This is a 24-week course.
Core Java (Week 1–5)
Review all the code, create small projects(Week 6)
JDBC(Week 7)
Servlets and JSP (Week 8)
Maven (Week 9)
Hibernate (Week 10–11)
Spring Framework (Week 13)
Spring Boot (Week 14)
Spring Data JPA (Week 15)
Spring MVC (Week 16)
Spring AOP (Week 17)
Spring REST (Week 18)
Spring Security (Week 19)

Docker (Week 21)
Cloud Deployment (Week 22)
Microservices (Week 23–24)

1. Practical Work (20 minutes)
Use the remaining 20 minutes for hands-on coding exercises or practical applications related to the day's topic. This can involve writing code, debugging, or working through small projects.
2. Friday Flexibility
Utilize Fridays for review, deeper practice, or catching up on content if needed. This could mean focusing more on practical work or revisiting complex topics from the video.

Show less

@brianlevi4964 7 months ago

Basic Java Concepts:

- 00:13:21 First code in java
- 00:24:35 How Java Works
- 00:36:34 Variables in java
- 00:48:11 Data types in java
- 01:00:28 Literal
- 01:04:49 Type conversion
- 01:17:20 Assignment Operators
- 01:27:32 Relational Operators
- 01:35:35 Logical Operators

Control Flow:

- 01:46:47 If Else
- 01:59:45 If Else if
- 02:05:18 Ternary
- 02:09:51 Switch Statement

Loops:

- 02:17:50 Need For Loop
- 02:21:12 While Loop
- 02:33:35 Do While Loop
- 02:36:48 For Loop
- 02:50:06 Which Loop To Use

Classes and Objects:

- 02:51:48 Class And Object Theory
- 02:57:30 Class and Object Practical
- 03:13:03 JDK JRE JVM
- 03:18:22 Methods
- 03:29:44 Method Overloading
- 03:35:41 Stack And Heap

Arrays:

- 03:48:12 Need of an Array
- 03:52:01 Creation of Array
- 03:59:28 Multi Dimensional Array
- 04:12:30 Jagged and 3D Array
- 04:18:08 Drawbacks of Array
- 04:20:54 Array of Objects
- 04:29:42 Enhanced for loop

Strings:

- 04:35:07 What is String
- 04:42:24 Mutable vs Immutable string
- 04:48:43 StringBuffer and StringBuilder

Static Members:

- 04:54:23 Static variable
- 05:01:26 Static block
- 05:08:43 Static method

Object-Oriented Programming (OOP) Concepts:

- 05:13:25 Encapsulation
- 05:25:04 Getters and setters
- 05:27:55 this keyword
- 05:37:36 Constructor
- 05:44:35 Default vs Parameterized Constructor
- 05:49:36 this and super method
- 06:01:42 Naming Convention
- 06:06:08 Anonymous Object

Inheritance:

- 06:10:51 Need of Inheritance
- 06:17:06 What is Inheritance
- 06:25:50 Single and Multilevel Inheritance
- 06:29:41 Multiple Inheritance
- 06:36:16 Method Overriding

Packages and Access Modifiers:

- 06:44:05 Packages
- 06:56:20 Access Modifiers

Polymorphism:

- 07:04:42 Polymorphism
- 07:08:31 Dynamic Method Dispatch
- 07:16:29 Final keyword
- 07:22:43 Object Class equals toString hashcode
- 07:34:41 Upcasting and Downcasting
- 07:41:17 Abstract keyword

Inner Classes:

- 07:53:26 Inner class
- 07:59:03 Anonymous Inner class
- 08:04:11 Abstract and Anonymous Inner class

Interfaces:

- 08:07:22 What is Interface
- 08:15:18 More on Interfaces
- 08:18:40 Need of Interface

Enum and Annotations:

- 08:27:11 What is Enum
- 08:34:04 Enum if and switch
- 08:37:59 Enum Class
- 08:45:45 What is Annotation
- 08:53:14 Functional Interface
- 08:56:43 Lambda Expression
- 09:02:51 Lambda Expression with return
- 09:06:08 Types of Interface

Exception Handling:

- 09:10:41 What is Exception
- 09:15:57 Exception Handling using try-catch
- 09:21:58 Try with multiple catch
- 09:32:14 Exception Hierarchy
- 09:36:30 Exception throw keyword
- 09:42:05 Custom Exception
- 09:45:35 Handling Exceptions using throws
- 09:55:29 User Input using BufferedReader and Scanner
- 10:07:17 Try with resources

Multithreading:

- 10:15:25 Threads
- 10:20:37 Multiple Threads
- 10:31:58 Thread Priority and Sleep
- 10:39:20 Runnable vs Thread
- 10:47:45 Race Condition
- 11:00:15 Thread states

Collections:

- 11:03:45 Collection API
- 11:08:30 ArrayList
- 11:20:23 Set
- 11:27:30 Map
- 11:37:32 Comparator vs Comparable

Advanced Java:
11:53:15 Need of Stream API
12:00:12 forEach Method
12:05:01 Stream API
12:14:09 Map Filter Reduce Sorted

