|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **III Year – I Semester** |  | **L** | **T** | **P** | **C** |
| **Course Code (enter course code only** | **INDUSTRIAL ORIENTED PYTHON PROGRAMMING** |  |  |  |  |

**COURSE OBJECTIVES:**

* To learn about Python programming language syntax, semantics, and the runtime environment.
* To be familiarized with universal computer programming concepts like data types, containers.
* To be familiarized with general computer programming concepts like conditional execution, loops & functions.
* To be familiarized with general coding techniques and object-oriented programming

**COURSE OUTCOMES:**

|  |  |  |  |
| --- | --- | --- | --- |
| **CO’s** | **At the end of the course, the student will have the ability to:** | **POs Mapped** | **Strength of mapping** |
| **CO1** | Develop essential programming skills in computer programming concepts like data types, containers | PO1 | 3 |
| **CO2** | Apply the basics of programming in the Python language | PO1,  PO2 | 3  3 |
| **CO3** | Solve coding tasks related to the fundamental notions and techniques used in object-oriented programming. | PO5 | 2  3 |
| **CO4** | Solve coding tasks in Dynamic programming. | PO12 | 3 |

\*\*Strength of mapping (Intensity Scale) – 1(Lightly mapped), 2(Moderately mapped), 3(Heavily mapped)

**UNIT- I**

**INTRODUCTION TO PYTHON**

Introduction: History of Python, Need of Python Programming, Variables, Keywords, Input-Output, Indentation. Types, Operators and Expressions: Types - Integers, Strings, Booleans; Operators- Arithmetic Operators, Comparison (Relational) Operators, Assignment Operators, Logical Operators, Bitwise Operators, Membership Operators, Identity Operators, Expressions and order of evaluations. Control Flow-: if, if-elif-else, for, while, break, continue, pass. [**6 HOURS]**

**UNIT-II**

**STRINGS AND DATA STRUCTURES**

Strings: Strings and text files, String manipulations: subscript operator, indexing, slicing a string; strings and number system: converting strings to numbers and vice versa. Binary, octal, hexadecimal numbers, Data Structures: Lists, tuples, and dictionaries; basic list operators, replacing, inserting, removing an element; searching and sorting lists; dictionary literals, adding and removing keys, accessing and replacing values; traversing dictionaries. **[8 HOURS]**

**UNIT-III**

**FUNCTIONS AND OOPS**

Defining Functions, Calling Functions, Passing Arguments, Keyword Arguments, Default Arguments, Variable-length arguments, Anonymous Functions, Fruitful Functions (Function Returning Values), Scope of the Variables in a Function - Global and Local Variables.

**Classes and Objects**: Introduction, classes and objects, class method and self-argument, init() method, class and object variables ,public and private data members, private methods, calling a class method from another class method, built-in class attributes class and static methods, Inheritance: Introduction, inheriting classes in python, types of inheritance. [**8 HOURS]**

**UNIT-IV**

Permutations and combinations of list, Pow(x,n) in log n, Modulo arithmetic - maximum subarray sum modulo M, Factorization - number of factors, sum of factors , nCr mod M using Fermat little theorem, Sieve of erasthones, GCD in logN, Nth Fibonacci number using matrix exponentiation, Binary search - Aggressive cows(SPOJ), Disjoin set union, String matching - KMP, Stack - Next greater element, Priority queue - Merge K sorted arrays, Median in running stream of integers. . [**10 HOURS]**

**UNIT-V**

**Dynamic Programming:** Dynamic Programming strategies, understanding DP approaches One dimensional DP- Maximum value contiguous subsequence ,Maximum sum subarray with constraint, House Robbing , SSS restaurants , Gas stations, Rod cutting problem, Longest increasing subsequence [LIS] with constraint , Building bridges, Finding optimal number of jumps, Frog river crossing.(Textbook -2)

Two dimensional DP: Climbing n stairs with taking only 1, 2 or 3 steps, Longest common subsequence, Computing a binomial coefficient: n choose k , Painting colony houses with k colors, Counting Boolean parenthesizations, Edit distance, Apple count, Apple count variant with 3 ways of reaching a location, Maximum size sub-matrix with all 1's, Bombing enemies.(Textbook-2) . [**10 HOURS]**

## **[10 HOURS]**

**Text Books:**

## Reema Thareja, “Python Programming Using Problem Solving Approach”, Oxford University Press,2019.

1. Karumanchi, Narasimha. Algorithm Design Techniques: Recursion, Backtracking, Greedy, Divide and Conquer, and Dynamic Programming. India, CareerMonk Publications, 2018.
2. **Reference Books:**
3. [Charles Severance, " Python for Informatics- Exploring Information", 1st edition Shroff Publishers,2017.
4. Guide to Competitive Programming: Learning and Improving Algorithms Through Contests (Undergraduate Topics in Computer Science)
5. Competitive Programming in Python: 128 Algorithms to Develop Your Coding Skills Book by Christoph Dürr and Jill-Jênn Vie
6. Core Python Programming - Covers Fundamentals to Advanced Topics Like OOPS, Exceptions, Data Structures, Files, Threads, Networking, GUI, DB Connectivity and Data Science Second Edition (English, Paperback, Rao R. Nageswara)

**E-Books:** https://www.python.org/doc/

**NPTEL/MOOC:**

https://cses.fi/book/index.php

Leetcode.com