

Mini Project 2: Programming Paradigm Exploration

Objectives

- Understand the differences between **programming paradigms** (imperative, functional, logic, object-oriented).
- Explore how the **same problem** can be expressed differently in multiple languages.
- Develop skills in comparing **syntax, semantics, readability, and expressiveness** of languages.

Description

Students (a team of 3 members maximum) will select a single computational problem and implement it using **three** different programming paradigms. The purpose is not to produce the most efficient code, but to highlight differences in style, abstraction, and reasoning.

Suggested problems (choose one or propose your own):

1. **Data Structure Processing:** List sorting, binary tree traversal, graph search (DFS/BFS).
2. **Practical Task:** A simple booking system, string pattern matching, database query, mini expert system, scheduling.
3. **Puzzles:** Sudoku,

Implementation Requirements

- **Object-Oriented Language (e.g., C++, Java):** Encapsulate data and behavior in classes.
- **Functional Language (e.g., ML, Haskell, Scheme):** Use recursion, higher-order functions, immutability.
- **Logic Language (e.g., Prolog):** Use facts, rules, and queries to express the solution.

Analysis and Report

Each group must write a report that includes:

1. **Source Code:** Full code in each paradigm with comments.
2. **Comparison of Expressiveness:**
 - How easy/hard was it to implement in each paradigm?
 - How much code was required?

- Which paradigm feels more natural for this problem?
- 3. **Semantics & Abstraction:**
 - What concepts from the paradigm influenced your design (e.g., recursion, backtracking, loops, classes)?
 - How do variables and memory behave differently across paradigms?
- 4. **Reflection:**
 - Which paradigm was most intuitive for you?
 - Which one would you recommend for this type of problem in real-world projects, and why?

Deliverables

1. Source code in all chosen languages.
2. A written report (8–10 pages).
3. A 15-minute presentation showing the analysis and demonstrating how each program runs.

Assessment Criteria

- Correctness of implementations (30%)
- Quality of comparative analysis (30%)
- Depth of discussion on paradigms (20%)
- Clarity of report & presentation (20%)