CHAROTAR UNIVERSITY OF SCIENCE & TECHNOLOGY

DEVANG PATEL INSTITUTE OF ADVANCE TECHNOLOGY & RESEARCH

Department of Computer Engineering

Practical – 1

1.1:

AIM: Introduction to Python Programming. Installation & Configuration of Python. Along with its all-major editors, IDLE, Pycharm, Anaconda, Jupyter, Interpreter etc.

Tools and Technologies used:

1. IDLE Editor:

- IDLE is an integrated development environment for Python, which has been bundled with the default implementation of the language.

2. Pycharm:

- The PyCharm editor is the main part of the IDE that you use to create, read and modify code. The editor consists of the following areas: The scrollbar shows errors and warnings in the current file.

3. Anaconda:

- Anaconda is a distribution of the Python and R programming languages for scientific computing (data science, machine learning applications etc.), that aims to simplify package management and deployment.

4. Jupyter:

- "Jupyter" is a loose acronym meaning Julia, Python, and R. These programming languages were the first target languages of the Jupyter application.

5. Interpreter:

- Python is an interpreted language, which means the source code of a Python program is converted into bytecode that is then executed by the Python virtual machine.

Python Programming:

- Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured, object-oriented and functional programming.

Python Installation:

Step 1: Download the Python Installer binaries

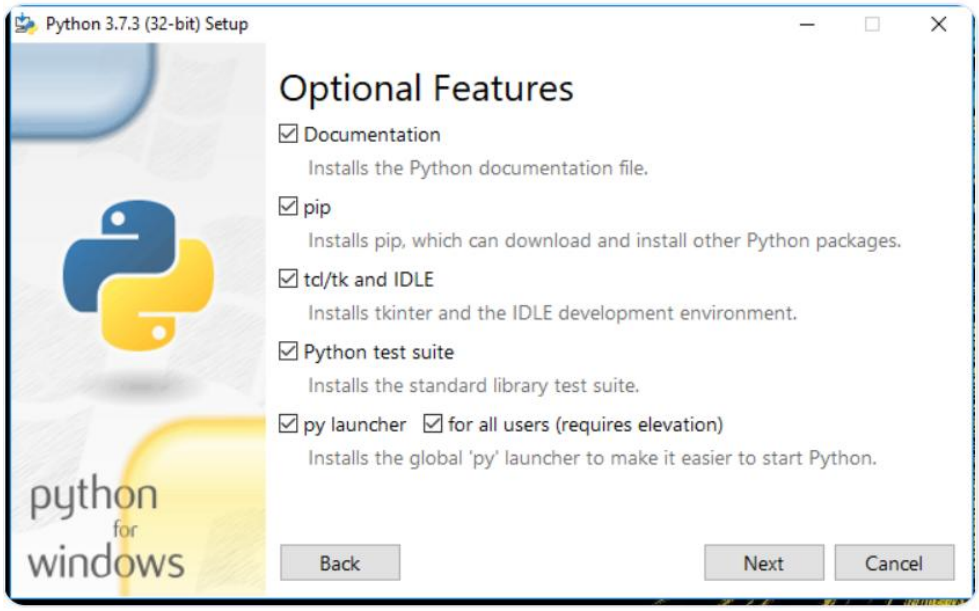
- Open the official Python website www.python.org in your web browser. Navigate to the Downloads tab for Windows.
- Choose the latest Python 3 release. In our example, we choose the latest Python 3.6.2 version.
- Click on the link to download **Windows x86 executable installer** if you are using a 32-bit installer. In case your Windows installation is a 64-bit system, then download **Windows x86-64 executable installer**.

Step 2: Run the Executable Installer

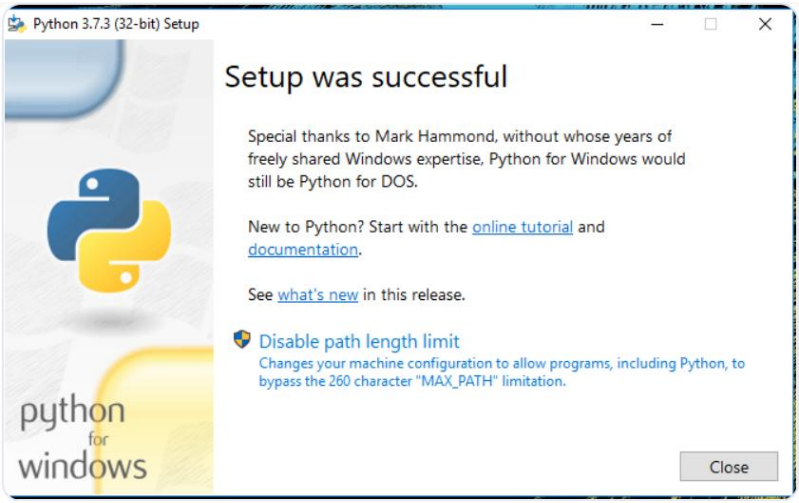
- Once the installer is downloaded, run the Python installer.
- Check the **Install launcher for all users** check box. Further, you may check the **Add Python 3.7 to path** check box to include the interpreter in the execution path.

- Select **Customize installation**. Choose the optional features by checking the following check boxes:
- Documentation
- pip

- tcl/tk and IDLE (to install tkinter and IDLE)
- Python test suite (to install the standard library test suite of Python)
- Install the global launcher for `.py` files. This makes it easier to start Python
- Install for all users.



- Once the installation is over, you will see a **Python Setup Successful** window.

