [40 32 17]
 [34 47 14]]
```

Program 4

AIM : Demonstrate the use of ndim, shape, size, dtype

PROGRAM :

```
import numpy as np
array1 = np.arange(10,100,10)
array2 = np.ones((3,2))
array3 = np.zeros((3,4))
print(" Array1 is : ",array1)
print(" Dimention of Array1 is : ",array1.ndim)
print(" Shape of Array1 is : ",array1.shape)
print(" Size of Array1 is : ",array1.size)
print(" Datatype of Array1 is : ",array1.dtype)
print(" Item size of Array1 is : ",array1.itemsize)

print(" Array2 is : ",array2)
print(" Dimention of Array2 is : ",array2.ndim)
print(" Shape of Array2 is : ",array2.shape)
print(" Size of Array2 is : ",array2.size)
print(" Datatype of Array2 is : ",array2.dtype)
print(" Item size of Array2 is : ",array2.itemsize)

print(" Array3 is : ",array3)
print(" Dimention of Array3 is : ",array3.ndim)
print(" Shape of Array3 is : ",array3.shape)
print(" Size of Array3 is : ",array3.size)
print(" Datatype of Array3 is : ",array3.dtype)
print(" Item size of Array3 is : ",array3.itemsize)
```

OUTPUT:

```
Array1 is :  [10 20 30 40 50 60 70 80 90]
Dimention of Array1 is :  1
Shape of Array1 is :  (9,)
Size of Array1 is :  9
Datatype of Array1 is :  int64
Item size of Array1 is :  8
```

```
Array2 is :  [[1. 1.]
[1. 1.]
[1. 1.]]
Dimention of Array2 is :  2
Shape of Array2 is :  (3, 2)
Size of Array2 is :  6
Datatype of Array2 is :  float64
Item size of Array2 is :  8

Array3 is :  [[0. 0. 0. 0.]
[0. 0. 0. 0.]
[0. 0. 0. 0.]]
Dimention of Array3 is :  2
Shape of Array3 is :  (3, 4)
Size of Array3 is :  12
Datatype of Array3 is :  float64
Item size of Array3 is :  8
```

Program 5

AIM : Python program to demonstrate basic slicing, integer and Boolean indexing

PROGRAM :

```
import numpy as np
x=np.array(np.random.randint(1,100,(4,3)))
print("The Marks of 4 students in 3 subjects is ")
print(x)
print("Marks of 3rd student in subject2 is ")
print(x[2,1])
print("Marks of all students in subject3 is ")
print(x[:,2])
print("Marks of 2nd student in all subjects is ")
print(x[1,:])
print("Marks of first 2 students in sub2 & sub3 is ")
print(x[0:2,1:3])
print("Students result(Fail/pass)in Sub2 are ")
print(x[:,1]>40)
print("Stu2 and stu4 marks in all subjects are ")
print(x[[False,True, False, True]])
```

OUTPUT:

```
The Marks of 4 students in 3 subjects is
[[65 91 49]
 [88  5 52]
 [44 15 61]
 [75 80 92]]
Marks of 3rd student in subject2 is
15
Marks of all students in subject3 is
[49 52 61 92]
Marks of 2nd student in all subjects is
[88  5 52]
Marks of first 2 students in sub2 & sub3 is
[[91 49]
 [ 5 52]]
Students result(Fail/pass)in Sub2 are
[ True False False  True]
Stu2 and stu4 marks in all subjects are
[[88  5 52]
 [75 80 92]]
```

Program 6

AIM :. Python program to find min, max, sum, cumulative sum of array

PROGRAM :

```
import numpy as np
x=np.array(np.random.randint(1,100,10))
print("The Given array is elements are : ")
print(x)
print("Maximum Element in the Array : ")
print(x.max())
print("Minimum Element in the Array : ")
print(x.min())
print("Sum of the Elements in the Array : ")
print(x.sum())
print("Cummulative Sum of the Array : ")
print(np.cumsum(x))
```

OUTPUT:

```
The Given array is elements are :
[33 21 27 36 87 16 38 79 56 59]
Maximum Element in the Array :
87
Minimum Element in the Array :
16
Sum of the Elements in the Array :
452
Cummulative Sum of the Array :
[ 33  54  81 117 204 220 258 337 393 452]
```

Program 7

AIM : Explore data through pandas data frame

Demonstrate how to download dataset and how to create DataFrame

i. Write a Pandas program to get the first 3 rows of a DataFrame

ii. Write a Pandas program to select the specified columns and rows from a given data frame.

iii. Write a Pandas program to select the rows where the score is missing, i.e. is NaN.

iv. Write a Pandas program to insert a new column in existing DataFrame.

PROGRAM :

I. Write a Pandas program to get the first 3 rows of a DataFrame

```
import pandas as pd
import numpy as np

exam_data  = {'name': ['Hema', 'Latha', 'Reddy', 'Jay', 'Madhu'],
              'score': [90, 70, np.nan, np.nan, 30 ],
              'attempts': [1,2,1,1,1],
              'qualify': ['yes', 'yes', 'no', 'no', 'no']}
labels = ['a', 'b', 'c', 'd', 'e']
df = pd.DataFrame(exam_data , index=labels)
print("First three rows of the data frame:")
print(df.iloc[:3])
```

II. Write a Pandas program to select the specified columns and rows from a given data frame.

