

# Training Courses

--Rushi Desai ([rushidesai1@gmail.com](mailto:rushidesai1@gmail.com))

---

## Summary of courses:

- [Build Tools \(Maven\)](#)
- [Version Control \(Git\)](#)
- [Core Java](#)
- [Advanced Java \(Depends on Core Java\)](#)
- [JEE \(Depends on Advanced Java\)](#)
- [Spring](#)
- [Hibernate](#)
- [Groovy \(Depends on Java\)](#)
- [Grails \(Depends on Groovy\)](#)

## Notes:

- The contents listed may not be taught in the same order
- The materials that needs to be covered in class needs to be discussed. Depending upon:
  - I. Time
  - II. Student Level
  - III. Objective etc. Course can be curtailed. Mixture of different courses is also an option.

## Details

### Core Java

1. Data Types
2. Variables: Scope & Lifetime
3. Evolution of Class primary data types:
  - Arrays
  - Structs
  - Class
4. Class: State and behavior
5. Operators
6. Flow Control
7. OOPs !! : Inheritance, Encapsulation, Polymorphism
8. Closer look at Class and Methods
9. Collections
10. Exception Handling
11. Enums
12. Generics
13. MultiThreading
14. Jar, Executable Jar
15. Annotations

### Testing Frameworks

1. Mockito
2. PowerMockito

3. Spock (Groovy and Java) (Newest on the block!!)

## Advanced Java (Enterprise web application developement and JEE)

1. Http request-response cycle
2. Servlets
3. Jsp's
4. Stress, Integration, Unit, Testing, mocking
5. REST, Web and Business Tier Java EE Security
6. Webservices:
  - i. JAXRS
  - ii. (REST / JSR311)
  - iii. JAXWS
  - iv. : JSR 224: JavaTM API for XMLBased
  - v. Web Services (JAXWS) 2.0
7. JAXB : JSR 222: JavaTM Architecture for XML Binding (JAXB) 2.0
8. Context and Dependency Injection (CDI / JSR299)
9. Enterprise Java Beans 3.1 (EJB / JSR318)
10. Dependency Injection for Java (JSR330)
11. and also Resource Injection
12. Bean Validation (JSR303)
13. Java Persistence API JPA (JSR317)
14. Database:
  - i. JDBC
  - ii. Database drivers
  - iii. Transactions
  - iv. Isolation Levels
  - v. Locking
  - vi. Relational Sql and Database modelling
15. Java Connector Architecture
16. JNDI
17. DataSource and Connection Pools
18. Internationalizing and Localizing Web Applications
19. JMS
20. Security:
  - i. Basic authentication
  - ii. Formbased
  - iii. authentication
  - iv. Digest authentication
  - v. Client authentication
  - vi. Mutual authentication
21. WAR, EAR

## Build tools

1. Maven
2. Gradle (Newest on the block !) (Course taught via gradle)

## IDE

1. IntelliJ Idea

## Version Control

1. Git (Course taught via git)
2. SVN

## Spring

1. Spring Core:
  - i. Spring DI (IOC)
    - a. Container
    - b. Bean
    - c. Dependencies
      - i. Constructor Injection
      - ii. Gettersetter
      - iii. Injection
      - iv. Factory Method
      - v. Collections
      - vi. Lazy Init
    - d. Bean Scopes
    - e. Customizing the nature of a bean
      - i. Lifecycle callbacks
        1. Initialization callbacks
        2. Destruction callbacks
        3. Default initialization and destroy methods
        4. Combining lifecycle mechanisms
        5. Startup and shutdown callbacks
        6. Shutting down the Spring IoC container gracefully in non-web applications
      - b. ApplicationContextAware and BeanNameAware
      - c. Other Aware interfaces
  - ii. Profiles
  - iii. Message Source
  - iv. Annotations
  - v. Using JSR 330 Standard Annotations
  - vi. Validation, Data Binding, and Type Conversion
  - vii. Spring Expression Language (SpEL)
  - viii. Resources
  - ix. Testing
  - x. Spring DataAccess
2. Spring AOP
  - a. Aspect Oriented Programming with Spring
3. Spring Security
4. Spring ORM
5. Spring MVC
6. SpringWS
7. Spring JMS
8. Spring Email