13:13:08 - JUnit

16:04:39 - DSA

16:04:42 DSA Intro

16:11:50 Abstract Data Type
16:19:01 Arrays
16:26:53 Time Complexity

Searching:

Linear Search - 16:29:25
Binary Search - 16:33:31
Time complexity - 16:40:03
Time complexity for Linear Search - 16:41:48
Time complexity for Binary Search - 16:43:54

16:47:55 Linear Search & Binary Search

Code for Linear Search - 16:49:00
Code for Binary Search(Using While loop) - 16:55:12
Code for Binary Search(Using Recursion) - 17:03:46

Sorting:

17:07:10 Sorting Techniques

Sorting - 17:07:53

Bubble Sort - 17:09:20:
Code for Bubble sort - 17:14:55

17:23:05 Selection Sort:
Code for Selection Sort - 17:30:00

17:37:13 Insertion Sort:
Code for Insertion Sort = 17:44:33

17:57:26 Quick Sort
Example for Quick Sort - 18:04:06(logic) , 18:18:45 (code)

18:25:44 Merge Sort
Example for Merge Sort - 18:35:49(Logic), 18:41:55 (Code)

Linked List:

18:53:18 Linked List
Code for Linked List - 19:06:41
- Insert - 19:11:43
- Show - 19:19:37
- InsertAtStart - 19:27:10
- InsertAt - 19:30:12
- DeleteAt - 19:38:44

Stack:

19:43:58 Stack
Code for Stack(Fixed size array) - 19:52:58
- Push - 19:54:15
- Show - 19:57:45
- Pop - 19:58:35
- Peek - 20:01:10
- Size - 20:02:50
- IsEmpty - 20:04:08

Code for Stack(Dynamic size array) - 20:09:42

Queue:
20:22:41 Queue
Code for Queue - 20:27:28
- Enqueue - 20:28:15
- Show - 20:30:02
- Dequeue - 20:32:13
- Size - 20:42:23
- IsEmpty - 20:43:23
- IsFull - 20:44:16

Tree:
20:46:55 Tree
Tree - 20:47:05
Binary Tree - 20:49:26
Binary Search Tree - 20:55:06
Code for Binary Search Tree(insert) - 20:56:59
Insert - 20:57:56
Tree Traversal(inOrder) - 21:06:01
Tree Traversal(preOrder) - 21:11:07

20:55:03 Binary Search Tree

21:11:50 - Git
23:12:20 - JDBC

24:33:04 - Servlet and JSP

24:33:14 Servlet
26:11:17 JSP
27:32:19 JSTL

30:29:58 - Hibernate

30:29:58 - Introduction to Hibernate
30:32:13 - Prerequisites for Hibernate
30:34:48 - hibernate theory
30:44:32 - Hibernate practical
30:56:29 - How to add Hibernate Plugin in Eclipse
30:58:06 - Configuration File
31:05:21 - Working
31:09:58 - show sql Property
31:13:20 - Annotation
31:18:07 - Fetch data using Hibernate
31:22:39 - How to use Embeddable Object
31:30:31 - Mapping Relations Theory
31:43:35 - Mapping Relations Practical

33:49:37 - Rest API Web Service
33:23:20 - JPA

36:24:17 - Spring Framework

36:16:19 - Introduction to Spring
36:24:20 - Spring
36:26:16 - Sprint documentation
36:29:09 - Perquisites
36:31:21 - Software requirements
36:33:48 - STS Setup
36:39:19 - Dependency Injection in Spring
36:44:56 - Creating spring starter project
36:50:07 - Dependency Injection in Spring Boot
36:55:42 - Spring Boot Autowire
36:59:30 - Bean Factory
37:11:45 - Application Context
37:14:27 - Spring Container
37:20:34 - Singleton vs Prototype
37:23:31 - Setter Injection
37:32:11 - constructor Injection
37:36:18 - Autowire
37:44:51 - Primary

37:46:54 Spring JDBC

38:16:19 Spring MVC

Spring Boot MVC
(38:46:48) @RequestParam
(38:57:44) Model
(39:00:37) ModelMap vs Model
(39:01:27) @ModelAttribute as a parameter -> captures data from de view (typically a form) and insert it into a model attribute (object)
(39:11:55) @ModelAttribute as a method
(39:14:51) Spring MVC project -> configuration on a Spring Project

---

39:41:39 Spring ORM Theory

40:12:45 Spring Data JPA

40:41:13 Rest API using Spring Boot

40:41:17 REST

41:17:32 Project Using Spring Boot MVC

41:18:09 Java Project

43:45:47 Java Spring Boot MongoDB Full Project

45:01:07 Spring AOP

45:21:59 Spring Security

(45:34:18) Spring Security -> Login
- (45:38:17) How to create user and password in memory -> configuration file
- (45:44:51)How to fetch and store user and password in a db
- (45:49:37) Creating User class
- (45:50:42) Creating Service (UserDetailService) and setting password encoder
- (45:55:42) Creatig UserDetails implementation
- (46:01:02) Recap
- (46:02:37) BCrypt Password encoder
- (46:08:57) Customize login

46:28:34 Microservice