Java Programming

(by - Ravi Sir)

SUNDAY 30

Java Course for Interview Preparation

- Practical
- Completely Industry Oriented
- SE Career Oriented.

Memorable Event
Appointment
Phone No.

Why learn Java?

- Simple
  - Java has thrown out many of the complex features like C++ or C resulting in a simpler language
- Platform independent
- Better Community support
- Excellent tools: Netbeans, Eclipse, Intellij .... etc
- Better documentation
- Concurrent programming
- Network programming

Where Java is Used? (became Generic language)

↳ We can use it almost in order to do anything.

- Web applications
- Android applications
- Desktop applications
- Big data applications
- Machine learning applications

Lec-01

13-03-2021

How is this course helpful?

- Course is for someone who is willing to start his programming journey with Java or someone who is preparing for his/her Software development interviews.
- This is not gonna be a plain boring course but a highly practical one where you will see how Java is used in industry and production-ready code is written.

8 00 pm
7 00 pm
6 00 pm
5 00 pm
4 00 pm
3 00 pm
2 00 pm
1 00 pm
12 00 pm
11 00 am
10 00 am
9 00 am
8 00 am

2007

28 ★ FRIDAY ★ DECEMBER ★

DECEMBER 2007

S	M	T	W	T	F	S
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

- The aim of this course is not just to teach, one language but also, to make learners habitual to writing highly extensible, modular, and concurrent code - which is something required in the industry as well as in the interviews. The learnings from this course should be beneficial for learners in the long run as well.

Course Content — Lec-1

1. Getting started with Java! (module-1)

- Before starting any language, it is always good to know its internal details, the industry-wide scope, and some of the most important use cases where Java is being used right now. In this section, we will be discussing how Java is different from its counterparts (C, C++, etc) and some of the interesting features of this language that makes it widely recognized in the software industry.

- We will also deep dive into some of the frequently used terminologies in the Java world such as JVM, JDK, JRE, etc. Finally, we will be demonstrating how learners can set up your (learners') local development environment on learners' machine with the help of an IDE. We will be using IntelliJ and Java 8 for this entire course.

- User (You) can also use Eclipse but this is really a matter of personal choice and we feel (RBR's team) that IntelliJ will be much better if you are just starting with Java. We'll not go into the specifics of different Java versions (Java 15 is the latest one right now) as that is not the goal of this course.

- What makes Java Different?
- Java Terminologies explained - JDK, JRE, JVM
- Setting up Java Environment on users (your) machine.

2. First Java Program (module)

- In this section, we will be doing a deep structural analysis of a Sample Java program. We'll not just run through the program but we'll also explain what are the steps and different entities (javac Compiler, class loader, Byte Code verifier, etc) involved when you compile and execute any Java program.

- We (RBR's team) will discuss the concept of Bytecode in detail which makes this language portable across platforms. We'll show user (You) how user (You) can execute any java program with and without an IDE.

- Although user (You) may not be actually using the terminal way of executing the java programs but it will give user (You) a nice idea of how an IDE executes it internally. Here, we'll also explain how the process is different for C++ programs just to give user (You) an idea about why Java is known to be slower as compared to C++.

- We've seen a lot of competitive coders saying that C++ is exceptionally faster than Java but do you really know why is it so? We'll be covering all of this in this section.

- Deep Code Inspection
- Compiling and Executing the Java program
- The ByteCode Magic

3. Java Tooling (module)

- In this section, we'll be covering all the necessary tooling which is required to create java programs. If you have done some programming in other languages, you may find this section very familiar. Although some things may be common across languages, we'll be covering some java specific features as well. For Eg - Data types such as int, long, float is present in all the languages but do you know Java has classes such as Long, Integer, BigDecimal, etc. for representing the data types.

- Why do we need these classes when we already had the support for primitive data types?

In case, you(users) are havitual of using null-terminated C/C++ style strings, you'll get to know why String in java are special and different!

Once we cover all the topics from this section, we will pick some of the most frequently asked problems in Software Industry and try to solve them by applying all the learnings from this section.

faang / flipKart / Goldman Sach

facebook Amazon Apple Netflix Google

- Basic introduction to classes and Objects
- Variables and Data Types
- AutoBoxing and Unboxing
- Manipulating Operators in Java
- Conditional statements in Java
- Introduction to Loops
- Introduction to Methods
- Introduction to Enums

- Access Modifiers in Java
- Typecasting in Java
- The special class - Strings
- Special keywords - static and final
- Practise - interview Problems

4. Object - Oriented Programming in Java

- Object - Oriented programing concepts which are going to help you (users) to write extensible and modular code.

