DEPARTMENT OF TECHNOLGY EDUCATION,IER
UNIVERSITY OF THE PUNJAB, LAHORE-PAKISTAN
Course Outline

Programme	BS Technology Education	**Course Code**	**BSTE308**	**Credit Hours**	3
Course TTitle	**Fundamentals of Programming and Computational Thinking**				

Course Introduction

This course provides a basic introduction to programming and computational thinking. Students will learn fundamental programming concepts, problem-solving techniques, and algorithm development. The course includes theoretical understanding and practical exercises to develop foundational skills in programming and computational thinking.

Learning Outcomes

On the completion of the course, the students will:

1. Understand basic programming concepts and terminology.
2. Develop simple algorithms to solve problems.
3. Write, test, and debug basic programs.
4. Apply computational thinking to break down complex problems.
5. Explore different programming paradigms and their applications.

Course Content		**Assignments/Readings**
Week 1	**Introduction to Programming** • **Unit 1.1**: Overview of Programming • **Unit 1.2**: History and Evolution of Programming Language	Reflective essay on the importance of programming in today's world
Week 2	**Basic Programming Concepts** • **Unit 2.1**: Variables and Data Types • **Unit 2.2**: Basic Input and Output Operations	Write a simple program to perform basic input and output operations
Week 3	**Control Structures** • **Unit 3.1**: Conditional Statements • **Unit 3.2**: Looping Constructs	Develop a program using conditional statements and loops

Week 4	**Functions and Procedures** • **Unit 4.1**: Introduction to Functions • **Unit 4.2**: Parameters and Return Values	Write a program using functions to perform a specific task
Week 5	**Arrays and Lists** • **Unit 5.1**: Understanding Arrays and Lists • **Unit 5.2**: Basic Operations on Arrays and Lists	Develop a program to manipulate arrays or lists
Week 6	**String Manipulation** • **Unit 6.1**: Basic String Operations • **Unit 6.2**: String Methods and Functions	Write a program to perform various string manipulations
Week 7	**Introduction to Algorithms** • **Unit 7.1**: What is an Algorithm? • **Unit 7.2**: Algorithm Development and Flowcharts	Create a flowchart for a simple algorithm
Week 8	**Problem-Solving Techniques** • **Unit 8.1**: Decomposition and Abstraction • **Unit 8.2**: Pattern Recognition and Generalization	Solve a problem using decomposition and abstraction techniques
Week 9	**Debugging and Testing** • **Unit 9.1**: Debugging Techniques • **Unit 9.2**: Testing and Validation	Write a program and debug any errors encountered
Week 10	**Introduction to Object-Oriented Programming** • **Unit 10.1**: Basic Concepts of Object-Oriented Programming • **Unit 10.2**: Classes and Objects	Develop a simple program using object-oriented principles
Week 11	**File Handling**	Write a program to perform

	• **Unit 11.1**: Reading from and Writing to Files • **Unit 11.2**: File Operations	basic file handling operations
Week 12	**Introduction to Recursion** • **Unit 12.1**: Understanding Recursion • **Unit 12.2**: Simple Recursive Algorithms	Develop a recursive program to solve a problem
Week 13	**Basic Data Structures** • **Unit 13.1**: Introduction to Data Structures • **Unit 13.2**: Stacks, Queues, and Linked Lists	Write a program using basic data structures
Week 14	**Introduction to Sorting and Searching** • **Unit 14.1**: Basic Sorting Algorithms • **Unit 14.2**: Basic Searching Algorithms	Implement and test a sorting algorithm
Week 15	**Advanced Topics in Programming** • **Unit 15.1**: Introduction to Advanced Programming Concepts • **Unit 15.2**: Exploring Different Programming Paradigms	Research and present on a specific advanced programming topic
Week 16	**Course Review and Final Assessment** • **Unit 16.1**: Review of Key Concepts and Themes • **Unit 16.2**: Comprehensive Final Exam	Develop a final project incorporating various programming concepts learned throughout the course

Textbooks and Reading Material

1. Textbooks.

- Python Crash Course by Eric Matthes

2. Suggested Readings

- Think Python: How to Think Like a Computer Scientist by Allen B. Downey

Teaching Learning Strategies
1. **Lectures:** To introduce and explain key concepts and theories. 2. **Hands-on Labs:** To provide practical experience with robotics components and programming. 3. **Assignments and Projects:** To reinforce learning and encourage application of concepts in real-world scenarios. 4. **Group Discussions:** To facilitate peer learning and collaborative problem-solving.

Assessment

Sr. No.	Elements	Weight age	Details
1.	Midterm Assessment	35%	Written Assessment at the mid-point of the semester.
2.	Formative Assessment	25%	Continuous assessment includes: Classroom participation, assignments, presentations, viva voce, attitude and behavior, hands-on-activities, short tests, projects, practical, reflections, readings, quizzes etc.
3.	Final Assessment	40%	Written Examination at the end of the semester. It is mostly in the form of a test, but owing to the nature of the course the teacher may assess their students based on term paper, research proposal development, field work and report writing etc.