Introduction to the Course

Hello there, and welcome to the Java Masterclass!

I'm Tim Buchalka, your instructor, and I'm excited to have you here.

I've been programming for over 40 years, with close to 25 years of those dedicated to Java.

My goal is simple: to help you become a confident Java programmer.

Introduction to the Course

Whether you're a total beginner, have experience in other languages, or even know a bit of Java, this course is designed for you.

It will take you from the basics all the way to advanced concepts and is constantly updated to stay relevant.

In fact, I just remastered this entire course for JDK 17!

Introduction to the Course

Now, you might be wondering, Why JDK 17 when there's already JDK 22?

Great question! I'll explain why in the "Which JDK?" video coming right up.

But even if you're using an older version, like Java 11 or 8, you'll still get tons of value from this course.

Introduction to the Course

Before we dive in, make sure you check out a few more videos in this section.

They cover important stuff like video quality, setup tips, and that explanation about JDK 17.

Alright, that's it! Let's get started. See you in the next video!

Remaster in Progress

I'm thrilled to be giving this entire course a major upgrade!

When it launched back in 2015, we were on Java 8, but now I've brought it all the way up to Java 17.

Don't worry, you can absolutely start the course right now; I'm releasing the remastered sections as I complete them.

Remaster in Progress

What does this upgrade mean for you?

- **Java 17 Power:** You'll learn to use all the best that Java 17 has to offer.
- **Downloadable Slides:** Grab those slides for reference and for studying offline.
- **Polished Learning Experience:** Say goodbye to old errors and confusing points – get clearer explanations.
- **Future-Proof:** This course isn't going anywhere. I'll keep it updated with the latest Java versions. My goal is for this to be THE go-to resource for mastering Java!

See you in the next video!

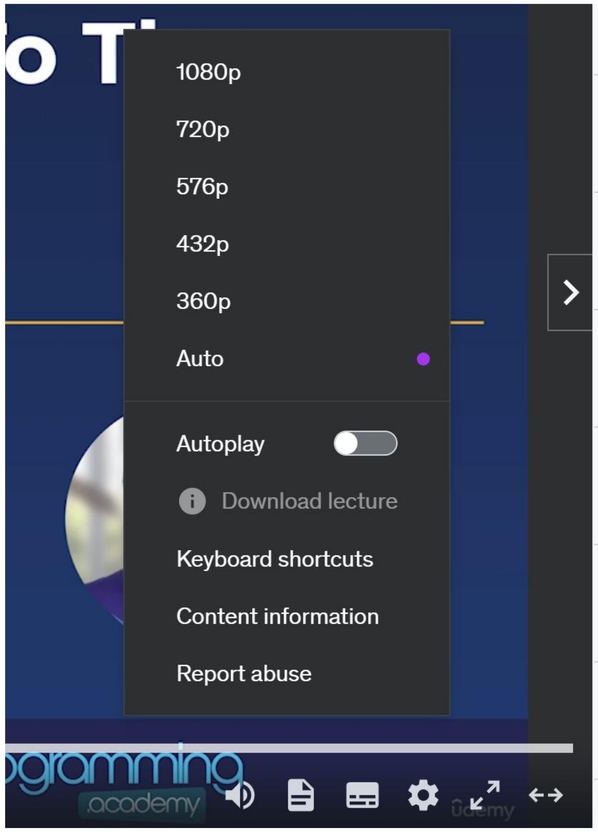
Video Quality

Great news! All videos in this course are uploaded in 1080p for the best possible viewing experience.

If you notice some initial blurriness, don't worry – it should clear up within a few seconds.

Video Quality

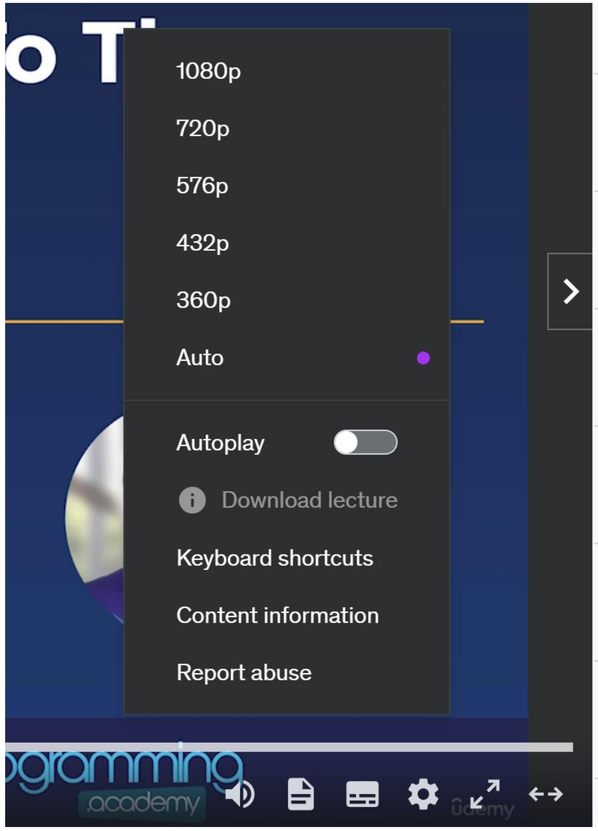
Here's how to ensure the best quality:



