

Python Programming Lab

Subject Code: PC/CDS/46-P

Python Programming Lab

General Course Information

Course Code: PC/CDS/46-P Course Credits:2 Type: Professional Core Lab. Course ContactHours:4hours/week Mode: Lab practice and assignments	Course Assessment Methods (internal: 50; external: 50) The internal and external assessment is based on the level of participation in lab. Sessions and the timely submission of lab experiments/assignments, the quality of solutions designed for the assignments, the performance in VIVA- VOCE, the quality of lab. file and ethical practices followed. The internal examination is conducted by the course coordinator. The external examination is conducted by external examiner appointed by the Controller of Examination in association with the internal examiner appointed by the Chairperson of the Department.
---	--

Course Outcomes: By the end of the course students will be able to:

- CO1. **implement** solutions to the given assignments in Python. (LOTS: Level3: Apply)
- CO2. **use** various Python packages for solving different programming problems. (LOTS: Level 3: Apply)
- CO3. **Devise** solutions for complex problems of data analysis and machine learning. (LOTS:Level 6: Create)
- CO4. **Evaluate** the output of data analysis and machine learning models. (LOTS: Level 5: Evaluate)
- CO5.**create** lab records of the solutions for the given assignments. (LOTS: Level6: Create)
- CO6. **Demonstrate** use of ethical practices, self-learning and teamspirit. (LOTS: Level3: Apply)

List of experiments/assignments

- Install Python and explore various popular IDE like IDLE, PyCharm, and Anaconda.
- Assignments to perform various number operations.
- Find maximum from a list of numbers
- GCD of two number
- Square root of a number
- Check number is prime or not.
- Print first N prime numbers
- Remove duplicate numbers from list
- Print the Fibonacci series.
- Assignments to perform various operations on Strings like creation, deletion, concatenation.

- Create a List L=[10,20,30].Write programs to perform following operations:
- Insert new numbers to list L.
- Delete numbers from list L.
- Sum all numbers in list L.
- Sum all prime numbers in list L.
- Delete the list L.

Note:

The actual experiments/assignments will be designed by the course coordinator. One assignment should be designed to be done in groups of two or three students. The assignments must meet the objective of the course and the levels of the given course outcomes. The list of assignments and schedule of submission will be prepared by the course coordinator at the beginning of the semester.

CO-PO Articulation Matrix Python Programming Lab. Course (PC/CDS/46-P)

Course Outcomes	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 13	PSO 14	PSO 15
CO1	2	1	–	–	3	–	–	–	–	–	–	–	–	–	3
CO2	2	3	–	3	3	–	–	–	–	–	–	–	–	–	3
CO3	3	3	1	3	3	–	–	–	–	–	–	–	–	–	3
CO4	3	3	–	3	3	–	–	–	–	–	–	–	–	–	3
CO5	–	–	–	–	–	–	–	–	–	3	–	–	–	–	–
CO6	–	–	–	–	–	–	–	3	3	–	–	3	–	–	–
3-High2-Medium1-Low															

Program 1: Installation of Python & IDEs

Aim

To install Python and explore different IDEs like IDLE, PyCharm, and Anaconda.

Theory

Python is a high-level programming language. IDEs (Integrated Development Environments) provide tools for coding, debugging, and execution.

