15 Things Java Developers to Learn in 2024

Tools, technologies, and framework Java programmers can learn in 2024

JAVINPAUL

JAN 02, 2024 · PAID

9 1 3 Share ···

Hello guys, Happy New Year, just like that another year has gone and a new year has started. As a programmer, our biggest challenge is to keep ourselves up-to-date. Technology changes very fast, and you will see a new version of the programming language and framework coming every couple of years.

As the tech landscape continues to evolve, Java remains a stalwart in the world of programming languages, prized for its reliability, portability, and widespread use. In 2024, staying ahead of the curve as a Java developer requires not just mastering the fundamentals but also embracing emerging trends and technologies.

Whether you're a seasoned Java developer or just starting your journey, this article serves as a guide to the 15 essential things you should learn to thrive in the dynamic and ever-expanding Java ecosystem.

From new language features and tools to best practices and industry trends, our list covers a diverse range of topics designed to empower Java developers with the skills and knowledge needed to tackle contemporary challenges. J

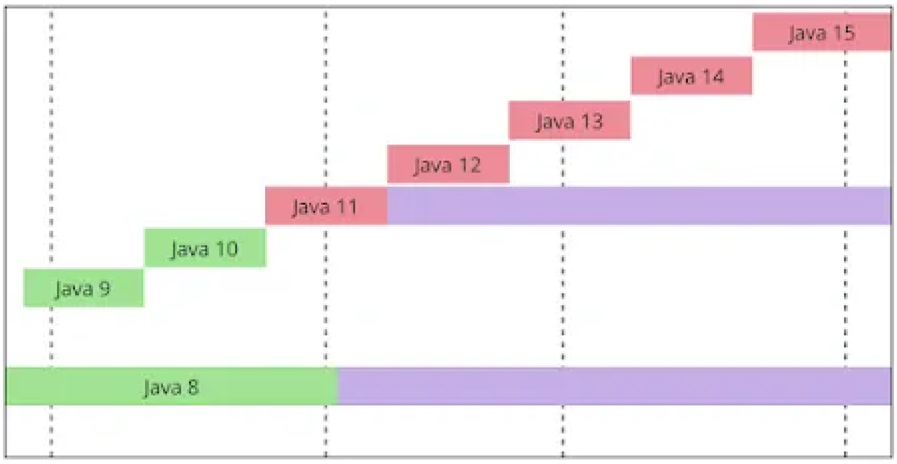
Let's dive into the 15 things that every Java developer should have on their learning radar this year.

What Java Programmers should learn in 2024?

Here is my list of things a Java developer should learn in 2024, there is a good chance that you already know most of them but learning them better and doing a deep dive doesn't hurt:

1. **Java 17 and Beyond**
   - Explore the new features introduced in the latest Java versions, such as records, sealed classes, pattern matching, and enhancements to the garbage collector.
   - Understand the benefits of upgrading to newer Java versions, including improved performance, security patches, and language enhancements.

- For resources, in the last Udemy 10$ sale, I have purchased a host of courses, and one of them is **The Complete Java Masterclass**, which is updated for Java 17, and I am looking forward to starting my Java 17 journey with that.



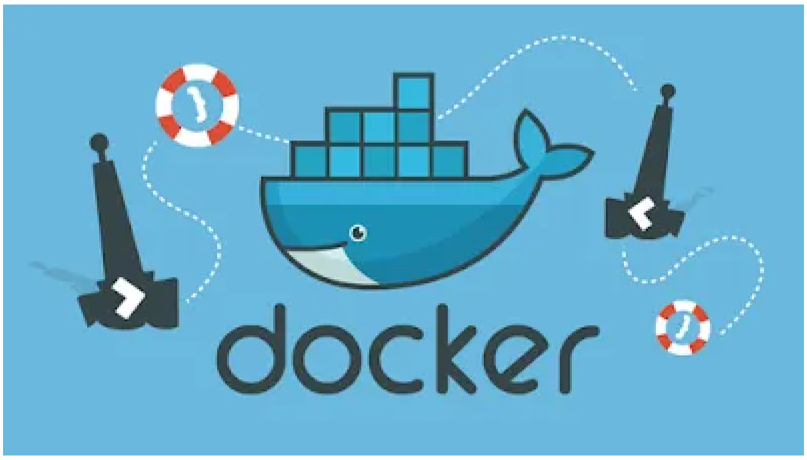
Btw, if you have yet to start with JDK 8 then here is my list of favorite Java 8 tutorials and courses which you can free of cost: 10 best tutorials to learn Java 8.