```
print("Select specific columns and rows:")
print(df.iloc[[0, 4 ], [0, 1]])
```

III. Write a Pandas program to select the rows where the score is missing, i.e. is NaN.

```
print("Rows where score is missing:")
print(df[df['score'].isnull()])
```

IV. Write a Pandas program to insert a new column in existing DataFrame.

```
color = ['Green','Blue','Orange','Red','White']
df['color'] = color
print("\nNew DataFrame after inserting the 'color' column")
print(df)
```

OUTPUT :

I:

First three rows of the data frame:

	name	score	attempts	qualify
a	Ravi	90.0	1	yes
b	Venkat	70.0	2	yes
c	Rajesh	NaN	1	no

II:

Select specific columns and rows:

	name	score
a	Ravi	90.0
e	Madhu	30.0

III:

Rows where score is missing:

	name	score	attempts	qualify
c	Rajesh	NaN	1	no
d	Jay	NaN	1	no

IV:

New DataFrame after inserting the 'color' column

	name	score	attempts	qualify	color
a	Ravi	90.0	1	yes	Green
b	Venkat	70.0	2	yes	Blue
c	Rajesh	NaN	1	no	Orange
d	Jay	NaN	1	no	Red
e	Madhu	30.0	1	no	White

Program 8

AIM : Apply head() function to pandas data frame

PROGRAM :

```
import pandas as pd
lang = ['Java', 'Python', 'PHP', 'JavaScript', 'C#', 'C++']
rating = [22.2, 17.6, 8.8, 8, 7.7, 6.7]
df = pd.DataFrame(rating , index=lang)
df=df.rename({0:'Rating'},axis='columns')
df.index.name='Languages'
print(" Top 3 Languages with their rating are " )
print(df.head(3))
```

OUTPUT :

```
           Rating
Languages
Java         22.2
Python       17.6
PHP           8.8
```

Program 9

AIM : Perform various data selection operations on data frame

PROGRAM :

```
import pandas as pd

data = {'Name': ['Rajesh', 'Venkat', 'Samba', 'David'],
        'Age': [25, 30, 22, 28],
        'City': ['New York', 'London', 'Paris', 'Tokyo']}
df = pd.DataFrame(data)
print(df)

print("Select the row with index 1 (Venkat)")
print(df.loc[1])

print("Select multiple rows by label ")
print(df.loc[[0, 2]])

print("Select a range of rows by label ")
print(df.loc[1:3])

print("Select the row at position 1 (Venkat)")
print(df.iloc[1])

print("Select multiple rows by integer position")
print(df.iloc[[0, 2]])

print("Select a range of rows by integer position")
print(df.iloc[1:3])

print("Select the 'Name' column ")
print(df.loc[:, 'Name'])

print("Select multiple columns by label ")
print(df.loc[:, ['Name', 'City']])

print("Select a range of columns by label ")
print(df.loc[:, 'Age':'City'])


print("Select the column at position 0 (Name)")
print(df.iloc[:, 0])
```

```
print("Select multiple columns by integer position ")
print(df.iloc[:, [0, 2]])

print("Select a range of columns by integer position ")
print(df.iloc[:, 1:3])

print("Select rows where Age is greater than 25 ")
print(df[df['Age'] > 25])

print("Select rows where City is 'London' or 'Paris'")
print(df[df['City'].isin(['London', 'Paris'])])
print(df.iloc[1:3])
```

OUTPUT :