- We'll not be focusing on the definitions and basic examples which you can find everywhere in any random java course.

- We'll explain each of the below-mentioned concepts from the low-level design perspective so that you really understand what exactly makes the code extensible and robust to new requirements. This is a very important skill that is desired in the Software industry.

- We will finish this section by designing and implementing a low-level Design interview (Multi level cache) problem asked in the Flipkart Machine Coding round.

- This is going to be a very practical and important section of this course. Please don't miss this! You'll not find such implementations present on the Internet freely. After completing this section, you'll start understanding why it's not enough to just code but also. How to Code.

- Memory Allocation and Constructors
- Reference vs Object vs Instance
- Abstract class vs Interface
- Method Overloading vs Method Overriding
- Abstraction vs Encapsulation
- Inheritance vs Composition
- Static Vs Dynamic Polymorphism

- Flipkart Interview Problem – Designing and Implementing a Multi-level cache

5. Java's Generic (Not everyone cover this module)

- This is a bit advanced topic but we will try to cover the areas that are required for interviews. With the help of generics, you can write code that is applicable to many data types with the same underlying behavior.

- Just to give a very high-level idea, let's say you're designing a method that can multiply 2 numbers given as input. Since a number could be an int, long, or a Double. so you might come with this:

```
public int Multiply(int a, int b)
public double Multiply(double a, double b)
public float Multiply(float a, float b)
```

with the help of generics, you'll be able to create a single method that will be customized for the type that invokes it.

```
public T Multiply<T>(T a, T b)
```

- Why do we need Generics?
- Designing a Generic Class.
- Designing a Generic Method
- Bounding Type Parameters
- Introduction to wildcards
- Adobe Interview Problem - Designing and Implementing a Generic Merge / Quick sort Algorithm

6. Java's Collection Library (Same as module 5)

- This section will go through some of the most important Data Structure and Algorithms that are readily available in Java.
- These libraries actually save a lot of time during coding competitions or in the online coding round of Product based companies as you do not have to write a piece of code from scratch which is already available in the Java language.
- We will not just explain the usage but we'll also explain the internal implementation, the time and space complexity, etc
- We have seen so many people using these libraries in interviews but they don't really know what's going on under the hood. Sometimes interviewer might question you about the internal details so we'll cover this important section in detail. In the end, we'll solve some interview problems asked in Google using Java collections.
  - Collection Framework Overview
  - List interface implementations - HashMap vs TreeMap vs LinkedHashMap
  - Set interface implementations - Hashset vs Treeset vs LinkedHashSet
  - Queue interface implementations - PriorityQueue
  - Comparable vs Comparator
  - Frequently Used Algorithms - Sorting & Searching
  - Google interview Problem

Memorable Event
Appointment
Phone No

8 00 pm
7 00 pm
6 00 pm
5 00 pm
4 00 pm
3 00 pm
2 00 pm
1 00 pm
11 00 am
10 00 am
9 00 am
8 00 am

2007

20 * THURSDAY * DECEMBER

DECEMBER 2007
S M T W T F S
1
2 3 4 5 6 7 8
9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30 31

(Laptop built on this)

[We are not focusing on full stack development. We are focusing on interview preparation.]

7. Multi-Threading And Concurrency in Java

↳ for Product Based Companies.

V.V.I

- Understanding and Knowledge of Multi-Threading display the maturity and technical expertise in a language and can be a differentiating factor in landing a great offer at many product based company. You might have already studied some of the theoretical concept in OS such as process, critical sections, locks, semaphores, etc.

- In this section, we'll go through the libraries which are available in Java for creating, running, and synchronizing threads.

- When multiple threads are executed in a program, they might be accessing a shared section of memory which can lead to problems such as race conditions, data-inconsistency, etc. We will be discussing in detail all the alternatives available in Java for providing synchronization.

- Even if users (you) are not familiar with this concepts, we'll be discussing all of this without making any assumptions. In the end, we will apply the learning to solve one of the famous interview problems - Designing a Multi-Threaded Job Scheduler.

- Introduction to Threads
- Critical sections, Race conditions, and Deadlock
- Mutex vs Semaphore vs Monitor
- Creating & Running Threads in Java
- Introduction to the Executor Framework and Thread pools
- Designing a Thread safe class
- Thread synchronization Tooling and Libraries in Java
  - Volatile & synchronized Keywords
  - Intrinsic locks - wait and notify
  - Re-entrant locks and condition Variables
  - Semaphore in Java
  - specialized concurrent Data structures
- Interview Problem - Designing & Implementing a Job scheduler

JDBC - to connect DBMS