Lab 1

```
# 1.1   1D array

import numpy as np
arr=np.array([1,2,3,4,5,6,7,8,9,10])
print(arr)
print(np.ndim(arr))
print(type(arr))
```

```
[ 1  2  3  4  5  6  7  8  9 10]
1
<class 'numpy.ndarray'>
```

```
# 1.2

arr = np.random.random((3,4))            #creates random array of 3 rows with 4 values e
print("The array is:\n", arr)
print(arr.shape)
```

```
The array is:
 [[0.88411511 0.09488158 0.98653586 0.1619359 ]
 [0.67650439 0.62287998 0.94764635 0.70619668]
 [0.43974987 0.50930703 0.82103618 0.54749712]]
(3, 4)
```

```
# 1.3

new_arr=arr.reshape(4,3) #reshaping arr array to (4,3)
print(new_arr)
```

```
[[0.88411511 0.09488158 0.98653586]
 [0.1619359  0.67650439 0.62287998]
 [0.94764635 0.70619668 0.43974987]
 [0.50930703 0.82103618 0.54749712]]
```

```
# 1.4

spec_arr=np.linspace(1,10,num=20)
print(spec_arr)
```

```
[ 1.          1.47368421  1.94736842  2.42105263  2.89473684  3.36842105
  3.84210526  4.31578947  4.78947368  5.26315789  5.73684211  6.21052632
  6.68421053  7.15789474  7.63157895  8.10526316  8.57894737  9.05263158
  9.52631579 10.        ]
```

```
# 1.5

a= np.array([1,2,3,4,5])
print(a)

b=np.flip(a,axis=0)
print(b)
```

```
[1 2 3 4 5]
[5 4 3 2 1]
```

LAB 2

```
# 2.1

a=np.array([1,2,3,4])
b=np.array([10,7,8,5])

c=np.add(a,b)
print(c)

sub=np.subtract(b,a)
print(sub)

product=np.multiply(a,b)
print(product)

div=np.divide(a,b)
print(div)
```

```
[11  9 11  9]
[9 5 5 1]
[10 14 24 20]
[0.1        0.28571429 0.375      0.8        ]
```

```
# 2.2

arr=np.array([1,2,3])

a_sum=np.sum(arr)
print("Sum:",a_sum)

a_mean=np.mean(arr)
print("Mean:",a_mean)

a_std=np.std(arr)
print("Standard deviation:",a_std)
```

```
Sum: 6
Mean: 2.0
Standard deviation: 0.816496580927726
```

```
# 2.3

arr=np.array([1,2,3])

max=np.max(arr)
print("Max:",max)

min=np.min(arr)
print("Min:",min)
```

```
Max:  3
Min:  1
```

```
# 2.4

a=5
b=np.array([1,2,3])

print(a+b)
```

```
[6 7 8]
```

```
# 2.5

a=np.array([[1,2],[3,4]])
b=np.array([[5,6],[7,8]])

print(np.dot(a,b))
```

```
[[19 22]
 [43 50]]
```

LAB 3

```
# 3.1

arr=np.array([[1,2],[3,4]])
print(arr[0,1])
```

```
2
```

```
#3.2

arr=np.array([1,2,3,4,5])
a=arr[2:4]
print(a)
```

```
[3 4]
```

```
# 3.3

arr=np.array([[1,2,3],[4,5,6]])

print(arr[:,1]) # prints the second column
print(arr[1,:]) #prints the second row
```

```
[2 5]
[4 5 6]
```

```
# 3.4

arr=np.array([[[1,2,3]]])
```

```
# 3.5

arr=np.array([1,2,3,4,5])
print(arr)

arr[2:4]=(7,8)   #Updating the values of the slice
print(arr)
```

```
[1 2 3 4 5]
[3 4]
[1 2 7 8 5]
```

Lab 4

```
# 4.1

arr1=np.array([1,2,3])  #1D array
arr2=np.array([[4,5,6],[7,8,9]])  # 2D array

print(arr1+arr2) #Broadcasting by adding 1D array to 2D array
```

```
[[ 5  7  9]
 [ 8 10 12]]
```

```
# 4.2

arr=np.array([[4,5,6],[7,8,9]])
b=3

print(np.multiply(arr,b))
```

```
     [[12 15 18]
      [21 24 27]]
```

```
# 4.3

a=np.arange(1,10).reshape(3,3)
b= np.arange(10,40,10).reshape(3,1)
c = a + b
print(a,"Original 3x3 matrix\n")

print(b,"\nColumn vector\n")

print(c,"\nResult of broadcasting\n")
```

```
     [[1 2 3]
      [4 5 6]
      [7 8 9]] Original 3x3 matrix

     [[10]
      [20]
      [30]]
     Column vector

     [[11 12 13]
      [24 25 26]
      [37 38 39]]
     Result of broadcasting
```

```
# 4.4


arr1=np.array([1,1,1])  #1D array
arr2=np.array([[4,5,6],[7,8,9]])  # 2D array

print(arr2-arr1) #Broadcasting by subtracting 1D array from 2D array
```

```
     [[3 4 5]
      [6 7 8]]
```

```
# 4.5
arr1=np.array([1,2,3])  #1D array
arr2=np.array([[4,5,6],[7,8,9]])  # 2D array
```

```
arr2=np.array([[4,5,6],[7,8,9]])  # 2D array

print(np.multiply(arr1,arr2)) #element wise multipilcation can be done
```

```
[[ 4 10 18]
 [ 7 16 27]]
```