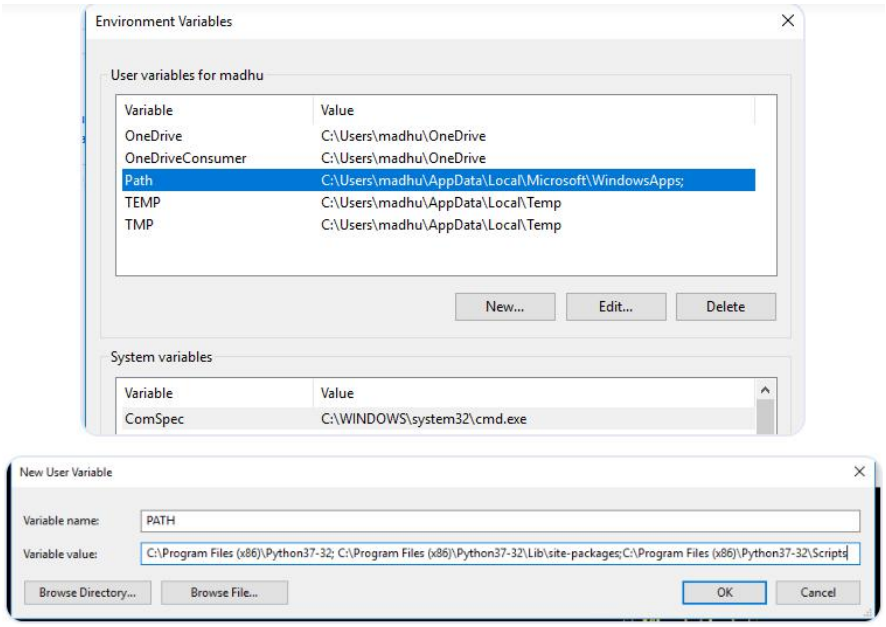


Step 3: Add Python to environmental variables

- The last (optional) step in the installation process is to add Python Path to the System Environment variables. This step is done to access Python through the command line. In case you have added Python to environment variables while setting the Advanced options during the installation procedure, you can avoid this step. Else, this step is done manually as follows. In the Start menu, search for “advanced system settings”. Select “View advanced system settings”. In the “System Properties” window, click on the “Advanced” tab and then click on the “Environment Variables” button. Locate the Python installation directory on your system. If you followed the steps exactly as above, python will be installed in below locations:

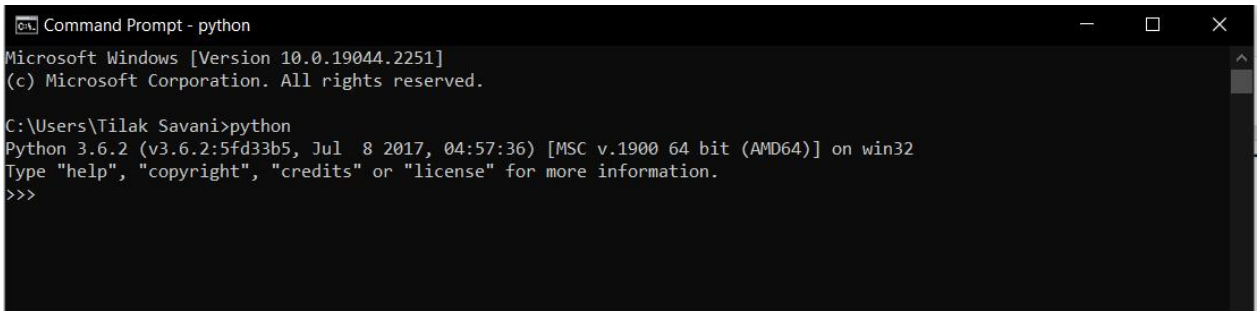
- C:\Program Files (x86)\Python37-32: for 32-bit installation
- C:\Program Files\Python37-32: for 64-bit installation

➢ The folder name may be different from "Python37-32" if you installed a different version. Look for a folder whose name starts with Python. Append the following entries to PATH variable as shown below:



Step 4: Verify the Python Installation

➢ You have now successfully installed Python 3.6.2 on Windows 10. You can verify if the Python installation is successful either through the command line or through the IDLE app that gets installed along with the installation. Search for the command prompt and type "python". You can see that Python 3.6.2 is successfully installed.



Learning Outcomes:

In this practical, following concepts were learnt:

i. Installation Process and check the version of Python in System
ii. Set the environment path of Python
iii. Knowledge about the components used by Python

1.2:

AIM: Write a python program to calculate simple interest.

Tools and Technologies used:

- VS Code
  - Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft with the Electron Framework, for Windows, Linux and macOS.
- Python Script
  - A Python script, the file containing the commands, is structured to be executed like a program. These files are designed to contain various functions and import various modules.

Program Code:

```
p = int(input("Enter the value of p: "))
r = float(input("Enter the value of r: "))
n = int(input("Enter the valur of n: "))

ans = p*r*n/100
print('\n')
print("Simple Intrest is: ", ans)
```

Output:

```
PS D:\Semester-4\Python\Python Practicals> python -u "d:\Semester-4\Python\Python Practicals\Simpleintrest.py"
Enter the value of p: 500
Enter the value of r: 1
Enter the valur of n: 5


Simple Intrest is:  25.0
PS D:\Semester-4\Python\Python Practicals>
```

Learning Outcomes:

In this practical, following concepts were learnt:

i. How to declare the variables and assign the value and how to print the value of variable and typecasting of input variable
ii. How to go inside the new line and print the value of variable after the details

Practical – 2

2.1:

AIM: Create a list and apply methods (append, extend, remove, reverse), arrange created list in ascending and descending order.

Tools and Technologies used:

- VS Code
  - Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft with the Electron Framework, for Windows, Linux and macOS.

- Python Script
  - A Python script, the file containing the commands, is structured to be executed like a program. These files are designed to contain various functions and import various modules.

Program Code:

```
list1 = [10, 20, 30, 40, 50]
print("list1:", list1)

list1.append(80)
print("append:", list1)

list2 = [60, 70, 90, 100]
print("list2:", list2)

list1.extend(list2)
print("Extended list1:", list1)

list1.remove(80)
print("remove of 80:", list1)

list1.reverse()
print("reverse:", list1)

list1.sort()
print("asending order:", list1)

list1.sort(reverse = True)
print("desending order:", list1)
```

Output:

```
PS D:\Semester-4\Python\Python Practicals> python -u "d:\Semester-4\Python
list1: [10, 20, 30, 40, 50]
append: [10, 20, 30, 40, 50, 80]
list2: [60, 70, 90, 100]
Extended list1: [10, 20, 30, 40, 50, 80, 60, 70, 90, 100]
remove of 80: [10, 20, 30, 40, 50, 60, 70, 90, 100]
reverse: [100, 90, 70, 60, 50, 40, 30, 20, 10]
asending order: [10, 20, 30, 40, 50, 60, 70, 90, 100]
desending order: [100, 90, 70, 60, 50, 40, 30, 20, 10]
PS D:\Semester-4\Python\Python Practicals>
```

Learning Outcomes:

In this practical, following concepts were learnt:

i. How to Create the list and basic concepts
ii. Create, Append, Extend, Remove Opration on list
iii. Ascending and Descending Order Opration using sort and reverse equal to True keyword

2.2:

AIM: List1 = [1, 2, 3, 4, ["python", "java", "c++", [10,20,30]], 5, 6, 7, ["apple", "banana", "orange"]] From above list get word “orange” and “Python” & repeat this list five times without using loops.

