

Year/Semester	I B. Tech/I Sem	L	T	P	C
Regulation Year	2020-21	3	0	0	3
Subject	Computational Thinking and Programming				

Course Objectives:

1. To teach problem solving through Algorithms and Flowcharts
2. To elucidate problem solving through Python programming language
3. To train in the development of solutions using modular concepts
4. To explain the role of data structures in programming
5. To introduce object oriented programming paradigm through Python

UNIT – I: Knowing the Computer

Definition and Block Diagram of a Computer. Basic parts of a computer (Memory, CPU, Input, and Output), Memory hierarchy, Circuits and Logic, Hardware vs Software, Representation of Data in memory (integer (including negative), floating points etc. to text, images, audio and video), Principle of Abstraction, Operating System, Language Hierarchy - Machine Language to High Level Language, Compiler, Interpreter, The Command Line Interface (basic Linux commands)

UNIT – II: Computational Thinking and Introduction to Python

Simple logic building through flowcharting. Flowchart symbols, conditional and repetition blocks. Computational Thinking, Algorithm, Pseudocode, Time/Space complexity. Only Big O notation. Basic structure of a Python program, Elements of Python programming Language: token, literals, identifiers, keywords, expression, type conversions, Numbers, Variables, Input/Output statements, basic data types. Operators and their types and precedence, expressions. Control structures in Python - conditionals and loops

UNIT – III: Python Data Structures and Modularization

List and List Operations, Using Lists to represent Matrices, Strings, String operations, Tuples, Dictionaries, Sets, Iterators and generators, comprehensions.

Basic math functions, User defined Functions, parameters to functions, positional, keyword and default arguments, Lambda Functions, recursion. Packages, modules and namespaces.

UNIT-IV: File Handling

Files, Types of Files, Creating and Reading Text Data, File Methods to Read and Write Data, Reading and Writing Binary Files, The Pickle Module, Reading and Writing CSV Files, Python os and os.path Modules

UNIT – V: Object Oriented Programming

Object Oriented Design. Classes and Objects. Polymorphism, Abstraction, Inheritance, Encapsulation. Constructors. Function and operator overloading. Exception Handling.

Course Outcomes: Student should be able to

1. Understand the working principles of various components of a computer
2. Develop computational thinking and be able to use Python constructs to solve basic problems
3. Understand modularization and data structures concepts in Python
4. Apply file handling concepts in problem solving
5. Solve Real world problems by applying Object Oriented Concepts

Text Book:

1. Think Python: How to Think Like a Computer Scientist , Allen B. Downey, 2nd Edition
(<https://www.greenteapress.com/thinkpython/thinkCSpy.pdf>)

Reference Books:

1. Core python programming, W Chun PHI
(http://emixam.sevla.free.fr/books/2.PythoProg_softarchive.net.pdf)
2. Python programming a modern approach, Vamsi Kurama, pearson

Web Resources:

1. <http://www.ict.ru.ac.za/Resources/cspw/thinkcs3/thinkcs3.pdf>
2. <https://snakify.org>