INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

1. **Name of the Academic Unit:** Computer Science & Engineering

2. **Subject Name:** Programming Paradigms for Computation
   L-T-P: 3-1-0 **Credits:** 4

3. **Pre-requisites:** None

4. **Syllabus and reference books:**

Syllabus:

1. Introduction to Models of Computation: Basics of Machine Models and Data Models. Data Structures and Algorithms. Computational Modeling of Real-world Phenomena. Computational Simulations and Learning. Modular Programming using Python.
2. Problem Solving and Algorithms: Concept of an Algorithm – Termination and Correctness. Algorithms to Programs – Problem Specification, Decomposition and Refinement. Program Design Issues – Structured Programming, Data and Control Abstractions. Numerical Analysis Problems.
3. Functional Paradigm: Abstractions with Functions, Designing Functions as Abstractions, Higher-order functions, Variable Scoping, Recursions.
4. Abstractions with Data: Data Abstraction, Data Sequences, Temporal and Spatial Data, Data Manipulation using Python Tuples, Strings and Iterators.
5. Introduction to Algorithmic Complexity: Efficiency Issues in Programming; Time and Space Measures. Illustration with Searching and Sorting Algorithms, Experiments with Running Time of Algorithms.
6. Object Oriented Paradigm: Object Oriented Programming – Classes and Objects. Object Abstraction, Inheritance. Recursive Objects – Linked Lists, Trees and Graph Model, Object Oriented Programming using Python.
7. Data Representation and Visualization: Working with Experimental Data, Loading Data and Basic Visualization Techniques in Python.
8. Numerical Computations: Simulations, Interpolations, Optimizations. Solving Various Systems of Equations. Aspects of Error/Accuracy, Convergence, Efficiency, etc., Estimation Metrics and Graphical Visualization using Python.
9. Introduction to Machine Learning Paradigm: Linear Classification and Regression, Plotting Error/Accuracy and Convergence.
10. Logic Programming Paradigm: Introduction to Declarative Programs and Logic Programming. Goal/Constraint Satisfaction, Backtracking, Unification, Logic Programming with Prolog.
11. Advanced Computing Paradigms: Parallel Computing, Quantum Computing.

Reference Books:

1) John V. Guttag; "Introduction to Computation and Programming Using Python"; 3rd Edition, The MIT Press, 2021 (Python Version).
2) Harold Abelson, Gerald Jay Sussman; "Structure and Interpretation of Computer Programs"; 2nd Edition, The MIT Press, 1996 (Scheme Version)
3) John DeNero; "Composing Programs"; Python3 Programming Language Version of SICP Book. URL: https://www.composingprograms.com/
4) Alfred V. Aho and Jeffrey D. Ullman; "Foundations of Computer Science"; 1st Edition, W. H. Freeman & Co Ltd., 1994. URL: http://infolab.stanford.edu/~ullman/focs.html
5) Additional NOTES and SLIDES Prepared by the Instructors.

5. Lecture-wise break-up:

Programming Paradigms for Computation			
Serial No	**Topics**	**Hours**	
		L	**T**
Lecture 1	**Introduction to Models of Computation:** Basics of Machine Models and Data Models. Data Structures and Algorithms. Computational Modeling of Real-world Phenomena. Computational Simulations and Learning.	**3**	
Tutorial 1	**Introduction to Python Programming Language:** Branching and Iteration, Modular Programming.		**1**
Lecture 2	**Problem Solving and Algorithms:** Concept of an Algorithm - Termination and Correctness. Algorithms to Programs - Problem Specification, Decomposition and Refinement. Program Design Issues - Structured Programming, Data and Control Abstractions.	**3**	
Tutorial 2	**Example Problem Solving:** Numerical Analysis Algorithm - Root Finding (Greedy vs Bisection method).		**1**
Lecture 3	**Functional Paradigm**: Abstractions with Functions. Designing Functions as Abstractions. Higher-order Functions, Recursions.	**5**	
Tutorial 3	**Functions using Python Programming:** Variable Scoping; Higher-order Functions.		**2**

Lecture 4	**Abstractions with Data:** Data Abstraction, Data Sequences, Temporal and Spatial Data.	**2**	
Tutorial 4	**Data Manipulation using Python Programming:** Tuples, Strings and Iterators.		**1**
Lecture 5	**Introduction to Algorithmic Complexity:** Efficiency Issues in Programming; Time and Space Measures. Illustration with Searching and Sorting Algorithms.	**3**	
Tutorial 5	**Experiments with Running Time of Algorithms:** Plotting running times of Different Algorithms.		**1**
Lecture 6	**Object Oriented Paradigm**: Object Oriented Programming - Classes and Objects. Object Abstraction, Inheritance. Recursive Objects – Linked Lists, Trees, and Graph Model.	**5**	
Tutorial 6	**Object Oriented Programming using Python:** Python Objects, Classes, Inheritance, Instances. Python Trees, Linked Lists, Graphs.		**2**
Lecture 7	**Data Representation and Visualization:** Working with Experimental Data.	**3**	
Tutorial 7	**Data Visualization using Python:** Loading Data and Basic Visualization Techniques in Python.		**1**
Lecture 8	**Numerical Computations:** Simulations, Interpolations, Optimizations. Solving Various Systems of Equations. Aspects of Error / Accuracy, Convergence, Efficiency, etc.	**5**	
Tutorial 8	**Implementation of Numerical Computations:** Estimation Metrics and Graphical Visualization.		**2**
Lecture 9	**Introduction to Machine Learning Paradigm:** Linear Classification and Regression.	**3**	
Tutorial 9	**Implementation of Linear Classification and Regression:** Plotting Error / Accuracy and Convergence.		**1**
Lecture 10	**Logic Programming Paradigm**: Introduction to Declarative Programs and Logic Programming. Goal / Constraint Satisfaction, Backtracking, Unification.	**3**	

Tutorial 10	**Logic Programming with Prolog:** Example Declarative Programs.		**1**
Lecture 11	**Advanced Computing Paradigms:** Parallel Computing, Quantum Computing.	**2**	
	Total number of Hours : 50	**37**	**13**