

NUMPY PROGRAM – Unit 6

21	Write a NumPy program to create an array of the integers from 30 to 70.
22	Write a NumPy program to create a 3x4 matrix filled with values from 10 to 21
23	Write a NumPy program to create a 10x10 matrix, in which the elements on the borders will be equal to 1, and inside 0.
24	Write a NumPy program to compute sum of all elements, sum of each column and sum of each row of a given array.
25	Input a 4*4 matrix and Write a NumPy program to extract third and fourth elements of the first and second rows from a given (4x4) array.

21	<pre>import numpy as np array=np.arange(30,71) print("Array of the integers from 30 to70") print(array)</pre>
22	<pre>import numpy as np m= np.arange(10,22).reshape((3, 4)) print(m)</pre>
23	<pre>import numpy as np x = np.ones((10, 10)) x[1:-1, 1:-1] = 0 print(x)</pre>
24	<pre>import numpy as np x = np.array([[0,1],[2,3]]) print("Original array:") print(x) print("Sum of all elements:") print(np.sum(x)) print("Sum of each column:") print(np.sum(x, axis=0)) print("Sum of each row:") print(np.sum(x, axis=1))</pre>
25	<pre>import numpy as np arra_data = np.arange(0,16).reshape((4, 4)) print("Original array:") print(arra_data) print("\nExtracted data: Third and fourth elements of the first and second rows ") print(arra_data[0:2, 2:4])</pre>

PANDAS PROGRAMS –Unit 7

26	Write a Pandas program to sort a Series.
27	<p>Write a Pandas program to select the specified columns and rows from a given DataFrame.</p> <p>Select 'name' and 'score' columns in rows 1, 3, 5, 6 from the following data frame.</p> <p>Sample DataFrame:</p> <pre>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']</pre>
28	<p>Write a Pandas program to select the rows where the score is missing, i.e. is NaN.</p> <p>Sample DataFrame:</p> <p>Sample Python dictionary data and list labels:</p> <pre>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']</pre>
29	<p>Write a Pandas program to sort the data frame first by 'name' in descending order, then by 'score' in ascending order.</p> <p>Sample DataFrame:</p> <pre>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']</pre> <p>Values for each column will be:</p> <p>name : 'Suresh', score: 15.5, attempts: 1, qualify: 'yes', label: 'k'</p>
30	<p>Write a Pandas program to find and replace the missing values in a given DataFrame which do not have any valuable information.</p> <p>Example:</p> <p>Missing values: ?, --</p> <p>Replace those values with NaN</p>

26	<pre> import pandas as pd s = pd.Series(['100', '200', 'python', '300.12', '400']) print("Original Data Series:") print(s) new_s = pd.Series(s).sort_values() print(new_s) </pre>
27	<pre> 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]]) </pre>
28	<pre> 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("Rows where score is missing:") print(df[df['score'].isnull()]) </pre>
29	<pre> 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.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) </pre>
30	<pre> import pandas as pd import numpy as np pd.set_option('display.max_rows', None) #pd.set_option('display.max_columns', None) </pre>

```

df = pd.DataFrame({
'ord_no':[70001,np.nan,70002,70004,np.nan,70005,"--",
70010,70003,70012,np.nan,70013],
'purch_amt':[150.5,270.65,65.26,110.5,948.5,2400.6,5760,"?",12.43,2480.4,250.45,
3045.6],
'ord_date': ['?', '2012-09-10', np.nan, '2012-08-17', '2012-09-10', '2012-07-27', '2012-09-10', '2012-10-10', '2012-10-10', '2012-06-27', '2012-08-17', '2012-04-25'],
'customer_id':[3002,3001,3001,3003,3002,3001,3001,3004,"--",3002,3001,3001],
'salesman_id':[5002,5003,"?",5001,np.nan,5002,5001,"?",5003,5002,5003,"--"]})
print("Original Orders DataFrame:")
print(df)
print("\nReplace the missing values with NaN:")
result = df.replace({ "?": np.nan, "--": np.nan })
print(result)