- **Choose "Auto"**: If you see the "Auto" option in the settings (little cog icon), select that. It'll automatically give you 1080p if your internet connection allows.

Video Quality

Here's how to ensure the best quality:



- **If No "Auto"**: Pick "720p" for consistently good quality.
- **Be Patient:** Sometimes videos need a moment to load fully.

Video Quality

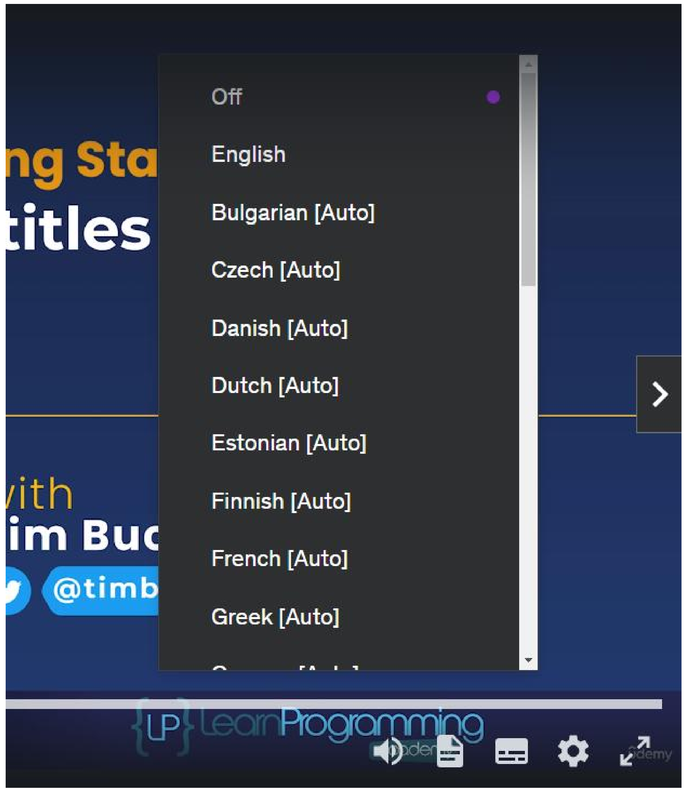
Rest assured; I want you to have the clearest learning experience possible!

Thanks for watching, and let's dive into the next video!

Subtitles

Hey everyone! If I sometimes speak too fast or my accent makes things a bit tricky, I totally understand.

That's where subtitles come in! Here's how to turn them on:



1. Click the little captions button at the bottom of your video.
2. Choose your preferred language.

Subtitles

Important Note: This course has hand-edited English subtitles, which means they're as accurate as possible.

I put in a lot of work to make sure they help you learn!

If you notice any errors, please let me know which video and the timestamp, and I'll get it fixed right away.

For languages other than English, the subtitles are machine-generated.

They might not be perfect, but they can still be helpful.

Subtitles

My goal is to make sure you have the best possible learning experience, so please use the subtitles if they make following along easier!

Alright, let's move on to the next video!

How to Get Help

Learning to program has its challenges, but don't worry – help is always available!

Here's how to get it:

- **Questions and Answers Section:**
    - Log in and use the "Q&A" tab.
    - Click "Ask a Question," give your question a clear title, and provide a detailed description.
    - I, or one of my teaching assistants, or even other students will be happy to help!

How to Get Help

Learning to program has its challenges, but don't worry – help is always available!

Here's how to get it:

- **Tips for Asking Great Questions:**
    - **Be Specific:** Mention the lecture you're stuck on, paste your code (use the code formatting button), and include any error messages.
    - **Think Like a Future Pro:** This skill will be vital in your programming career! Help your helpers by providing as much context as possible.