2. **Project Loom:**
   - Study Project Loom to understand how it aims to simplify concurrent programming in Java by introducing lightweight threads known as fibers.
   - Explore the potential impact on application scalability and responsiveness when adopting a more efficient concurrency model.
3. **Micronaut and Quarkus:**
   - Dive into Micronaut and Quarkus to build microservices and cloud-native applications with minimal resource consumption.
   - Understand the key features of each framework, such as ahead-of-time compilation, low memory footprint, and support for GraalVM native images.
   - If you need resources you can start with Starting with Quarkus, an excellent Udemy course for Java and Microservices developers that want to understand Quarkus. Antonio Goncalves, a Java Champion and author of multiple Java EE books, is the creator of this course.
4. **Spring Boot 3.0:**
   - Stay updated on the latest developments in Spring Boot, including features like improved startup times, enhanced support for GraalVM, and updates to Spring Framework components.
   - Explore the migration strategies and best practices for upgrading existing Spring Boot applications to newer versions.
   - If you need resources, you can check **Master Spring Boot 3 & Spring Framework 6 with Java**, it's an updated course to learn both Spring 6 and Spring Boot 3.0

    If you like books, you can also check out this list of advanced Spring Books for Java developers from Manning and Packt Publications.

5. **JEPs (Java Enhancement Proposals):**
    - Regularly review Java Enhancement Proposals to stay informed about upcoming changes and experimental features.
    - Experiment with JEPs in non-production environments to understand their implications and benefits.
6. **Modern Java Build Tools:**
    - Master modern build tools such as Gradle and Maven, exploring advanced features for dependency management, task customization, and build optimization.
    - Integrate build tools with continuous integration pipelines to automate the build and deployment processes.
    - If you need resources, **Apache Maven: Beginner to Guru is a good starting point for Java developers who want to learn Maven**. The Gradle Masterclass is good for starting with Gradle.
7. **Kotlin for Java Developers:**
    - Learn Kotlin syntax and features, focusing on interoperability with Java.
    - Explore Kotlin's concise syntax, null safety, and extension functions to understand how it complements Java in modern application development.
    - If you need resources, **Kotlin for Java Developers** is a good one for Java developers to learn Kotlin in a quick time.
8. **Cloud-Native Development:**
    - Understand the principles of cloud-native development, including containerization with Docker, microservices architecture, and the use of cloud platforms.
    - Explore cloud-native tools and services for monitoring, logging, and scaling applications in a cloud environment.
    - If you need resources, s, I recommend checking **Master Microservices with Spring Boot and Spring Cloud** course by In28Minutes on Udemy for Java developers.
9. **Reactive Programming:**
    - Dive into reactive programming concepts using libraries like Reactor or Akka.
    - Learn how reactive programming facilitates the development of responsive, scalable, and resilient applications.
    - If you need resources, I recommend checking out Build Reactive MicroServices using the Spring WebFlux/Spring Boot course on Udemy. It's a good one to learn both Reactive programming and WebFlux.

10. **Machine Learning with Java:**
    - Explore Java libraries for machine learning, such as Deeplearning4j and Apache OpenNLP.
    - Understand how to integrate machine learning models into Java applications for tasks like natural language processing or predictive analytics.
    - If you need resources, **Complete Machine Learning & Data Science Bootcamp** by ZTM Academy is a nice course to elarn about ML but I haven't found a good course that uses Java as a language yet.
11. **Jigsaw** (**Project Jigsaw**):
    - Study Project Jigsaw to understand modular programming in Java and how it improves code maintainability and scalability.
    - Experiment with creating modular Java applications and libraries.
12. **JPA** (**Java Persistence API**) **Best Practices:**
    - Master best practices for using JPA and Hibernate for efficient database interactions.
    - Explore techniques for optimizing queries, handling relationships, and managing entity lifecycle in JPA.
    - If you need resources, then I suggest going through **Hibernate and Spring Data JPA: Beginner to Guru** by John Thompson, one of the Java, Spring, and Hibernate experts.
13. **Containerization with Docker:**
    - Gain hands-on experience with Docker to containerize Java applications.
    - Learn about Docker images, containers, and orchestration tools for deploying and managing containerized applications.
    - If you have already realized the importance of Kubernetes and looking to learn Kubernetes in 2024 then **Docker and Kubernetes: The Practical Guide** by AcadMind on Udemy is an excellent course to start with.