Tools and Technologies used:

- VS Code
  - Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft with the Electron Framework, for Windows, Linux and macOS.

- Python Script
  - A Python script, the file containing the commands, is structured to be executed like a program. These files are designed to contain various functions and import various modules.

Program Code:

```
List1 = [1, 2, 3, 4, ["python", "java", "c++", [10,20,30]], 5, 6, 7, ["apple", "banana", "orange"]]
print(List1)

print("Orange Keyword:",List1[8][2])

print("Python Keyword:",List1[4][0])

print("20 value:",List1[4][3][1])

print("Print the List 5 times:\n",List1 * 5)
```

Output:

```
PS D:\Semester-4\Python\Python Practicals> python -u "d:\Semester-4\Python\Python Practicals\prac2_2.py"
[1, 2, 3, 4, ['python', 'java', 'c++', [10, 20, 30]], 5, 6, 7, ['apple', 'banana', 'orange']]
Orange Keyword: orange
Python Keyword: python
20 value: 20
Print the List 5 times:
 [1, 2, 3, 4, ['python', 'java', 'c++', [10, 20, 30]], 5, 6, 7, ['apple', 'banana', 'orange'], 1, 2, 3, 4, ['python', 'java', 'c++', [10, 20, 30]],
 5, 6, 7, ['apple', 'banana', 'orange'], 1, 2, 3, 4, ['python', 'java', 'c++', [10, 20, 30]], 5, 6, 7, ['apple', 'banana', 'orange'], 1, 2, 3, 4, [
'python', 'java', 'c++', [10, 20, 30]], 5, 6, 7, ['apple', 'banana', 'orange'], 1, 2, 3, 4, ['python', 'java', 'c++', [10, 20, 30]], 5, 6, 7, ['app
le', 'banana', 'orange']]
PS D:\Semester-4\Python\Python Practicals>
```

Learning Outcomes:

In this practical, following concepts were learnt:

i. find the element inside the list using list and array index in 2D and 3D or many
ii. print 5 times list using list * n(times)

2.3:

AIM: Create a list and copy it using slice function.

Tools and Technologies used:

- VS Code
  - Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft with the Electron Framework, for Windows, Linux and macOS.

- Python Script
  - A Python script, the file containing the commands, is structured to be executed like a program. These files are designed to contain various functions and import various modules.

Program Code:

```
list1 = [27, 13, -11, 60, 34, 15]

list2 = slice(0, 5)
print("Using Slice:",list1[list2])

list3 = slice(2)
print("Using Slice:",list1[list3])'


list_copy = list1[slice(len(list1))]
print("Orignal list:", list1)

print("Copy list:", list_copy)
```

Output:

```
PS D:\Semester-4\Python\Python Practicals> python -u "d:\Semester-4\Python\Python Practicals\prac2_3.py"
Using Slice: [27, 13, -11, 60, 34]
Using Slice: [27, 13]
Orignal list: [27, 13, -11, 60, 34, 15]
Copy list: [27, 13, -11, 60, 34, 15]
PS D:\Semester-4\Python\Python Practicals>
```

Learning Outcomes:

In this practical, following concepts were learnt:

i. Create the list and copy element inside the another list using slice
ii. Also learn new function of find length

2.4:

AIM: Create a tuple and apply different type of mathematical operation on it (Sum, Maximum, minimum etc.)

Tools and Technologies used:

- VS Code
  - Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft with the Electron Framework, for Windows, Linux and macOS.

- Python Script
  - A Python script, the file containing the commands, is structured to be executed like a program. These files are designed to contain various functions and import various modules.

Program Code:

```
t1 = (4, 5, 1, 2)

t2 = (7, 9, 8)

t3 = t1 + t2
print("Contactination of t1 and t2 is:", t3)

print("Sum of t3 element is:",sum(t3))

x = min(t3)

print("Minmum element of t3 is:",x)

y = max(t3)

print("Maximum element of t3 is:",y)
```

Output:

```
PS D:\Semester-4\Python\Python Practicals> python -u "d:\Semester-4\Python\Python Practicals\prac2_4.py"
Contactination of t1 and t2 is: (4, 5, 1, 2, 7, 9, 8)
Sum of t3 element is: 36
Minmum element of t3 is: 1
Maximum element of t3 is: 9
PS D:\Semester-4\Python\Python Practicals>
```

Learning Outcomes:

In this practical, following concepts were learnt:

i. Learn how to create the tuples and contactination of 2 tuples using + operator or sign
ii. Sum of all number are inside the tuples using sum function
iii. And find the minimum and maximum element of tuples using min and max function

Practical – 3

3.1

Aim: String Operations:

- Reverse a string, replace string with other string, merge two strings.
- Find character is in string or not without using loops.
- Split string into multiple words.

Tools and Technologies used:

- Python IDLE
- Python Script

Program code:

```
txt1 = "DEPSTAR is my collage"

print(txt1)

a = txt1[::-1]

print(a)

txt2 = txt1.replace("DEPSTAR","CSPIT")

print(txt2)

txt3 = txt1.replace("E","S")

print(txt3)

txt4=txt1 + " " + txt2
```

```
print(txt4)

txt5=txt4.index("STAR")

print(txt5)

txt6=txt4.split(" ")

print(txt6)

print("D22DCE190")
```

Output:

```
PS C:\Users\admin> python -u "g:\DSA WITH C\Untitled-1.py"
DEPSTAR is my collage
egalloc ym si RATSPED
CSPIT is my collage
DSPSTAR is my collage
DEPSTAR is my collage CSPIT is my collage
3
['DEPSTAR', 'is', 'my', 'collage', 'CSPIT', 'is', 'my', 'collage']
D22DCE190
```

3.2

Aim: Dictionaries Operations:

• Apply “Update, Delete, clear, pop item, pop, get, keys and values” operation in dictionary.
• Create 3 dictionaries and merge them into 1 dictionary.

