SCALA

1: Introduction to Scala Programming

1.1. Getting Started with Scala

- 1.1.1. What is Scala?
- 1.1.2. Why Scala?
- 1.1.3. Installing Scala
- 1.1.4. Setting up Development Environment

1.2. Scala Basics

- 1.2.1. Variables and Data Types
- 1.2.2. Expressions and Operators
- 1.2.3. Control Structures (if, else, while, for)
- 1.2.4. Functions and Methods
- 1.2.5. Working with Collections (Arrays, Lists, Sets)

2: Object-Oriented Programming in Scala

2.1. Classes and Objects

- 2.1.1. Defining Classes
- 2.1.2. Constructors
- 2.1.3. Companion Objects

2.2.Inheritance and Polymorphism

- 2.2.1. Inheritance
- 2.2.2. Overriding Methods
- 2.2.3. Abstract Classes and Traits

2.3.Pattern Matching

- 2.3.1. Pattern Matching Syntax
- 2.3.2. Case Classes
- 2.3.3. Pattern Matching in Collections

3: Functional Programming in Scala

3.1.Introduction to Functional Programming

- 3.1.1. Functional vs. Imperative Programming
- 3.1.2. Pure Functions and Immutability
- 3.1.3. Higher-Order Functions

3.2.Working with Functions

- 3.2.1. Function Literals and Anonymous Functions
- 3.2.2. Closures
- 3.2.3. Currying and Partially Applied Functions

3.3.Collections and Functional Programming

- 3.3.1. Map, Filter, and Reduce
- 3.3.2. For Comprehensions
- 3.3.3. Option and Try for Error Handling

4: Concurrency and Parallelism

4.1.Introduction to Concurrency

- 4.1.1. Threads and Processes
- 4.1.2. Concurrency vs. Parallelism
- 4.1.3. Thread Safety

4.2.Scala's Approach to Concurrency

- 4.2.1. Futures and Promises
- 4.2.2. Actors and Message Passing
- 4.2.3. Thread Pools and Execution Contexts

5: Advanced Scala Topics

5.1.Type System

- 5.1.1. Type Inference
- 5.1.2. Type Bounds and Variance
- 5.1.3. Structural Types

5.2.Implicit Conversions and Parameters

- 5.2.1. Implicit Conversions
- 5.2.2. Implicit Parameters
- 5.2.3. Implicit Classes

5.3.Using Libraries and Frameworks

- 5.3.1. Using Akka for Distributed Systems
- 5.3.2. Working with Play Framework for Web Development

6: Scala Best Practices and Advanced Topics

6.1.Error Handling and Exception Handling Best Practices

- 6.1.1. Try, Success, and Failure
- 6.1.2. Dealing with Exceptions

6.2.Testing and Debugging Scala Code

- 6.2.1. Unit Testing with ScalaTest
- 6.2.2. Debugging Scala Applications

6.3.Performance Optimization

- 6.3.1. Profiling and Benchmarking

- 6.3.2. Optimizing Code

Module 7: Scala Projects and Real-World Applications

7.1.Building a Scalable Web Application

- 7.1.1. Setting up a Web Application Project
- 7.1.2. Integrating with Databases
- 7.1.3. RESTful API Development

7.2.Distributed Systems with Akka

- 7.2.1. Building a Distributed Application
- 7.2.2. Message Passing and Supervision

7.3.Data Analysis and Processing with Scala

- 7.3.1. Using Spark for Big Data Processing
- 7.3.2. Analyzing Data Sets

8: Final Project and Course Review

8.1.Capstone Project

- 8.1.1. Project Proposal and Planning
- 8.1.2. Implementation
- 8.1.3. Testing and Deployment

8.2.Course Review and Certification

- 8.2.1. Recap of Key Concepts
- 8.2.2. Assessment and Certification