## Hibernate

1. Domain Model
  - a. Mapping Types
  - b. Entity Types
2. Naming
  - a. Implicit Strategies
  - b. Physical Strategies

- c. Basic Types
  - d. Embeddable types
  - e. Entity Types
  - f. Identifiers
  - g. Associations
  - h. Collections
  - i. Natural Id's
  - j. Dynamic Model
  - k. Inheritance
- 3. Bootstrap
  - a. JPA
  - b. Native
- 4. Persistence Contexts
- 5. Flushing
- 6. Database Access
- 7. Transactions and concurrency control
- 8. JNDI
- 9. Locking
- 10. Fetching
- 11. Batching
- 12. Caching
- 13. Interceptors and Events
- 14. HQL and JPQL
- 15. Criteria
- 16. Native SQL Queries
- 17. Spatial
- 18. Multitenancy

## Groovy (Language)

- 1. Groovy vs Java
- 2. Groovy Strings
- 3. Groovy Operators
- 4. Groovy Collections
- 5. Groovy Script
- 6. Groovy Duck typing
- 7. Groovy Constructors
- 8. Groovy Methods
- 9. Fields and Properties
- 10. Closures
  - i. Owner, delegate and this
  - ii. Delegation strategy
- 11. Groovy Annotations
- 12. Generics
- 13. traits
- 14. Multiple Inheritance
- 15. Chaining
- 16. SAM Coersions
- 17. AST transforms
- 18. Closure coercion
- 19. Functional programming
  - i. Currying
  - ii. Left currying
  - iii. Right currying

- iv. Index based currying
  - v. Memoization
  - vi. Composition
  - vii. Trampoline
- 20. GPath
- 21. Groovy Truth
- 22. CompileStatic, CompileDynamic
- 23. Type inference
- 24. Closures and type inference
- 25. Metaprogramming
  - i. Runtime metaprogramming
    - a. GroovyObject interface
    - b. invokeMethod
    - c. get/setProperty
    - d. get/setMetaClass
    - e. get/setAttribute
  - ii. methodMissing
  - iii. propertyMissing
  - iv. GroovyInterceptable
  - v. ExpandoMetaClass
- 26. Compile-time metaprogramming
  - 1. Available AST transformations
  - 2. Code generation transformations
- 27. Processing JSON
- 28. Processing XML
- 29. GPath
- 30. Template engines

## Grails 2 and Grails 3

- 1. The Web Layer
  - i. Controllers
    - a. Understanding Controllers and Actions
    - b. Controllers and Scopes
    - c. Models and Views
    - d. Redirects and Chaining
    - e. Data Binding
    - f. XML and JSON Responses
    - g. More on JSONBuilder
    - h. Uploading Files
    - i. Command Objects
    - j. Handling Duplicate Form Submissions
    - k. Simple Type Converters
    - l. Declarative Controller Exception Handling
  - ii. GSP
  - iii. TagLibs
  - iv. Url Mappings
  - v. Interceptors (In grails 3 only)
  - vi. Filters (In Grails 2 only)
- 2. Traits
- 3. Webservices
  - i. REST (In Grails 3 only)
  - ii. JSON Views (In Grails 3 only)
- 4. Asynchronous Programming

- i. Promises
  - ii. Events
    - i. 1. Consuming Events
    - ii. Event Notification
    - iii. Reactor Spring
  - iii. Annotations
  - iv. Events from GORM
  - v. Events from Spring
  - vi. Asynchronous GORM
  - vii. Asynchronous Request Handling
  - viii. Servlet 3.0 Async
- 5. Validation
- 6. The Service Layer
- 7. GORM and Hibernate
- 8. Testing
  - i. Unit Testing
    - a. Unit Testing Controllers
    - b. Unit Testing Tag Libraries
    - c. Unit Testing Domains
    - d. Unit Testing Filters
    - e. Unit Testing URL Mappings
    - f. Mocking Collaborators
    - g. Mocking Codecs
    - h. Unit Test Metaprogramming
    - i. Integration Testing
  - ii. Functional Testing
- 9. Grails plugin
- 10. Grails Spring Security (via plugin)