If you prefer free resources, like free online courses, then you can also check out this list of free Docker and Kubernetes courses for Java developers to start your journey.

14. **Serverless Java:**
    - Explore serverless computing concepts and use cases for Java applications.
    - Implement serverless functions using platforms like AWS Lambda or Azure Functions, understanding their benefits and limitations.
    - For those who need resources, the **Serverless using AWS Lambda for Java Developers** course by Bharath Thippireaddy is a good place to start with
15. **Continuous Integration/Continuous Deployment** (CI/CD):
    - Set up CI/CD pipelines for Java projects using tools like Jenkins, GitLab CI, or GitHub Actions.
    - Implement automated testing, code quality checks, and deployment strategies to streamline the software development lifecycle.
    - If you need resources, **Jenkins, From Zero To Hero: Become a DevOps Jenkins Master** is my favorite course to start with. It will teach you Jenkins in good detail along with its role in DevOps.

By delving deeper into these areas, Java developers can not only stay current with the latest trends but also enhance their proficiency in key technologies and methodologies crucial for building robust and modern applications.

Bonus

Three more things you can learn in 2024

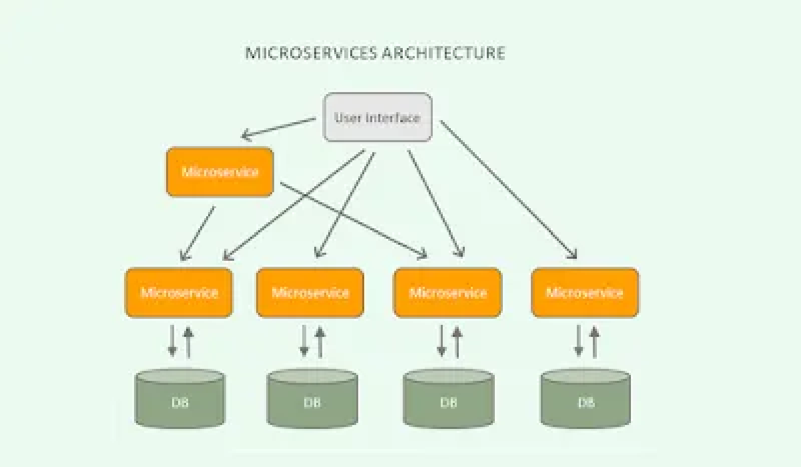
16. Microservices

So, we have already covered 15 things Java developers should learn in 2024 but if you can add one more thing then I would suggest learning Microservice architecture. This is the architecture that goes hand-in-hand with cloud computing because it's easier to deploy and scale in the cloud and that's why most companies are now developing their application using Microservice architecture.

Since Cloud Native development is the future of Software development it makes sense to learn both Cloud Computing and Microservice architecture and how to develop Microservice in Java. Well, when it comes to Java, you have many options.

For example, you can choose the tried and tested duo of Spring Boot and Spring Cloud to create Microservices. Alternatively, you can also use frameworks like MicroNaut and platforms like Quarkus for developing high-performance Kubernetes-based Java Microservice solutions.

Whichever framework you choose, the key is to learn Microservices in 2024 and if you need some courses and books, I recommend checking **Master Microservices with Spring Boot and Spring Cloud** course by In28Minutes on Udemy for Java developers.



And, if you love books, you can check out these best Spring and microservice books for Java developers. My personal favorite is Microservice in Action by Manning's publication.

17. Cloud Technologies

Java developers should learn cloud technologies in 2024 for several compelling reasons. Cloud computing offers scalable infrastructure, enabling developers to deploy, manage, and scale applications more efficiently.

With cloud services, developers can leverage resources on demand, reducing infrastructure costs and improving flexibility. Additionally, cloud platforms provide a variety of managed services, such as databases, storage, and machine learning, simplifying development tasks and accelerating time-to-market. Integration with cloud services allows Java developers to build resilient, distributed systems and facilitates the adoption of microservices architecture.

Furthermore, understanding cloud technologies aligns with industry trends and positions developers to stay competitive in a landscape increasingly focused on cloud-based solutions.