```

Gui pgms -- UNIT 9

31) Python Code for Password validation with Name and Password stored in a Dictionary

```

import tkinter as tk
from tkinter import messagebox as m
d1={"Arun":"hello","Anu":"2012"}

def click():
    name=tx1.get()
    pw=tx2.get()
    if((name, pw) in d1.items()):
        m.showinfo("Valid",pw)
    else:
        m.showinfo("Invalid",)

w1=tk.Tk()
l1=tk.Label(w1,text="Name")
l1.pack()
l2=tk.Label(w1,text="password")
l2.pack()
tx1=tk.Entry(w1,width=10)
tx1.pack()
tx2=tk.Entry(w1,width=10,show="*")
tx2.pack()
b1=tk.Button(w1,text="validate",command=click)
b1.pack()
l3=tk.Label(w1)
w1.mainloop()

```

32)Python GUI code for getting Password with Entry Widget

```
import tkinter as tk

from tkinter import messagebox as m

def show():

    str=t1.get()

    m.showinfo("password",str)

w1=tk.Tk()

w1.title("First")

l1=tk.Label(w1,text="Password")

l1.grid(column=0,row=0)

t1=tk.Entry(w1,width=10,show="*")

t1.grid(column=1,row=0)

b1=tk.Button(w1,text="Show",command=show)

b1.grid(column=1,row=1)

w1.mainloop()
```

33)Python Code for Checkbutton Widget

```
import tkinter as tk

from tkinter import messagebox as m

def click():

    v=cv1.get()

    if v==1:

        m.showinfo("5thsem","checked")

window = tk.Tk()

cv1 = tk.IntVar()

cv2 = tk.IntVar()

cb1 = tk.Checkbutton(window, text = "5th sem", variable = cv1, \

    onvalue = 1, offvalue = 0, height=5, \
```

```

        width = 20,command=click)

cb2 = tk.Checkbutton(window, text = "3rd sem", variable = cv2, \
        onvalue = 1, offvalue = 0, height=5, \
        width = 20)

cb1.pack()
cb2.pack()

window.mainloop()

```

34)Python Code for Radiobutton widget

```

import tkinter as tk

def sel():
    #ss=str(var.get())

    selection = "You selected the option " + str(var.get())
    l1.config(text = selection)

window = tk.Tk()
var = tk.IntVar()
R1 = tk.Radiobutton(window, text="Option 1", variable=var, value=1,command=sel)
R1.pack()

R2 = tk.Radiobutton(window, text="Option 2", variable=var, value=2, command=sel)
R2.pack()

R3 = tk.Radiobutton(window, text="Option 3", variable=var, value=3, command=sel)
R3.pack()

l1 = tk.Label(window)
l1.pack()

window.mainloop()

```

35) Python Code for Listbox widget

```

import tkinter as tk

```

```

from tkinter import *

def items_selected(event):
    selected_indices = lb1.curselection()

    # lb1.curselection gives the indices of the selected items(by the cursor)
    print(selected_indices)
    print(lb1.get(selected_indices))
    #lb1.get(index)-> name of the selected item

w1=tk.Tk()
w1.geometry('500x600+50+50')
lb1=tk.Listbox(w1,)
lb1.insert(1,"Python")
lb1.insert(2,"C++")
lb1.insert(3,"C")
lb1.insert(4,"VC++")
lb1.bind('<<ListboxSelect>>', items_selected)

# List box selection is bound with the function mentioned. The function is invoked
when an item is selected

lb1.pack()
w1.mainloop()

w1,
listvariable=var,
height=6,
selectmode='extended')

#if selectmode='extended' makes to do multiple selection

lb1.pack()
w1.mainloop()

```

String Manipulation -- UNIT 4

36] Write a Python program to count the number of characters (character frequency) in a string.

37] Write a program with a user defined function to count the number of times a character (passed as argument) occurs in the given string.

38] Write a program to input a string from the user and print it in the reverse order without creating a new string.

39] Write a program with a user defined function with string as a parameter which replaces all vowels in the string with '*'.

40] Python Program to Slice a given string.

