



Programming Paradigms Report

Introduction

- This report investigates three programming paradigms: Procedural, Object-Oriented (OOP), and Event-Driven programming. Each paradigm has unique characteristics, use cases, and advantages.

Procedural Programming

- - Follows a step-by-step approach using functions and procedures.
- - Code is structured into blocks or routines.
- - Focuses on process and logic flow.
- - Used in applications like simple scripts, system utilities, and mathematical computations.
- Example: C, Python (using functions), Pascal.

Object-Oriented Programming (OOP)

- - Based on objects, which are instances of classes.
- - Uses concepts like encapsulation, inheritance, and polymorphism.
- - Helps in code reusability and scalability.
- - Used in applications like GUI development, game development, and large-scale software systems.
- Example: Java, C++, Python (OOP style).

Event-Driven Programming

- - Program execution is determined by user actions or system events.
- - Uses event listeners, handlers, and callbacks.
- - Commonly used in UI-based applications, web development, and gaming.
- Example: JavaScript (browser events), Visual Basic, C# (Windows Forms).

Relationship Between Paradigms

- - Procedural programming provides the foundation for structured coding.
- - OOP builds upon procedural programming by introducing objects and reusability.
- - Event-driven programming often incorporates OOP and procedural principles to handle user interactions and events.

Conclusion

- Each programming paradigm serves a specific purpose. Understanding their differences helps in choosing the right approach for different types of software development projects.