To get started, the Udemy course "[NEW] Building Microservices with Spring Boot & Spring Cloud" is recommended. This course covers the fundamentals of building microservices using Java and Spring Boot, with a focus on cloud-native development and integration with Spring Cloud.

Another way is to prepare for cloud certifications like AWS Solution Architect Associate which will help you to learn AWS and cloud technologies in depth. If you need a resource, I also recommend joining the **Ultimate AWS Certified Solutions Architect Associate SAA-C03 course** by Stephane Maarek, an AWS Guru in 2024.

18. Prompt Engineering

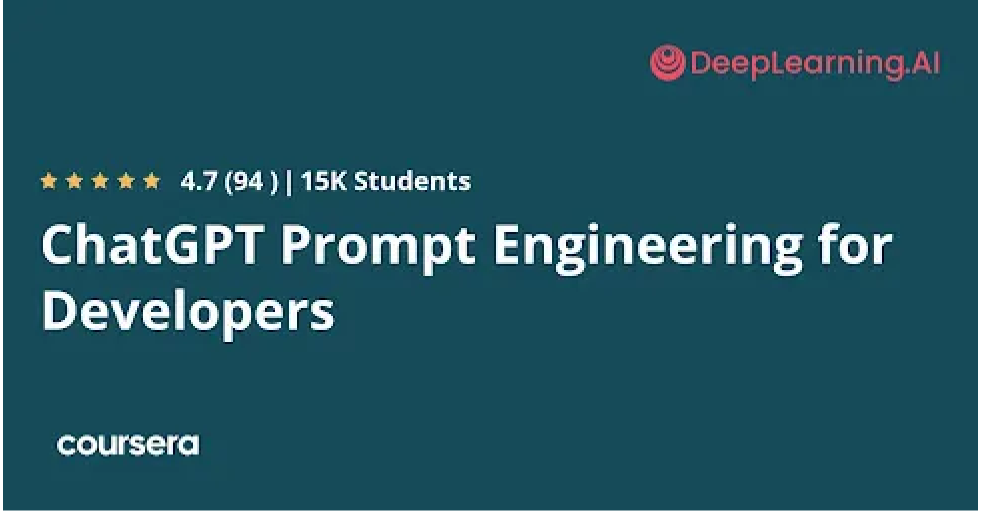
Java developers can benefit significantly from learning prompt engineering in 2024, especially given the growing importance of generative AI like ChatGPT. Prompt engineering involves crafting specific instructions or queries to guide the responses of large language models, enhancing their usefulness.

Java developers can leverage prompt engineering to interact effectively with ChatGPT for various tasks, such as generating code snippets, solving programming challenges, or obtaining insights into complex problems.

By mastering prompt engineering, Java developers can optimize their interactions with ChatGPT, ensuring more accurate and tailored responses to their queries. To delve into prompt engineering, developers can explore the Coursera course "Prompt Engineering for ChatGPT."

This course provides practical insights into creating effective prompts, understanding the principles of prompting, and employing advanced strategies for improved interactions with generative AI models like ChatGPT. It's a valuable resource for Java developers seeking to enhance their skills in leveraging generative AI technologies for diverse applications.

There is also a Coursera project called ChatGPT Prompt Engineering for Developers by Andrew Ng, one of Coursera's founders and AI experts. You can also combine join this course to learn Prompt Engineer for developer productivity.


DeepLearning.AI
★★★★★ 4.7 (94) | 15K Students
ChatGPT Prompt Engineering for Developers
coursera

By the way, instead of joining these courses and specializations individually, you can also join Coursera Plus, a subscription plan from Coursera that gives you unlimited access to their most popular courses, specializations, professional certificates, and guided projects.

That's all about **what Java programmers should learn in 2024.** As I said, Technology changes at a rapid speed, and the biggest challenge for programmers is to keep themselves up-to-date. Apart from this list, there is plenty of other stuff that you can look up in the new year like learning a new programming language like Kotlin, but for me, I will be more than happy if I can achieve these goals in 2024.

I wish you guys a Happy New Year 2024.

P. S. - If you are looking for some gift ideas for your programmer friend this holiday season, then you can check out my list of the 10 best gifts for programmers and tech geeks.

9 Likes · 3 Restacks

← Previous

Next →

1 Comment

Write a comment...

Soma Soma's Substack Jan 2

Thanks, I am also thinking to learn about Quarkus this year, heard a lot of good things about it.

LIKE REPLY SHARE