How to Get Help

Learning to program has its challenges, but don't worry – help is always available!

Here's how to get it:

- **Tips for Asking Great Questions:**
    - **Search First:** Many questions have already been answered. The search bar is your friend!
    - **New Question = Faster Help:** Usually, it's better to start a fresh question rather than adding yours to an old thread.

How to Get Help

Remember, you're not alone!

We're a community here to support you.

Asking effective questions is a KEY skill for any programmer, so take this chance to practice.

Alright, ready to keep learning?

Let's dive into the next video!

Important Tip - Source Code

Let's talk source code. This is where the magic happens!

Watching me code is helpful, but the real learning comes when YOU start typing.

Here's why you MUST type the code yourself:

- **Understanding Through Doing:**

  Programming isn't about memorization; it's about problem-solving.

  Typing, making mistakes, then fixing them – that's how you build those skills.

- **Muscle Memory:** Your fingers need practice to become fluent in the language of Java.

Important Tip - Source Code

How to Get the Most Out of It:

- **Option 1 - Shadow Me:** Watch the video once for the big picture, THEN type along during the second viewing.

  Pause and rewind as needed!

- **Option 2 - Dive Right In:** If you're feeling brave, type along the first time.

  Be ready to pause and experiment.

- **Download My Code**: Each lecture has resources!

  Use this to compare your solution to mine or jumpstart your project.

Important Tip - Source Code

Bottom Line:

Typing the code is your path to becoming a confident Java programmer.

It's NOT optional!

Alright, enough talk. See you in the next video.

Biggest Tip to Succeed as a Java Programmer

The biggest tip for becoming a programmer? Never give up.

This journey will have challenges.

You might feel frustrated and want to quit.

But those who push through become the programmers they dream of being.

Remember, learning to code is tough but incredibly rewarding.

It takes time for things to click.

Trust in the process, and you WILL get better.

Biggest Tip to Succeed as a Java Programmer

To help you stay motivated, I have a lecture coming up with two essential videos:

- **The Power of Persistence:** A short video to inspire you to keep going.
- **The Four Stages of Learning to Code:** Understanding this process keeps you on track.

Your mindset is key!

I believe in you.

Take a few minutes to watch those videos, then let's get back to coding!

Which Version of Java?

Let's dive into a super important concept in the Java world: the LTS, or Long-Term Support.

This is a question that comes up all the time when choosing a Java Development Kit.

Which Version of Java?

What is the Java LTS?

- Oracle, the "owners" of Java, release Java updates a few times a year.

  Special ones are designated "Long-Term Support", or LTS.

- Regular releases are only supported for six months.
- LTS versions, on the other hand, receive extended security updates, bug fixes, and patches for years.
- As a result, LTS releases are your rock-solid choice, especially for production environments and when learning Java.

Which Version of Java?

Why Use a Java LTS version?

- **Stability:** Companies need stable systems.

  LTS versions minimize surprises with bugs and security issues already ironed out.

- **Support:** If something goes wrong with an LTS version, you've got a longer window where official support and updates are guaranteed.

  This is vital for businesses.

Which Version of Java?

Why Use a Java LTS version?

- **Ecosystem:** Libraries, tools, and frameworks often focus their support on LTS versions.

  That means a much smoother experience for you.

- **Regular versions:** While they have cool new features, they lack long-term support.

  Neat stuff eventually gets rolled into the next LTS version, making it the smarter choice for most production work.

Which Version of Java?

JDK 17 vs. JDK 21: Does it matter?

- You might see a course like this one using JDK 17 and wonder, "Isn't JDK 21 the latest and greatest?". True, but...
- Java LTS versions are designed for long-term use.

  Companies don't instantly jump to the newest JDK.

  Why? They prioritize the testing, ensuring compatibility with existing systems, and just letting the LTS mature a bit.

Which Version of Java?

Key Takeaways

- For serious projects, learning with an LTS like JDK 17 is the way to go.

  You'll master core Java skills relevant across versions.

- When a new LTS comes out, companies steadily migrate.

  There's no need to chase every JDK update, especially as cool non-LTS features get
  merged into the next LTS anyway.

- And hey, for those eager to try JDK 21, don't worry!

  I'll be adding a section on its new features to this course soon.

Which Version of Java?

There you have it!

Java LTS versions of the JDK are your friend when it comes to building reliable applications.

And that's why we are using JDK 17 in this course.