# Tech Programming - Comprehensive Notes

## Introduction to Software Development

- Software Development Lifecycle: Understand the stages from requirement gathering to maintenance.
- Agile and Waterfall Methodologies: Key differences and applications.
- Tools: IDEs, version control systems, and collaborative platforms.

```
Data Flow Diagram (DFD):
- Represents data flow in the system.
- Example here.
```

## Programming Fundamentals

- Basic Concepts: Variables, data types, control structures, and functions.
- Error Handling: Importance and techniques.
- Modular Programming: Breaking problems into smaller units.

## Frontend Development

- HTML, CSS, JavaScript: Foundations of web development.
- Responsive Design: Media queries and frameworks like Bootstrap.
- Modern Frameworks: React, Angular, or Vue introduction.

## Backend Development

- Server-side Programming: Node.js, Django, or Flask basics.

- RESTful APIs: Importance in modern applications.

- Database Integration: Connecting and querying SQL/NoSQL databases.

UML Diagram:
- Illustrates system architecture.
- Example here.

## Database Management

- Relational Databases: Design and normalization concepts.

- Non-Relational Databases: Flexible data handling with MongoDB, Firebase.

- CRUD Operations: Create, Read, Update, and Delete basics.

## Version Control and Collaboration

- Git Basics: Branching, merging, and pull requests.

- GitHub Workflow: Collaborative coding techniques.

- CI/CD: Automating builds and tests.

## Testing and Debugging

- Types of Testing: Unit, integration, and system testing.

- Debugging Tools: Popular tools and strategies.

- Test-driven Development (TDD): Writing tests before code.

**Software Design and Architecture**

- Design Patterns: Common patterns and their applications.

- Microservices Architecture: Dividing systems into independent services.

- Scalability and Optimization: Tips for handling large-scale systems.

**DevOps and Deployment**

- DevOps Culture: Bridging development and operations.

- Containerization: Docker fundamentals.

- Cloud Services: AWS, Azure, and Google Cloud basics.

**Capstone Project**

- End-to-End Project: Implementing a full software solution.

- Real-world Problem Solving: Tackling industry-relevant issues.

- Documentation: Best practices for maintaining project clarity.