

Chapter 1: Concepts and Principles

You will learn:

- the difference between analysis and design.
- the difference between functional and OO approach to design.
- OO terminology.
- how to define an object and its characteristics.
- the tools of OOD, OOA and OO programming.
- how objects interact.
- to appreciate relationships between objects.
- the concepts of actors, servers and agents.
- inheritance, aggregation, association and instantiation.

Chapter 2: Object-Oriented Design

You will learn:

- the concepts of cohesion and coupling.
- difference between information and functional cohesion in system design.
- the use of encapsulation in OOD.
- how to recognize the practical usage of an abstract data type.
- what a class is and its components including constructor, destructor, and members.
- how to create and utilize the advantages associated with inheritance.
- polymorphism and where it is used in OOD and OOP.

Chapter 3: UML Concepts

You will learn:

- the purpose and features of UML.
- the background of UML.
- the components of UML.
- how to understand and recognize class relationships with UML.
- UML diagrams.
- the different UML diagrams and their functions.
- to appreciate the UML philosophy.
- purpose of modeling.
- how to extend UML.

Chapter 4: Software Development Methodologies

You will learn:

- Waterfall model.
- Spiral model.
- process and activities.
- iteration.
- refactoring.
- interative and incremental development.
- RUP: Rational Unified Process.
- key principles of business-driven development.
- artifacts by phase.

Chapter 5: Gathering Requirements

You will learn:

- the requirements phase.
- modeling the business context and system functionality.
- how to record the system requirements on a complete use case model.

Chapter 6: Analysis

You will learn:

- what analysis is.
- how to build a static analysis model.
- the use of dynamic models.

Chapter 7: System Design

You will learn:

- the steps in system design and how a system is decomposed into physical and logical components.
- concurrency and security issues.
- how to partition a system.
- how software design may be represented as a set of interacting objects which manages its own state and operations.
- the activities in the object-oriented design process.
- the models that describe an object-oriented design.
- how the UML can be used to represent object-oriented design models.

Chapter 8: Design Patterns

You will learn:

- how to identify the appropriate design pattern for a given scenario.
- the benefits of using design patterns.

