DEPARTMENT OF COMPUTER SCIENCE
B.Sc. (H) Computer Science

CATEGORY-I

DISCIPLINE SPECIFIC CORE COURSE – 1: PROGRAMMING USING PYTHON

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Programming using Python	4	3	0	1	Class XII pass	Nil

Learning Objectives

This course is designed as the first course that:
- Introduces programming concepts using Python to Computer Science students.
- Focuses on the development of Python programming to solve problems of different domains.
- Introduces the concept of object- oriented programming.

Learning Outcomes:

On successful completion of the course, students will be able to:
- Understand the basics of programming language
- Develop, document, and debug modular Python programs.
- Apply suitable programming constructs and built-in data structures to solve a problem.
- Use and apply various data objects in Python.
- Use classes and objects in application programs and handle files.

SYLLABUS OF DSC-1

Theory

Unit – 1   **(6 hours)**
Introduction to Programming
Problem solving strategies; Structure of a Python program; Syntax and semantics; Executing simple programs in Python.

Unit – 2   **(12 hours)**

Creating Python Programs

Identifiers and keywords; Literals, numbers, and strings; Operators; Expressions; Input/output statements; Defining functions; Control structures (conditional statements, loop control statements, break, continue and pass, exit function), default arguments.

Unit – 3 (15 hours)
Built-in Data Structures

Mutable and immutable objects; Strings, built-in functions for string, string traversal, string operators and operations; Lists creation, traversal, slicing and splitting operations, passing list to a function; Tuples, sets, dictionaries and their operations.

Unit – 4 (6 hours)
Object Oriented Programming

Introduction to classes, objects and methods; Standard libraries.

Unit – 5 (6 hours)
File and Exception Handling

File handling through libraries; Errors and exception handling.

Practical (30 hours)

List of Practicals:

1. WAP to find the roots of a quadratic equation
2. WAP to accept a number 'n' and

    j. Check if 'n' is prime

    k. Generate all prime numbers till 'n'

    l. Generate first 'n' prime numbers This program may be done using functions

3. WAP to create a pyramid of the character '*' and a reverse pyramid

```
    *
   ***
  *****
 *******
*********

        *********
         *******
          *****
           ***
            *
```

8. WAP that accepts a character and performs the following:

    a. print whether the character is a letter or numeric digit or a special character

    b. if the character is a letter, print whether the letter is uppercase or lowercase

    c. if the character is a numeric digit, prints its name in text (e.g., if input is 9, output is NINE)

9. WAP to perform the following operations on a string
    a. Find the frequency of a character in a string.
    b. Replace a character by another character in a string.
    c. Remove the first occurrence of a character from a string.
    d. Remove all occurrences of a character from a string.

10. WAP to swap the first n characters of two strings.

11. Write a function that accepts two strings and returns the indices of all the occurrences of the second string in the first string as a list. If the second string is not present in the first string then it should return -1.

12. WAP to create a list of the cubes of only the even integers appearing in the input list (may have elements of other types also) using the following:
    a. 'for' loop
    b. list comprehension

13. WAP to read a file and
    m. Print the total number of characters, words and lines in the file.
    n. Calculate the frequency of each character in the file. Use a variable of dictionary type to maintain the count.
    o. Print the words in reverse order.
    p. Copy even lines of the file to a file named 'File1' and odd lines to another file named 'File2'.

14. WAP to define a class Point with coordinates x and y as attributes. Create relevant methods and print the objects. Also define a method distance to calculate the distance between any two point objects.

15. Write a function that prints a dictionary where the keys are numbers between 1 and 5 and the values are cubes of the keys.

16. Consider a tuple t1=(1, 2, 5, 7, 9, 2, 4, 6, 8, 10). WAP to perform following operations:
    a. Print half the values of the tuple in one line and the other half in the next line.
    b. Print another tuple whose values are even numbers in the given tuple.
    c. Concatenate a tuple t2=(11,13,15) with t1.
    d. Return maximum and minimum value from this tuple

17. WAP to accept a name from a user. Raise and handle appropriate exception(s) if the text entered by the user contains digits and/or special characters.

Essential Readings
- Taneja, S., Kumar, N. Python Programming- A modular Approach, 1st edition, Pearson Education India, 2018.

- Balaguruswamy E. Introduction to Computing and Problem Solving using Python, 2nd edition, McGraw Hill Education, 2018.

Suggestive Readings

- Brown, Martin C. Python: The Complete Reference, 2nd edition, McGraw Hill Education,
2018.
- Guttag, J.V. Introduction to computation and programming using Python, 2nd edition, MIT Press, 2016.

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

DISCIPLINE SPECIFIC CORE COURSE – 2: COMPUTER SYSTEM ARCHITECTURE

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Computer System Architecture	4	3	0	1	Class XII pass	NIL

Learning Objectives

The Learning Objectives of this course are as follows:

- Introduces the students to the fundamental concepts of digital computer organization, design and architecture.
- Develop a basic understanding of the building blocks of the computer system and highlights how these blocks are organized together to architect a digital computer system.

Learning Outcomes

On successful completion of the course, students will be able to:

- Design Combinational Circuits using basic building blocks. Simplify these circuits using Boolean algebra and Karnaugh maps. Differentiate between combinational circuits and sequential circuits.
- Represent data in binary form, convert numeric data between different number systems and perform arithmetic operations in binary.
- Determine various stages of instruction cycle, pipelining and describe interrupts and their handling.
- Explain how CPU communicates with memory and I/O devices and distinguish between different types of processors.
- Simulate the design of a basic computer using a software tool.