SEMESTER-V

St. Xavier's College (Autonomous), Ahmedabad

Syllabus of Semester – V of the following departments under Faculty of Computer Science based on Undergraduate Curriculum Framework to be implemented from the Academic Year 2025-26.

DEPARTMENT OF COMPUTER SCIENCE

Major Course – 1: Python Programming

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course Title & Code	Credit Distribution of The Course			Eligibility Criteria	Pre-requisite(s) of the Course (if any)
	Lecture	Tutorial	Practical / Practice		
Python Programming CSMC551C	2	0	2	Nil	Programming Basics

Learning Objectives:
This course introduces students to Python programming, covering fundamental concepts such as syntax, control structures, and data structures. Students will learn to write efficient programs, handle files, and implement Object-Oriented Programming (OOP).

Learning Outcomes:
Upon successful completion, students will be able to implement:

- Write and execute Python programs using variables, operators, and expressions.
- Implement decision-making statements (if-else) and loops (for, while).
- Define functions, pass arguments, and handle exceptions effectively.
- Work with strings, lists, tuples, dictionaries, and sets.
- Perform file handling operations like reading, writing, and appending files.
- Implement mathematical and logical operations such as factorial, prime numbers, Fibonacci series, HCF, and LCM.
- Use list comprehension and set operations for efficient data processing.
- Apply OOP principles including classes, inheritance, and operator overloading.
- Read and process files for statistical analysis, computing mean, median, and mode.
- Handle errors and exceptions in Python programs.

UNIT	Particulars	Hours
I	▪ **Introduction to Python** ▪ History and evolution of Python	15

	▪ Features and applications of Python ▪ Installing Python and setting up the environment ▪ **Python Basics** ▪ Variables, constants, and data types ▪ Operators: Arithmetic, Relational, Logical, Bitwise ▪ Taking input and printing output ▪ Type conversion and typecasting ▪ **Control Structures** ▪ Conditional statements: if, elif, else ▪ Looping structures: for, while ▪ Loop control statements: break, continue, pass ▪ **Functions and Exception Handling** ▪ Defining and calling functions ▪ Function arguments and return values ▪ Recursion vs. Iteration ▪ Exception handling (try, except, finally)	
2	▪ Basic Python Programs ▪ Demonstration of Decision Statements ▪ Implementation of Looping Structures ▪ Demonstration on Functions ▪ Implementation on Exception Handling	30
3	▪ **Strings and String Manipulation** • String operations (split(), join(), replace(), find()) • String formatting (format(), f-strings) • Reversing and modifying strings ▪ **Data Structures** ▪ Lists, tuples, dictionaries, sets ▪ List comprehension and set comprehension ▪ Sorting and filtering data ▪ **File Handling** • Reading, writing, and appending files • Counting lines and words in a file • Copying and modifying files • Checking file status (closed or not)	15
4	▪ **Object-Oriented Programming** • Classes and objects • Constructors (__init__) • Operator overloading (e.g., Complex number operations) • Inheritance and polymorphism ▪ Implementation on String Manipulation ▪ Working with Data Structures ▪ File Handling Programs ▪ Object-Oriented Programming Applications	30

Textbook:

1. Python Programing for the absolute beginner 3rd edition – Michael Dawson – Cengage Learning.
2. Python 3 for Absolute Beginners – Tim Hall and J P Stacey – Apress.

Reference Books:

1. Introducing Python - by Bill Lubanovic - O'Reilly
2. Python Cookbook – Alex Martelli, Anna Martelli Ravenscroft and David Ascher - O'Reilly
3. Think Python – Allen B Downey - O'Reilly

4. Python Data Science Cookbook – Gopi Subramanian – PACKT Publishing.
5. Python for Data Analysis – Wes McKinney - O'Reilly
6. Python Machine Learning – Sebastian Rachka - PACKT Publishing

Sample Programs

1. Write a function to print the factorial of a number{no recursion}
2. Calculate simple interest. accept P,T,R from user.
3. To check the given number is armstrong.
4. To print all armstrong numbers between two range.
5. To check a given number is prime. use function
6. print all prime numbers between two range
7. To print nth prime number
8. To print all composite numbers within n. use function
9. Find the area of circle.
10. To print all fibonacci number till use{ 0,1,1,2,3,.}use function
11. To print nth fibonacci number
12. sum of the square of 1st n natural numbers using function
13. sum of the cube of 1st n natural numbers
14. to check palindrome.use 2 functions
15. Reverse words in a long string using function
16. To remove a character from a string at specifc position and print the remaining string.
17. input a string and split the string into words and print only the odd length of words in that string
18. Find the length of a string without using built in function len() 3.. swap commas and dots in a string
19. find words whose length greater than n in a given sentence.
20. count the number of vowels in a list.
21. prints all the numbers from 12 to 31 except 17 and 19.using 'continue statement
22. calculate the sum and average of n integer numbers given by user. Input 0 to finish
23. to check the triangle is equilateral, isosceles, scalene.(nested if)
24. print all numbers which are divisible by 2 and 7 in the range of within n
25. convert temperatures for celsius entry to fahrenheit and vice versa{c/5=f-32/9}
26. Input a year and check for leapyear.
27. count the number of even and odd numbers from a given tuple
28. Print mutiplication table of 11,12,13 (1*11 to 10*11)
29. print number of alphabets,special characters and numbers in the string input numbers between two numbers, print the odd numbers separated by comma{237=3,7}
30. Program to calculate the HCF and LCM of two numbers.
31. Program to make a simple calculator
32. Write a Python program to read an entire text file.
33. Write a Python program to read first n lines of a file.
34. Write a Python program to append text to a file and display the text.
35. Write a Python program to read a file line by line and store it into a list.
36. Write a Python program to count the number of lines in a text file.
37. Write a Python program to count the frequency of words in a file.
38. Write a Python program to get the file size of a plain file.
39. Write a Python program to write a list to a file.
40. Write a Python program to copy the contents of a file to another file.

41. Write a Python program to combine each line from first file with the corresponding line in second file.
42. Write a Python program to read a random line from a file.
43. Write a Python program to assess if a file is closed or not.
44. Write a Python program to remove newline characters from a file.
45. Write a function using exceptions that asks the user to enter a set of integers. Calculate mean. The user indicates that she is done by typing “Done”.
46. Write two versions of a function that calculates the harmonic mean of a list of integers The first version uses an if-statement to make sure that the list element is not zero. List elements with zero value are ignored. The second version uses exceptions to do the same thing.

    n/(1/a1+1/a2+1/a3….)

47. Write a function that takes a list as a single argument. The function returns the number of items that cannot be converted into a floating point number. Use exceptions.
48. Perform the above question with the help of isinstance() method
49. Program to add, subtract, equality of two complex numbers
50. Program to multiply and divide of two complex numbers.
51. Use list comprehension to generate a. list of all cube between 1 and 100
52. Use list comprehension to generate a list of all multiples of 7 between 1 and 100
53. Use set comprehension and intersection to find all numbers divisible by 15 and divisible y 21 between 1 and 1000
54. Create a list of the first 20 powers of 2 [1,2,4,8,16……]
55. Create a list of all numbers between 10000 and 20000that have last digit 3 and are divisible by 13
56. Work with all widgets of Tkinter
57. Write a program to perform descriptive statistics of a set of numbers