Experiment – 1

Basics of Python and Numpy

Aim:

1) Learn the procedure to Install Python, creating a Project in Pycharm and add packages

2) To code six examples for Python Programming

3) To code five Examples for using Numpy and Matplotlib Packages

Software/ Package Used:

1. Pycharm IDE with Python 3.7
2. Libraries used:
   a) NumPy
   b) Matplotlib

Procedure:
Install Python:

1. Download Python:
- Go to the [Python downloads page](https://www.python.org/downloads/).
- Choose the latest version of Python for Windows.
- Click on the download link and run the installer.

2. Run the Installer:
- Check the box that says "Add Python x.x to PATH" before installing. This option is essential for easily using Python from the command line.
- Click "Install Now" or customize the installation by clicking "Customize installation" if you need to modify certain components.

3. Verify Installation:
- Open Command Prompt and type `python --version` or `python` to ensure Python is installed and working. You should see the version number displayed.

Create a Project in Pycharm:

1. Open PyCharm: Launch the PyCharm IDE on your computer.

2. Create a New Project: Click on "File" > "New Project" or use the "Create New Project" option on the welcome screen.

3. Choose Interpreter: Select the Python interpreter you want to use for your project. If you don't have one, click on "New environment using" and choose a Python interpreter.

4.Set Project Location: Define the location where you want to save your project files.

5. Select Project Type: Choose the project type/template you want (e.g., pure Python, web development, data science).

6. Configure Project: Configure project-specific settings if needed, like project name, project interpreter, project structure, etc.

7. Create the Project: Click "Create" or "OK" to generate the project.

8. Start Coding: Once the project is created, you'll see the project structure on the left sidebar. You can start coding by creating Python files or adding existing ones to your project.

Add Packages:

1. Check if `pip` is installed: Open your command prompt or terminal and enter `pip --version` or `pip3 --version` (depending on your Python version). If it's installed, it will display the version number; if not, you'll need to install Python or `pip`.

2. Install a package: Use `pip install` followed by the package name. For example, to install a package called `requests`, you'd enter `pip install requests`.

3. Specific versions: You can specify a particular version by adding `==` followed by the version number. For example, `pip install requests==2.25.1`.

4. Install from a requirements file: If you have a list of packages to install saved in a file (usually named `requirements.txt`), you can install them all at once using `pip install -r requirements.txt`.

Programs:

1)PRINT AND ADDITION

CODE

```
#print hello world
print("Hello world!")
#declare variables
a = 15
b = 23
#add and store value
c = a+b
print(c)
```

OUTPUT

```
Hello world!
```

2) **COMPARISON**

CODE

```
x = int(input("input"))
y = int(input("input"))
z = int(input("input"))
if x>y:
   if x>z:
      print("x is greater")
   else:
      print("z is greater")
else:
   if y>z:
      print("y is greater")
   else:
      print("z is greater")
```

INPUT AND OUTPUT

```
input15
input26
input23
y is greater
```

3)LOOPS

CODE

```
a)for i in range(2,6,2):
   print(i , end ="")
i = 1


b)while i < 6:
  print(i)
  if i == 3:
    break
  i += 1


c)for i in range(5):
   for j in range(5,i,-1):
      print( "*",  end = "")
   print("")



d) count=0
while count<5:
  count+=1
  print ("Iteration no. {}".format(count))
else:
```

```
   print ("While loop over. Now in else block")
print ("End of while loop")
```

```
e) a = 33
b = 33
if b > a:
  print("b is greater than a")
elif a == b:
  print("a and b are equal")
```

```
f) x = 41
if x > 10:
  print("Above ten,")
  if x > 20:
    print("and also above 20!")
  else:
    print("but not above 20.")
```

INPUT AND OUTPUT

a)241

b)2
3

c)*****
****
***
**
*

d) Iteration no. 1
Iteration no. 2
Iteration no. 3
Iteration no. 4
Iteration no. 5
While loop over. Now in else block
End of while loop

e) a and b are equal

f) Above ten,
and also above 20!