=====

36] Write a Python program to count the number of characters (character frequency) in a string.

```
def char_frequency(str1):  
    dict = {}  
    for n in str1:  
        keys = dict.keys()  
        if n in keys:  
            dict[n] += 1  
        else:  
            dict[n] = 1  
    return dict  
print(char_frequency('Nhce is the best'))
```

37] Write a program with a user defined function to count the number of times a character (passed as argument) occurs in the given string.

```
def charCount(ch,st):  
    count = 0  
    for character in st:
```



```

        if character == ch:
            count += 1
    return count
#end of function
st = input("Enter a string: ")
ch = input("Enter the character to be searched: ")
count = charCount(ch,st)
print("Number of times character",ch,"occurs in the string is:",count)

```

38] Write a program to input a string from the user and print it in the reverse order without creating a new string.

```

st = input("Enter a string: ")
for i in range(-1,-len(st)-1,-1):
    print(st[i],end="")

```

39] Write a program with a user defined function with string as a parameter which replaces all vowels in the string with '*'.

```

def replaceVowel(st):
    #create an empty string
    newstr = ""
    for character in st:
        #check if next character is a vowel
        if character in 'aeiouAEIOU':
            #Replace vowel with *
            newstr += '*'
        else:
            newstr += character
    return newstr
#end of function

```

```
st = input("Enter a String: ")
st1 = replaceVowel(st)
print("The original String is:",st)
print("The modified String is:",st1)
```

40] Python Program to Slice a given string.

```
x = 'Hello Nhce and CSE'
```

```
# Slicing the String using two indexes
```

```
a = x[2:13]
print("Both Indexes = ", a)
```

```
# Slicing the String using Second indexes
```

```
b = x[:8]
print("No First Index = ", b)
```

```
# Slicing the String using First indexes
```

```
c = x[4:]
print("No Second Index = ", c)
```

```
# Slicing the String without using two indexes
```

```
d = x[:]
print("No Indexes = ", d)
```

```
# Slicing the String using Negative indexes
```

```
e = x[-3:]
print("Negative First Index = ", e)
```

OOPS In Python:-- UNIT 8

41] What is Multiple Inheritance with example.

42] Write a Python class to find a pair of elements (indices of the two numbers) from a given array whose sum equals a specific target number.

43] Write a Python class named Rectangle constructed by a length and width and a method which will compute the area of a rectangle.

44] What is Encapsulation with example.

45] What is Polymorphism with example.

Solutions:

41] class Calculation1:

```
    def Summation(self,a,b):  
        return a+b;
```

class Calculation2:

```
    def Multiplication(self,a,b):  
        return a*b;
```

class Derived(Calculation1,Calculation2):

```
    def Divide(self,a,b):  
        return a/b;
```

d = Derived()

print(d.Summation(10,20))

print(d.Multiplication(10,20))

print(d.Divide(10,20))

42]

class py_solution:

```
    def twoSum(self, nums, target):  
        lookup = {}  
        for i, num in enumerate(nums):  
            if target - num in lookup:  
                return (lookup[target - num], i )  
            lookup[num] = i
```

print("index1=%d, index2=%d" % py_solution().twoSum((10,20,10,40,50,60,70),50))

```
43] class Rectangle():
    def __init__(self, l, w):
        self.length = l
        self.width = w

    def rectangle_area(self):
        return self.length*self.width

newRectangle = Rectangle(12, 10)
print(newRectangle.rectangle_area())
```

```
44] class Person:
    def __init__(self, name, age=0):
        self.name = name
        self._age = age

    def display(self):
        print(self.name)
        print(self._age)
```

```
person = Person('Dev', 30)
#accessing using class method
person.display()
#accessing directly from outside
print(person.name)
print(person._age)
```

```
45] class Cat:
    def __init__(self, name, age):
```

```
self.name = name
```

```
self.age = age
```

```
def info(self):
```

```
    print(f"I am a cat. My name is {self.name}. I am {self.age} years old.")
```

```
def make_sound(self):
```

```
    print("Meow")
```

```
class Dog:
```

```
    def __init__(self, name, age):
```

```
        self.name = name
```

```
        self.age = age
```

```
    def info(self):
```

```
        print(f"I am a dog. My name is {self.name}. I am {self.age} years old.")
```

```
    def make_sound(self):
```

```
        print("Bark")
```

```
cat1 = Cat("Kitty", 2.5)
```

```
dog1 = Dog("Fluffy", 4)
```

```
for animal in (cat1, dog1):
```

```
    animal.make_sound()
```

```
    animal.info()
```

```
    animal.make_sound()
```