Tools and Technologies used:

• Python IDLE
• Python Script

Program code:

```
d = {"collage":"DEPSTAR",

"DEPT":"CE",

"ID":"D22DCE172"}

print(d)

a = d.update({"ID" : "D22DCE190"})

print(a)

print(d)

key = d.keys()

print(key)

value = d.values()

print(value)

pop = d.pop("ID")

print(pop)

d1 = {"Hostel" : "Ashirwad",

"Room" : 214}

d2 = {"City" : "Changa",

"Pin" : 388421}

d3 = {**d,**d1,**d2}
```

```
print(d3)

del d1['Hostel']

print(d1)

d1.clear()

print(d1)

print("D22DCE190")
```

Output:

```
PS C:\Users\admin> python -u "g:\DSA WITH C\Untitled-1.py"
{'collage': 'DEPSTAR', 'DEPT': 'CE', 'ID': 'D22DCE172'}
None
{'collage': 'DEPSTAR', 'DEPT': 'CE', 'ID': 'D22DCE190'}
dict_keys(['collage', 'DEPT', 'ID'])
dict_values(['DEPSTAR', 'CE', 'D22DCE190'])
D22DCE190
{'collage': 'DEPSTAR', 'DEPT': 'CE', 'Hostel': 'Ashirwad', 'Room': 214, 'City': 'Changa', 'Pin': 388421}
{'Room': 214}
{}
D22DCE190
```

Conclusion:

In this practical, following concepts were learnt:

I. Various string operations

II. Various dictionary operations

Practical – 4

4.1:

AIM:

These all programs should be done by declaring a function. Found which grade student will get based on SGPA.

Tools and Technologies used:

- Python IDLE:

  IDLE is an integrated development environment for Python. It is packaged as an optional part of the Python packaging with many Linux distributions.

- Python Script:

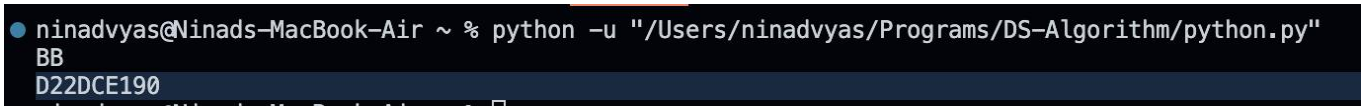
  A Python script, the file containing the commands, is structured to be executed like a program. These files are designed to contain various functions and import various modules.

Program Code:

```
def determine_grade(sgpa):
    if sgpa >= 9.0:
        return "AA"
    elif sgpa >= 8.0:
        return "AB"
    elif sgpa >= 7.0:
        return "BB"
    elif sgpa >= 6.0:
        return "BC"
    elif sgpa >= 5.0:
        return "CC"
    elif sgpa >= 4.0:
        return "CD"
    else:
        return "FF"

list = [90, 80, 50, 70, 60]
sum = 0
for i in list:
    sum = sum + i

sgpa = (sum/ len(list))/10
```

```
print(determine_grade(sgpa))
print("ID : D22DCE190")
```

Output:



Learning Outcomes:

In this practical we have learnt about :

i. How to use for loop in practical
ii. How to make a function and how to call

4.2:

AIM:

Find max from three numbers.

Tools and Technologies used:

- Python IDLE:

  IDLE is an integrated development environment for Python. It is packaged as an optional part of the Python packaging with many Linux distributions.

- Python Script:

  A Python script, the file containing the commands, is structured to be executed like a program. These files are designed to contain various functions and import various modules.

Program Code:

```
def maximum(a,b,c) :
    if(a>b and a>c):
     largest = a
    elif(b>a and b>c):
     largest = b
    else:
     largest = c
    return largest
#Taking input from user
a = int(input("Enter num1 :"))
b = int(input("Enter num2 :"))
c = int(input("Enter num3 :"))
#Calling a maximum  function
print("Largest number :",maximum(a,b,c))
print("Id No.: D22DCE190")
```

Output:

Learning Outcomes:

In this practical, we have learnt about :

i. How to create a function and how to use if else
ii. How to print the output in different ways

4.3:

AIM:

Calculate number of Uppercase and lowercase letters of string given by user.

Tools and Technologies used:

- Python IDLE:

  IDLE is an integrated development environment for Python. It is packaged as an optional part of the Python packaging with many Linux distributions.

- Python Script:

  A Python script, the file containing the commands, is structured to be executed like a program. These files are designed to contain various functions and import various modules.

Program Code:

```
def string_test(s):
   d={"UPPER_CASE":0, "LOWER_CASE":0}
   for c in s:
      if c.isupper():
        d["UPPER_CASE"]+=1
      elif c.islower():
        d["LOWER_CASE"]+=1
      else:
        pass
   print ("Original String : ", s)
   print ("No. of Upper case characters : ", d["UPPER_CASE"])
   print ("No. of Lower case Characters : ", d["LOWER_CASE"])
str = input("Enter string :")
string_test(str)
print("Id No.: D22DCE190")
```

Output:

Learning Outcomes:

In this practical, we have learnt about :

i. Different functions like isupper, islower
ii. About the if else and last of else you can only pass through it

4.4:

AIM:

Find a Square of a given list using lambda function.

Tools and Technologies used:

- Python IDLE:

  IDLE is an integrated development environment for Python. It is packaged as an optional part of the Python packaging with many Linux distributions.

- Python Script:

  A Python script, the file containing the commands, is structured to be executed like a program. These files are designed to contain various functions and import various modules.