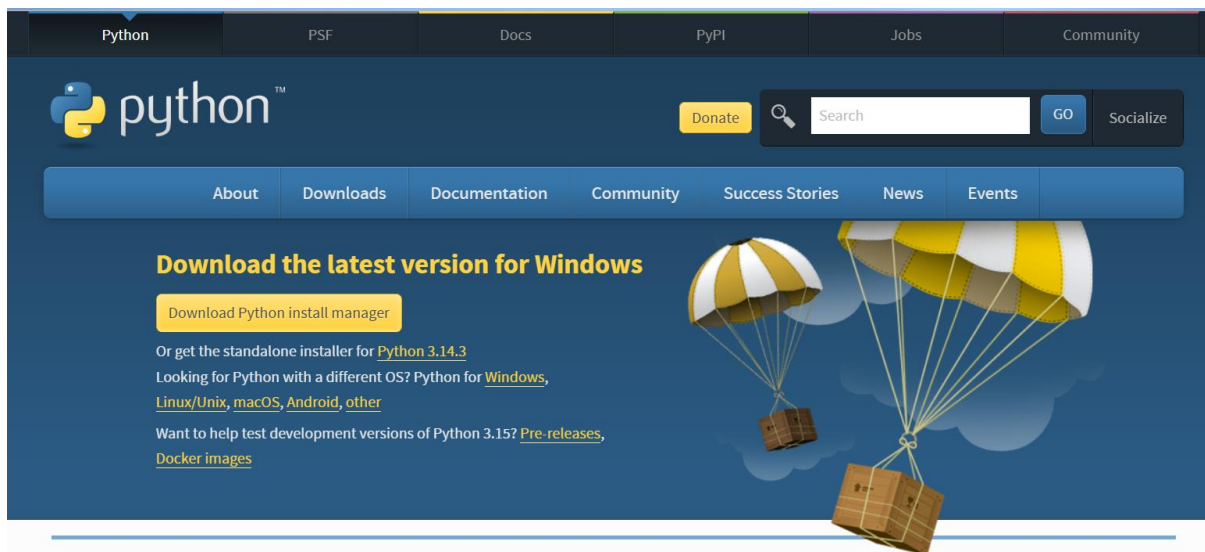
Procedure

1. Download Python from official website.
2. Install Python.
3. Open IDLE and run a simple program.
4. Install PyCharm/Anaconda.
5. Explore features like editor, debugger.

Installation of Python on windows.

To install Python on Windows, follow these steps:

Download the Python installer from the official Python website. Choose the installer that matches your operating system.

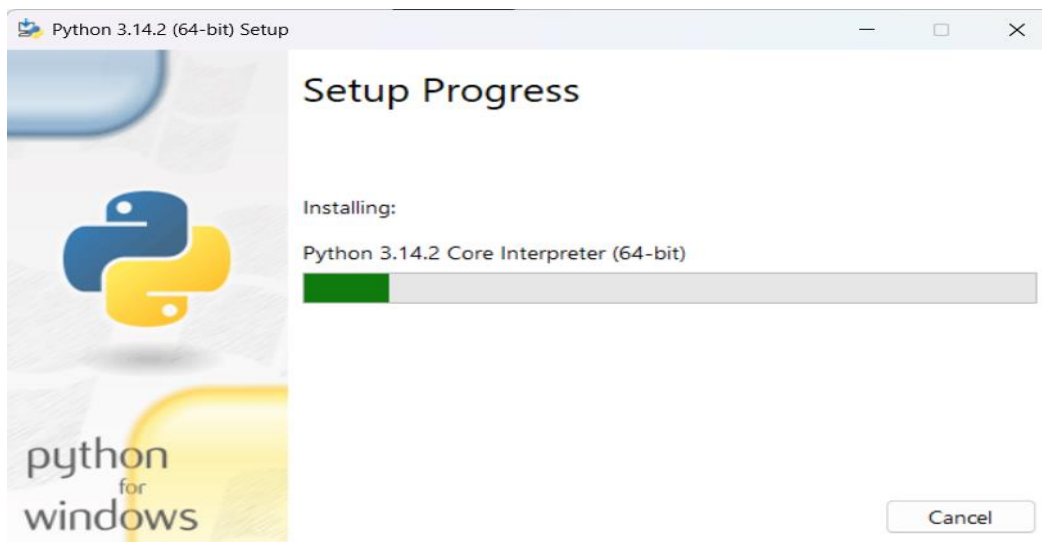


Run the downloaded installer file by double-clicking it.

Click on "Install Now" to proceed with the default installation settings. Ensure that the "Add Python 3.x to PATH" option is checked to allow Python to be run directly from the Command Prompt.



Wait for the installation to complete. Once done, you will see a window indicating that the setup was successful. Click "Close" to finish.





To verify the installation, open the Command Prompt and type “python— version”. This command will display the installed Python version, confirming that Python is correctly installed.

```
Microsoft Windows [Version 10.0.22631.5039]
(c) Microsoft Corporation. All rights reserved.

C:\Users\HP>python --version
Python 3.13.2

C:\Users\HP>|
```

Program
`print("Hello, Python!")`

Output
Hello, Python!

Practical 2: Maximum Number from List.

Aim

To find the maximum number from a list.

Program

```
lst = [10, 45, 67, 23]  
print("Maximum:", max(lst))
```

Output

Maximum: 67

Practical 3: GCD of Two Numbers

Aim

To find GCD of two numbers.