4) **FUNCTIONS**

CODE

```
def aivison(name):
   print(name)
aivison("Shri")
aivison("Raghavi")
# function definition
def find_square(num):
   result = num * num
   return result
# function call
square = find_square(3)
print('Square:',square)
```

INPUT AND OUTPUT

```
Shri
Raghavi
Square: 9
```

5)NUMPY

CODE

```
import numpy as np

# Creating array object

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

# Printing type of arr object

print("Array is of type: ", type(arr))

# Printing array dimensions

print("No. of dimensions: ", arr.ndim)

# Printing shape of array

print("Shape of array: ", arr.shape)

# Printing size (total number of elements) of array

print("Size of array: ", arr.size)

# Printing type of elements in array

print("Array stores elements of type: ", arr.dtype)

# Create a sequence of integers from 0 to 30 with steps of 5

f = np.arange(0, 30, 5)

print ("A sequential array with steps of 5:\n", f)

# Create a sequence of 10 values in range 0 to 5
```

```
g = np.linspace(0, 5, 10)
print ("A sequential array with 10 values between"
                    "0 and 5:\n", g)
```

OUTPUT

```
Array is of type:  <class 'numpy.ndarray'>
No. of dimensions:  2
Shape of array:  (2, 3)
Size of array:  6
Array stores elements of type:  int32
A sequential array with steps of 5:
 [ 0  5 10 15 20 25]
A sequential array with 10 values between0 and 5:
 [0.         0.55555556 1.11111111 1.66666667 2.22222222 2.77777778
 3.33333333 3.88888889 4.44444444 5.        ]
```

6) MATPLOT LIB

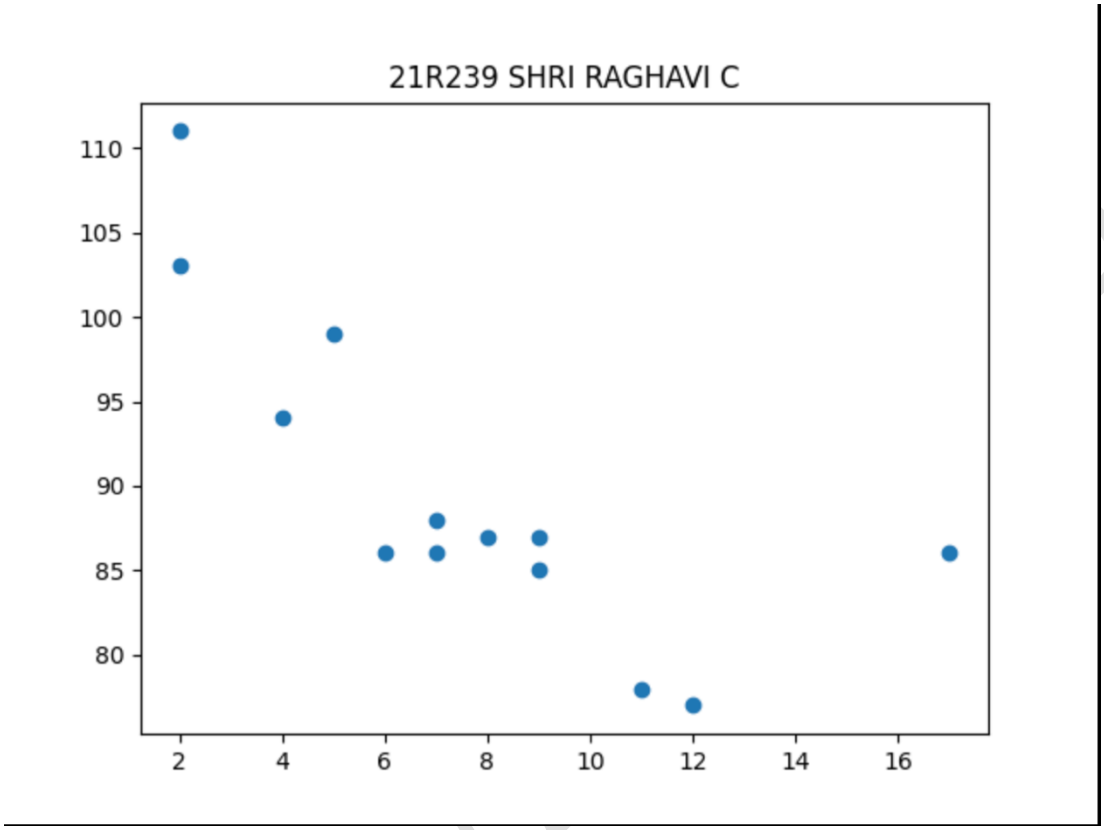
CODE

```
#Three lines to make our compiler able to draw:
import sys
import matplotlib
matplotlib.use('Agg')
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.title("21R239 SHRI RAGHAVI C")
plt.scatter(x, y)
plt.show()
#Two  lines to make our compiler able to draw:
```

plt.savefig(sys.stdout.buffer)

sys.stdout.flush()

OUTPUT



Department of RAE			
Criteria	**Excellent (75% - 100%)**	**Good (50 – 75%)**	**Poor (<50%)**
Preparation (30)			
Performance (30)			
Evaluation (20)			
Report (20)			
Sign:		**Total (100)**	

Result:

The basics of python, installation of packages and usage of packages Matplotlib and Numpy was demonstrated and successfully executed