Program Code:

```
num = [1, 2, 3, 4, 5]
print("Original List:",num)
s_num = list(map(lambda x: x ** 2, num))
print(s_num)
print("Id No.: D22DCE190")
```

Output:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

python -u "/Users/ninadvyas/Programs/DS-Algorithm/python.py"
ninadvyas@Ninads-MacBook-Air ~ % python -u "/Users/ninadvyas/Programs/DS-Algorithm/python.py"
Original List: [1, 2, 3, 4, 5]
[1, 4, 9, 16, 25]
Id No.: D22DCE190
```

Learning Outcomes:

In this practical, we have learnt about :

i. The lambda and use of it and learn the map
ii. It is perform any operations and print the result

4.5:

AIM:

Enter value from user and print multiplication table.

Tools and Technologies used:

- Python IDLE:
  IDLE is an integrated development environment for Python. It is packaged as an optional part of the Python packaging with many Linux distributions.
- Python Script:
  A Python script, the file containing the commands, is structured to be executed like a program. These files are designed to contain various functions and import various modules.

Program Code:

```
def print_table(num):
    for i in range(1,11):
        print(num,' x ', i, ' = ',num*i)
n = int(input("Enter a number:"))
print("Multiplication table:")
print_table(n)
print("Id No.: D22DCE190")
```

Output:

```
ninadvyas@Ninads-MacBook-Air ~ % python -u "/Users/ninadvyas/Programs/DS-Algorithm/python.py"
Enter a number:90
Multiplication table:
90  x  1  =  90
90  x  2  =  180
90  x  3  =  270
90  x  4  =  360
90  x  5  =  450
90  x  6  =  540
90  x  7  =  630
90  x  8  =  720
90  x  9  =  810
90  x  10  =  900
Id No.: D22DCE190
```

Learning Outcomes:

In this practical, we have learnt about :

i. The range and it's uses to print table
ii. Print the table with perfect manner

4.6:

AIM:

Create a list by user given value and make sum of it using loop.

Tools and Technologies used:

- Python IDLE:

  IDLE is an integrated development environment for Python. It is packaged as an optional part of the Python packaging with many Linux distributions.

- Python Script:

  A Python script, the file containing the commands, is structured to be executed like a program. These files are designed to contain various functions and import various modules.

Program Code:

```
list = []
n = int(input("Enter no of element:"))
for i in range(0,n):
    ele = int(input())
    list.append(ele)
print(list)
count = 0
for i in list:
    count+= i;
avg = count/len(list)
#avg = count/n
print("Sum of list element :",count)
print("Average of list element :",avg)
print("Id No.: D22DCE190")
```

Output:

```
ninadvyas@Ninads-MacBook-Air ~ % python -u "/Users/ninadvyas/Programs/DS-Algorithm/python.py"
Enter no of element:4
3
6
7
9
[3, 6, 7, 9]
Sum of list element : 25
Average of list element : 6.25
Id No.: D22DCE190
```

Learning Outcomes:

In this practical, we have learnt about:

i. The append method for enter the element in list
ii. Perform basic operations to get the sum and avg

4.7:

AIM:

Use comprehension method:

- Create a two separate list of even and odd numbers from 1 to 50.
- Get value which are divided by 5 from 1 to 100.

Tools and Technologies used:

- Python IDLE:

  IDLE is an integrated development environment for Python. It is packaged as an optional part of the Python packaging with many Linux distributions.

- Python Script:

  A Python script, the file containing the commands, is structured to be executed like a program. These files are designed to contain various functions and import various modules.

Create a two separate list of even and odd numbers from 1 to 50.

Program Code:

```
def EvenOdd(A ,B):
    List1 = [i for i in range(A , B) if i%2==0]
    List2 = [i for i in range(A , B) if i%2!=0]
    print("Even List : ", List1)
    print("Odd List : ", List2)
a = int(input("Enter Starting Number : "))
b = int(input("Enter Ending Number : "))
EvenOdd(a , b)
print("Id No.: D22DCE190")
```

Output:

Get value which are divided by 5 from 1 to 100.

Program Code:

```
def divisible5(A ,B):
    List1 = [i for i in range(A , B) if i%5==0]
    print("Numbers divisible by 5 : ", List1)
a = int(input("Enter Starting Number : "))
b = int(input("Enter Ending Number : "))
divisible5(a , b)
print("Id No.: D22DCE190")
```

Output:

```
ninadvyas@Ninads-MacBook-Air ~ % python -u "/Users/ninadvyas/Programs/DS-Algorithm/python.py"
Enter Starting Number : 5
Enter Ending Number : 16
Numbers divisible by 5 :  [5, 10, 15]
Id No.: D22DCE190
```

Learning Outcomes:

In this practical, we have learnt about :

i. Comprehension method to make long code to short code
ii. The direct pass the value and perform the operation on that values

Practical – 5

5.1

Aim: Create a class employee and display employee details.

Tools and Technologies used:

- Python IDLE: IDLE is Python's Integrated Development and Learning Environment. IDLE has two main window types, the Shell window and the Editor window. It is possible to have multiple editor windows simultaneously
- Python Script: A Python script, the file containing the commands, is structured to be executed like a program. These files are designed to contain various functions and import various modules.

Program Code:

```
class Employee:
    count=0
    def __init__(self, name, desig, salary):
        self.name=name
        self.desig=desig
        self.salary=salary
        Employee.count+=1
    def displayCount(self):
        print("There are %d employees" % Employee.count)
    def displayDetails(self):
        print("Name:", self.name, ", Designation:", self.desig, ", Salary:", self.salary)
e1=Employee("jyot", "Manager", 80000)
e2=Employee("ninad", "Team Leader", 50000)
e3=Employee("jay", "Programmer", 30000)
e4=Employee("Raj", "Assistant", 25000)
e4.displayCount()
print("Details of all employee:")
e1.displayDetails()
e2.displayDetails()
e3.displayDetails()
e4.displayDetails()
print("D22DCE190")
```

Output:

```
ninadvyas@Ninads-MacBook-Air hello % python -u "/Users/ninadvyas/Desktop/hello/src/components/Account/new.py"
There are 4 employees
Details of all employee:
Name: jyot , Designation: Manager , Salary: 80000
Name: ninad , Designation: Team Leader , Salary: 50000
Name: jay , Designation: Programmer , Salary: 30000
Name: Raj , Designation: Assistant , Salary: 25000
D22DCE190
```

Learning Outcomes:

I. In this practical I learnt how to use oop's concept.

5.2

Aim: From above create class count number of employee and display a salary amount if the salary is raised to 1.04%.

Tools and Technologies used:

- Python IDLE: IDLE is Python's Integrated Development and Learning Environment. IDLE has two main window types, the Shell window and the Editor window. It is possible to have multiple editor windows simultaneously
- Python Script: A Python script, the file containing the commands, is structured to be executed like a program. These files are designed to contain various functions and import various modules.

Program Code:

```
class Employee:
    def __init__(self, name, age, salary):
        self.name = name
        self.age = age
        self.salary = salary
    def display_employee_details(self):
        print("Name: ", self.name)
        print("Age: ", self.age)
        print("Salary: ", self.salary)
    def raise_salary(self):
        self.salary = self.salary * 1.04
        print("New Salary: ", self.salary)
employee1 = Employee("jyot Patel", 30, 50000)
employee1.raise_salary()
print("D22DCE190")
```

Output:

```
ninadvyas@Ninads-MacBook-Air hello % python -u "/Users/ninadvyas/Desktop/hello/src/components/Account/new.py"
New Salary:  52000.0
D22DCE190
```

Learning Outcomes:

I. In this practical I learnt how to use oop's concept and counting variable

5.3

Aim: Fetch children class details using different types of inheritance (Single, Multilevel, and Multiple) With constructor.

Tools and Technologies used:

- Python IDLE: IDLE is Python's Integrated Development and Learning Environment. IDLE has two main window types, the Shell window and the Editor window. It is possible to have multiple editor windows simultaneously
- Python Script: A Python script, the file containing the commands, is structured to be executed like a program. These files are designed to contain various functions and import various modules.

Program Code:

```
class Employee:
    def __init__(self, name, age, salary):
        self.name = name
        self.age = age
        self.salary = salary
    def display_employee_details(self):
        print("Name: ", self.name)
        print("Age: ", self.age)
        print("Salary: ", self.salary)
class Manager(Employee):
    def __init__(self, name, age, salary, department):
        super().__init__(name, age, salary)
        self.department = department
    def display_manager_details(self):
        super().display_employee_details()
        print("Department: ", self.department)
class ProjectManager(Manager):
    def __init__(self, name, age, salary, department, project):
        super().__init__(name, age, salary, department)
        self.project = project
    def display_project_manager_details(self):
        super().display_manager_details()
        print("Project: ", self.project)
manager1 = Manager("Jane Doe", 35, 60000, "Marketing")
```

```
manager1.display_manager_details()

projectManager1 = ProjectManager("Mike Smith", 40, 70000, "IT", "Project X")

projectManager1.display_project_manager_details()

print("D22DCE190")
```

Output

```
ninadvyas@Ninads-MacBook-Air hello % python -u "/Users/ninadvyas/Desktop/hello/src/components/Account/new.py"
Name:  Jane Doe
Age:  35
Salary:  60000
Department:  Marketing
Name:  Mike Smith
Age:  40
Salary:  70000
Department:  IT
Project:  Project X
D22DCE190
```

:

Learning Outcomes:

I. In this practical I learnt concepts of inheritance.

5.4

Aim: Find who will be first among two students using polymorphism.

Tools and Technologies used:

- Python IDLE: IDLE is Python's Integrated Development and Learning Environment. IDLE has two main window types, the Shell window and the Editor window. It is possible to have multiple editor windows simultaneously
- Python Script: A Python script, the file containing the commands, is structured to be executed like a program. These files are designed to contain various functions and import various modules.

Program Code:

```
class Student:
    def __init__(self, name, age, marks):
        self.name = name
        self.age = age
        self.marks = marks
    def display_student_details(self):
        print("Name: ", self.name)
        print("Age: ", self.age)
        print("Marks: ", self.marks)
class FirstStudent(Student):
    def __init__(self, name, age, marks):
        super().__init__(name, age, marks)
    def find_first(self, student2):
        if self.marks > student2.marks:
            print(self.name, " is first")
        else:
            print(student2.name, " is first")
student1 = FirstStudent("John Smith", 20, 85)
student2 = FirstStudent("Jane Doe", 22, 90)
student1.find_first(student2)
print("D22DCE190")
```

Output:

```
ninadvyas@Ninads-MacBook-Air hello % python -u "/Users/ninadvyas/Desktop/hello/src/components/Account/new.py"
Jane Doe  is first
D22DCE190
```

Learning Outcomes:

I. In this practical I learnt concepts of polymorphism.

Practical – 6

6.1

Aim: Consider an example of declaring the examination result. Design three classes: Student, Exam, and Result. The Student class has data members such as those representing rollNumber, Name, etc. Create the class Exam by inheriting Student class. The Exam class adds fields representing the marks scored in six subjects. Derive Result from the Exam class, and it has its own fields such as total marks. Write an interactive program to model this relationship.

Tools and Technologies used:

- Python IDLE: IDLE is Python's Integrated Development and Learning Environment. IDLE has two main window types, the Shell window and the Editor window. It is possible to have multiple editor windows simultaneously
- Python Script: A Python script, the file containing the commands, is structured to be executed like a program. These files are designed to contain various functions and import various modules.

Program Code:

```
class Student:
 def __init__(self,name,eno):
    self.name=name
    self.eno=eno
class Exam(Student):
 def __init__(self,name,eno,sub1,sub2,sub3,sub4,sub5,sub6):
    self.sub1=sub1
    self.sub2=sub2
    self.sub3=sub3
    self.sub4=sub4
    self.sub5=sub5
    self.sub6=sub6
    super().__init__(name,eno)
class Result(Exam):
 def __init__(self,name,eno,sub1,sub2,sub3,sub4,sub5,sub6):
    super().__init__(name,eno,sub1,sub2,sub3,sub4,sub5,sub6)
    self.total_marks=self.sub1+self.sub2+self.sub3+self.sub4+self.sub5+self.sub5+self.sub6
 def show_detail(self):
```