```
   Name  Age      City
0  Rajesh   25  New York
1  Venkat   30    London
2   Samba   22     Paris
3   David   28     Tokyo
Select the row with index 1 (Venkat)
Name    Venkat
Age         30
City    London
Name: 1, dtype: object
Select multiple rows by label
    Name  Age      City
0  Rajesh   25  New York
2   Samba   22     Paris
Select a range of rows by label
    Name  Age    City
1  Venkat   30  London
2   Samba   22   Paris
3   David   28   Tokyo
Select the row at position 1 (Venkat)
Name    Venkat
Age         30
City    London
Name: 1, dtype: object
Select multiple rows by integer position
    Name  Age      City
0  Rajesh   25  New York
2   Samba   22     Paris
Select a range of rows by integer position
    Name  Age    City
1  Venkat   30  London
```

2 Samba 22 Paris
Select the 'Name' column
0 Rajesh
1 Venkat
2 Samba
3 David
Name: Name, dtype: object
Select multiple columns by label
Name City
0 Rajesh New York
1 Venkat London
2 Samba Paris
3 David Tokyo
Select a range of columns by label
Age City
0 25 New York
1 30 London
2 22 Paris
3 28 Tokyo
Select the column at position 0 (Name)
0 Rajesh
1 Venkat
2 Samba
3 David
Name: Name, dtype: object
Select multiple columns by integer position
Name City
0 Rajesh New York
1 Venkat London
2 Samba Paris
3 David Tokyo
Select a range of columns by integer position
Age City
0 25 New York
1 30 London
2 22 Paris
3 28 Tokyo
Select rows where Age is greater than 25
Name Age City
1 Venkat 30 London
3 David 28 Tokyo
Select rows where City is 'London' or 'Paris'
Name Age City
1 Venkat 30 London
2 Samba 22 Paris
Name Age City
1 Venkat 30 London
2 Samba 22 Paris

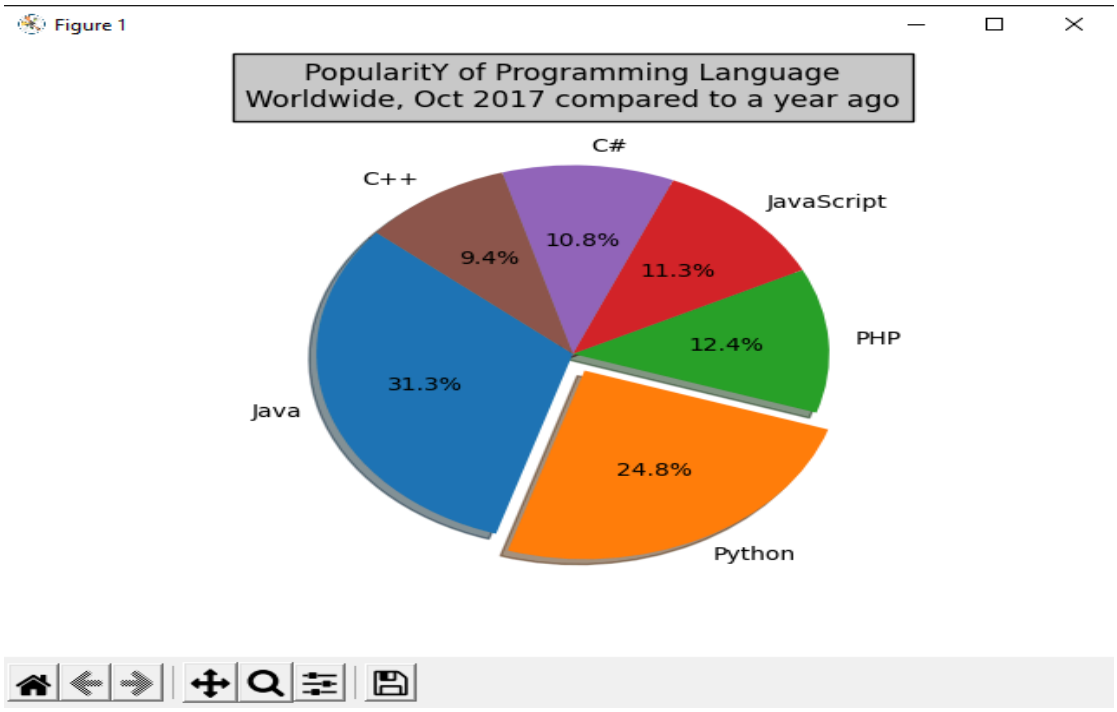
Program 10

AIM : Write Python programming to create a pie chart with a title.

PROGRAM :

