

Training Courses

--Rushi Desai (rushidesai1@gmail.com)

Summary of courses:

- [Core Java](#)
- [Advanced Java \(Depends on Core Java\)](#)
- [JEE \(Depends on Advanced Java\)](#)
- [Build Tools](#)
- [IDE](#)
- [Version Control](#)
- [Spring](#)
- [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 behaviour
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. JAX-RS (REST / JSR-311)
 - ii. JAX-WS : JSR 224: JavaTM API for XML-Based Web Services (JAX-WS) 2.0
7. JAX-B : JSR 222: JavaTM Architecture for XML Binding (JAXB) 2.0
8. Context and Dependency Injection (CDI / JSR-299)
9. Enterprise Java Beans 3.1 (EJB / JSR-318)
10. Dependency Injection for Java (JSR-330) and also Resource Injection
11. Bean Validation (JSR-303)
12. Java Persistence API JPA (JSR-317)
13. Database:
 - i. JDBC
 - ii. Database drivers
 - iii. Transactions
 - iv. Isolation Levels
 - v. Locking
 - vi. Relational Sql and Database modelling
14. Java Connector Architecture
15. JNDI
16. DataSource and Connection Pools
17. Internationalizing and Localizing Web Applications
18. JMS
19. Security:
 - i. Basic authentication
 - ii. Form-based authentication
 - iii. Digest authentication
 - iv. Client authentication
 - v. Mutual authentication
20. 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
 - a. Constructor Injection
 - b. Getter-setter Injection
 - c. Factory Method
 - d. Collections
 - e. Lazy Init
 - d. Bean Scopes
 - e. Customizing the nature of a bean
 - a. Lifecycle callbacks
 - a. Initialization callbacks
 - b. Destruction callbacks
 - c. Default initialization and destroy methods
 - d. Combining lifecycle mechanisms
 - e. Startup and shutdown callbacks
 - f. 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 Data-Access
2. Spring AOP
 - i. Aspect Oriented Programming with Spring
3. Spring Security
4. Spring ORM
5. Spring MVC
6. Spring-WS
7. Spring JMS
8. Spring Email

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 1. Promises 2. Events 1. Consuming Events 2. Event Notification 3. Reactor Spring Annotations 4. Events from GORM 5. Events from Spring 3. Asynchronous GORM 4. Asynchronous Request Handling 5. 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
 - ii. Integration Testing
 - iii. Functional Testing
 - 9. Grails plugin
 - 10. Grails Spring Security (via plugin)