```
    print("Student name is:",self.name)
    print("Student number is:",self.eno)
    print("Student sub1 mark is:",self.sub1)
    print("Student sub2 mark is:",self.sub2)
    print("Student sub3 mark is:",self.sub3)
    print("Student sub4 mark is:",self.sub4)
    print("Student sub5 mark is:",self.sub5)
    print("Student sub6 mark is:",self.sub6)
    print("Student total mark is:",self.total_marks)
name=input("enter the name:")
eno=input("enter the id number:")
sub1=int(input("enter marks of sub1:"))
sub2=int(input("enter marks of sub2:"))
sub3=int(input("enter marks of sub3:"))
sub4=int(input("enter marks of sub4:"))
sub5=int(input("enter marks of sub5:"))
sub6=int(input("enter marks of sub6:"))
r=Result(name,eno,sub1,sub2,sub3,sub4,sub5,sub6)
r.show_detail()
print("D22DCE190")
```

Output:

```
ninadvyas@Ninads-MacBook-Air hello % python -u "/Users/ninadvyas/Desktop/hello/src/components/Account/new.py"
enter the name:ninad
enter the id number:d22dce190
enter marks of sub1:98
enter marks of sub2:89
enter marks of sub3:78
enter marks of sub4:67
enter marks of sub5:98
enter marks of sub6:90
Student name is: ninad
Student number is: d22dce190
Student sub1 mark is: 98
Student sub2 mark is: 89
Student sub3 mark is: 78
Student sub4 mark is: 67
Student sub5 mark is: 98
Student sub6 mark is: 90
Student total mark is: 618
D22DCE190
```

Learning Outcomes:

I. In this practical I learnt concepts of multilevel inheritance

Practical – 7

7.1 :

Aim: Create a different package of addition, division, multiplication, subtraction, factorial, and Fibonacci-series and use it in result.py file.

Tools and Technologies used:

- Python IDLE: IDLE is Python's Integrated Development and Learning Environment. IDLE has two main window types, the Shell window and the Editor window. It is possible to have multiple editor windows simultaneously
- Python Script: A Python script, the file containing the commands, is structured to be executed like a program. These files are designed to contain various functions and import various modules.

addition.py

Program Code:

```
def add():
a=5+7
print(a)
```

subtraction.py

Program Code:

```
def sub():
a=9-8
print(a)
```

multiplication.py:

Program Code:

```
def mul():
a=5*7
47
```

division.py

Program Code:

```
def div():
a=9/3
print(a)
```

factorial.py

Program Code:

```
def fac():
   e=7*6*5*4*3*2*1;
     print(e)
```

fibonacci.py

```
def fibonacci_series():
n = int(input("Enter your Value:"))
if n <= 0:
print("Invalid input")
elif n == 1:
print(0)
elif n == 2:
print(0,1)
else:
series = [0,1]
for i in range(2,n):
series.append(series[i-1]+series[i-2])
print(series)
```

result.py

Program Code:

```
from python_package import addition
```

```
addition.add()
from python_package import subtraction
subtraction.sub()
from python_package import mul
mul.mul()
from python_package import div
div.div()
from python_package import factorial
factorial.fac()
from python_package import fibonacci
fibonacci.fibonacci_series()
print("D22DCE190")
```

Output:

```
ninadvyas@Ninads-MacBook-Air Python % python -u "/Users/ninadvyas/Programs/Python/result.py"
12
5
10
5.0
5040
Enter your Value:10
[0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
D22DCE190
```

Learning Outcomes:

i. In this practical I have learnt how to build a package.

ii. Also learnt how to import package in desired file and how to perform task using packages.

7.2:

Aim: Different type of pandas and NumPy operations, and create charts using matplotlib.

Tools and Technologies used:

- Python IDLE: IDLE is Python's Integrated Development and Learning Environment. IDLE has two main window types, the Shell window and the Editor window. It is possible to have multiple editor windows simultaneously
- Python Script: A Python script, the file containing the commands, is structured to be executed like a program. These files are designed to contain various functions and import various modules.

Pandas:

Program Code:

```
import pandas as pd
# create two Series
s1 = pd.Series([1, 2, 3, 4])
s2 = pd.Series([5, 6, 7, 8])
# addition
s3 = s1 + s2
print("Addition:")
print(s3)
# subtraction
s3 = s1 - s2
print("Subtraction:")
print(s3)
# multiplication
s3 = s1 * s2
print("Multiplication:")
print(s3)
# division
```

```
s3 = s1 / s2
print("Division:")
print(s3)
print("D22DCE190")
```

Output:

```
Addition:
0     6
1     8
2    10
3    12
dtype: int64
Subtraction:
0   -4
1   -4
2   -4
3   -4
dtype: int64
Multiplication:
0     5
1    12
2    21
3    32
dtype: int64
Division:
0    0.200000
1    0.333333
2    0.428571
3    0.500000
dtype: float64
D22DCE190
```

Numpy:

```
import numpy as np
num1=np.array([1,4,5])
num2=np.array([5,6,7])
add=num1+num2
print(add)
```

```
sub=num2-num1
print(sub)
mul=num1 * num2
print(mul)
div=num2/num1
print(div)
print("D22DCE190")
```

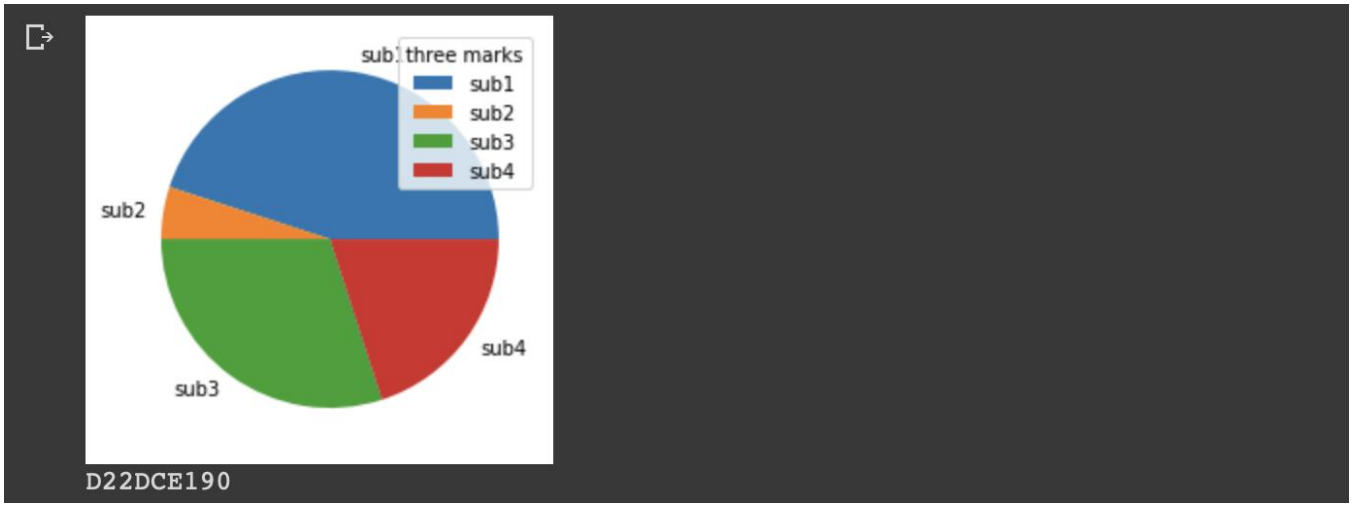
Output:

```
[ 6 10 12]
[4 2 2]
[ 5 24 35]
[5.  1.5 1.4]
D22DCE190
```

Matplotlib:

```
import numpy as np
import matplotlib.pyplot as plt
num1=np.array([45,5,30,20])
mylabels=["sub1","sub2","sub3","sub4"]
plt.pie(num1,labels=mylabels)
plt.legend(title="three marks")
plt.show()
print("D22DCE190")
```

Output:



Learning Outcomes:

i. In this practical I have learnt about some python libraries like python,numpy and matplotlib.

ii. I have also learnt how to perform basic operations using these libraries like basic mathematics operations.

Practical – 8

8.1:

Aim: Perform below operations

- **Create database**
- **Create table**
- **Database version**
- **Delete operation**
- **Insert data to database**
- **Select data from database**
- **Update data in database**

Tools and Technologies used:

- Python IDLE: IDLE is Python's Integrated Development and Learning Environment. IDLE has two main window types, the Shell window and the Editor window. It is possible to have multiple editor windows simultaneously
- Python Script: A Python script, the file containing the commands, is structured to be executed like a program. These files are designed to contain various functions and import various modules.

Create Database

Program Code:

```
import mysql.connector
conn=mysql.connector.connect(host="localhost",user="D22DCE180",password="avadh@007")
mycursor=conn.cursor()
mycursor.execute("CREATE DATABASE pythondb")
print(conn)
print("D22DCE190")
```

Output:

```
<mysql.connector.connection.MySQLConnection object at 0x000002A3B6570748>
D22DCE190
```

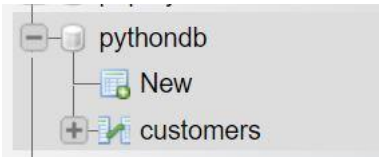
Create Table

Program Code:

```
import mysql.connector

conn=mysql.connector.connect(host="localhost",user="D22DCE180",password="avadh@007",database
="pythondb")

mycursor=conn.cursor()

mycursor.execute("CREATE TABLE customers (name VARCHAR(255), address VARCHAR(255))")

print("Table has been created")

print("D22DCE190")
```

Output:

```
Table has been created
D22DCE190
```



MySql Version

Program Code:

```
import mysql.connector

conn=mysql.connector.connect(host="localhost",user="D22DCE180",password="avadh@007",database
="pythondb")

mycursor=conn.cursor()

mycursor.execute("SELECT VERSION()")

result=mycursor.fetchone()

print("MySql Version: ",result[0])

print("D22DCE190")
```

Output:

```
MySql Version:  10.4.27-MariaDB
D22DCE190
```

Insert Data

Program Code:

```
import mysql.connector

conn=mysql.connector.connect(host="localhost",user="D22DCE180",password="avadh@007",database
="pythondb")

mycursor=conn.cursor()

insert_query="INSERT INTO customers(name,address) VALUES(%s,%s)"

val=[
   ('abc', 'DEPSTAR'),
   ('xyz', 'CSPIT'),
   ('def', 'EC'),
   ('hij', 'IIIM'),
   ('kbc', 'CMPICA'),
]

mycursor.executemany(insert_query,val)

conn.commit()
```

```
print(mycursor.rowcount,"record Was inserted.")
print("D22DCE190")
```

Output:

```
5 record Was inserted.
D22DCE190
```

name	address
abc	DEPSTAR
xyz	CSPIT
def	EC
hij	IIIM
kbc	CMPICA

Select Data

Program Code:

```
import mysql.connector
conn=mysql.connector.connect(host="localhost",user="D22DCE180",password="avadh@007",database
="pythondb")
mycursor=conn.cursor()
mycursor.execute("SELECT * FROM customers")
select_result=mycursor.fetchall()
for record in select_result:
    print(record)
print("D22DCE190")
```

Output:

```
('abc', 'DEPSTAR')
('xyz', 'CSPIT')
('def', 'EC')
('hij', 'IIIM')
('kbc', 'CMPICA')
D22DCE190
```

Update Data

Program Code:

```
import mysql.connector

conn=mysql.connector.connect(host="localhost",user="D22DCE180",password="avadh@007",database
="pythondb")

mycursor=conn.cursor()

mycursor.execute("UPDATE customers SET address='DEPSTAR' WHERE name='xyz'")

conn.commit()

print(mycursor.rowcount,"record(s) affected")

print("D22DCE190")
```

Output:

```
1 record(s) affected
D22DCE190
```

abc	DEPSTAR
xyz	DEPSTAR
def	EC
hij	IIIM
kbc	CMPICA

Delete Data

Program Code:

```
import mysql.connector

conn=mysql.connector.connect(host="localhost",user="D22DCE180",password="avadh@007",database
="pythondb")

mycursor=conn.cursor()

mycursor.execute("DELETE FROM customers WHERE address='DEPSTAR'")

conn.commit()

print(mycursor.rowcount,"record(s) deleted")

print("D22DCE190")
```

Output:

```
3 record(s) deleted
D22DCE190
```

def	EC
kbc	CMPICA
def	EC
kbc	CMPICA

Learning Outcomes:

i. In this practical I have learnt how to deal with the database in python

ii. To do so I have installed mysql-connector library to perform different operations.

iii. I have performed operations like create database, create table, insert data in to table, select data from the table, update the table data as well as delete record from the table.