Program

```
import math
a = 36
b = 60
print("GCD:", math.gcd(a, b))
```

Output

GCD: 12

Practical 4: Square Root of a Number

Aim

To calculate square root.

Program

```
import math
num = 25
print("Square Root:", math.sqrt(num))
```

Output

Square Root: 5.0

Practical 5: Check Prime Number

Aim

To check whether a number is prime.

Program

```
num = 7
flag = True

if num <= 1:
    flag = False
else:
    for i in range(2, num):
        if num % i == 0:
            flag = False
            break

if flag:
    print("Prime Number")
else:
    print("Not Prime")
```

Output

Prime Number

Practical 6: First N Prime Numbers

Aim

To print first N prime numbers.

Program

```
n = 5
```

```
count = 0
```

```
num = 2
```

```
while count < n:
```

```
    for i in range(2, num):
```

```
        if num % i == 0:
```

```
            break
```

```
    else:
```

```
        print(num, end=" ")
```

```
        count += 1
```

```
        num += 1
```

Output

```
2 3 5 7 11
```

Practical 7: Remove Duplicates from List

Aim

To remove duplicate elements.

Program

```
lst = [1, 2, 2, 3, 4, 4]
unique = list(set(lst))
print(unique)
```

Output

```
[1, 2, 3, 4]
```

Practical 8: Fibonacci Series

Aim

To print Fibonacci series.

Program

```
n = 6
```

```
a, b = 0, 1
```

```
for i in range(n):  
    print(a, end=" ")  
    a, b = b, a + b
```

Output

```
0 1 1 2 3 5
```

Practical 9: String Operations

Aim

To perform string operations.

Program

```
str1 = "Hello"  
str2 = "World"
```

```
# Concatenation  
print(str1 + " " + str2)
```

```
# Length  
print(len(str1))
```

```
# Deletion  
del str2
```

Output

```
Hello World  
5
```

Practical 10: List Operations

Aim

To perform operations on list L = [10, 20, 30]

Program

```
L = [10, 20, 30]
```

```
# Insert
```

```
L.append(40)
```

```
print("After Insert:", L)
```

```
# Delete
```

```
L.remove(20)
```

```
print("After Delete:", L)
```

```
# Sum
```

```
print("Sum:", sum(L))
```

```
# Sum of Prime Numbers
```

```
def is_prime(n):
```

```
    if n < 2:
```

```
        return False
```

```
    for i in range(2, n):
```

```
        if n % i == 0:
```

```
            return False
```

```
    return True
```

```
prime_sum = sum(x for x in L if is_prime(x))
```

```
print("Sum of Prime:", prime_sum)
```

```
# Delete List
```

```
del L
```

Output

```
After Insert: [10, 20, 30, 40]
```

```
After Delete: [10, 30, 40]
```

```
Sum: 80
```

```
Sum of Prime: 0
```

Practical 11: Factorial of a Number

Aim

To find factorial of a number.

Program

```
num = 5  
fact = 1
```

```
for i in range(1, num+1):  
    fact *= i
```

```
print("Factorial:", fact)
```

Output

Factorial: 120

Practical 12: Palindrome Number

Aim

To check whether a number is palindrome.

Program

```
num = 121
```

```
temp = num
```

```
rev = 0
```

```
while num > 0:
```

```
    digit = num % 10
```

```
    rev = rev * 10 + digit
```

```
    num //= 10
```

```
if temp == rev:
```

```
    print("Palindrome")
```

```
else:
```

```
    print("Not Palindrome")
```

Output

Palindrome

Practical 13: Armstrong Number

Aim

To check Armstrong number.

Program

```
num = 153
temp = num
sum = 0
```

```
while num > 0:
    digit = num % 10
    sum += digit**3
    num //= 10
```

```
if temp == sum:
    print("Armstrong Number")
else:
    print("Not Armstrong")
```

Output

Armstrong Number

Practical 14: Reverse a String

Aim

To reverse a string.

Program

```
str1 = "Python"  
print("Reversed:", str1[::-1])
```

Output

Reversed: nohtyP

Practical 15: Count Vowels in String

Aim

To count vowels in a string.

Program

```
str1 = "hello world"  
count = 0
```

```
for ch in str1:  
    if ch in "aeiouAEIOU":  
        count += 1
```

```
print("Vowels:", count)
```

Output

Vowels: 3