```
import matplotlib.pyplot as plt
# Plot data
languages = 'Java', 'Python', 'PHP', 'JavaScript', 'C#', 'C++'
popurarity = [22.2, 17.6, 8.8, 8, 7.7, 6.7]
#colors = ['red', 'gold', 'yellowgreen', 'blue', 'lightcoral', 'lightskyblue']
colors = ["#1f77b4", "#ff7f0e", "#2ca02c", "#d62728", "#9467bd", "#8c564b"]
# explode 1st slice
explode = (0, 0.1, 0, 0, 0, 0)
# Plot
plt.pie(popurarity, explode=explode, labels=languages, colors=colors,
autopct='%1.1f%%', shadow=True, startangle=140)
plt.title("PopularitY of Programming Language\n" + "Worldwide, Oct 2017 compared to a
year ago", bbox={'facecolor':'0.8', 'pad':5})
plt.show()
```

OUTPUT :



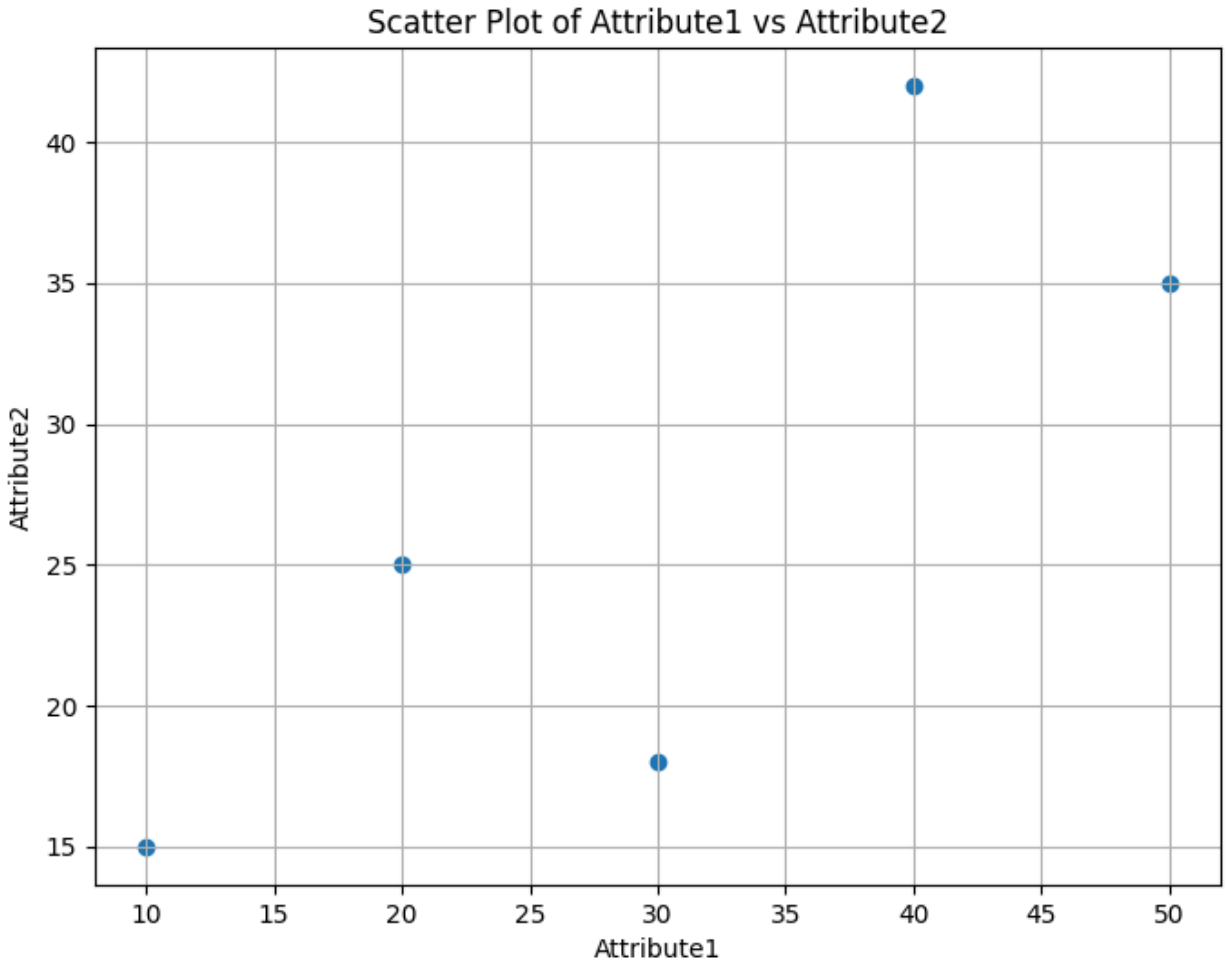
Program 11

AIM : Select any two columns from data frame and observe change in one attribute with respect to other attribute with scatter plot in matplotlib.

PROGRAM :

```
import pandas as pd
import matplotlib.pyplot as plt

data = {'Attribute1': [10, 20, 30, 40, 50],
        'Attribute2': [15, 25, 18, 42, 35],
        'Attribute3': ['A', 'B', 'A', 'C', 'B']}
df = pd.DataFrame(data)


x_attribute = 'Attribute1'
y_attribute = 'Attribute2'


plt.figure(figsize=(8, 6))
plt.scatter(df[x_attribute], df[y_attribute])

plt.xlabel(x_attribute)
plt.ylabel(y_attribute)
plt.title(f'Scatter Plot of {x_attribute} vs {y_attribute}')

plt.grid(True)
plt.show()
```

OUTPUT:


Scatter Plot of Attribute1 vs Attribute2
Attribute2
Attribute1
40
35
30
25
20
15
10
15
20
25
30
35
40
45
50