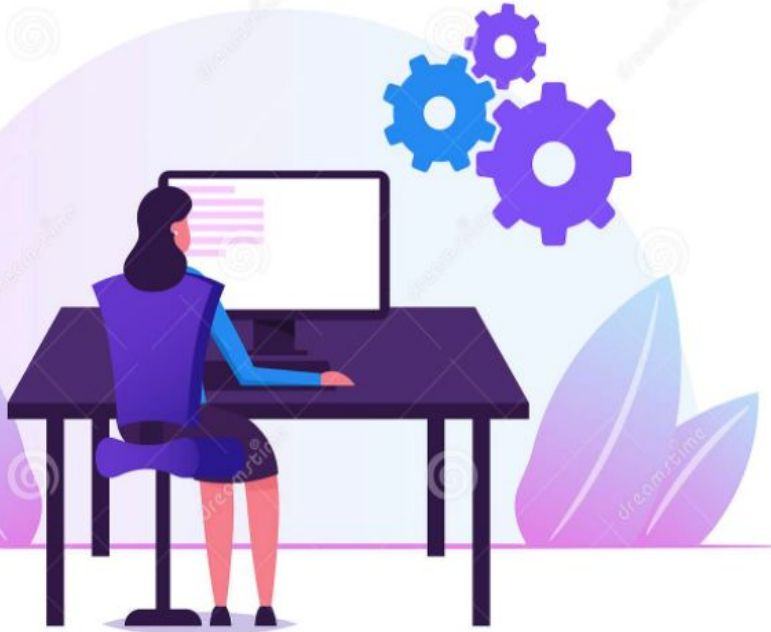


Welcome!



WWCode Digital + Backend

November 2, 2022

- We'll start in a moment. :)
- We are **RECORDING** tonight's event.
- We may plan to take screenshots for social media.
- If you are comfortable, turn the video ON. If you want to be anonymous, then turn the video off.
- We'll introduce the hosts & make some time for Q&A at the end of the presentation.
- Feel free to take notes.
- Online event best practices:
 - Don't multitask. Distractions reduce your ability to remember concepts.
 - Mute yourself when you aren't talking.
 - We want the session to be interactive.
 - Use the 'Raise Hand' feature to ask questions.
- **By attending our events, you agree to comply with our [Code of Conduct](#).**

Introduction & Agenda

- Welcome from WWCode!
- Our mission: Empower diverse women to excel in technology careers.
- Our vision: A tech industry where diverse women and historically excluded people thrive at any level.
- Improve your code debugging skills:
- What is code debugging?
- Why is debugging an important skill?
- Examples and demo.
- Q & A.



Prachi Shah
Instructor,
Senior Software Engineer.
Director, WWCode SF



Harini Rajendran
Host,
Software Engineer, Confluent.
Lead, WWCode SF

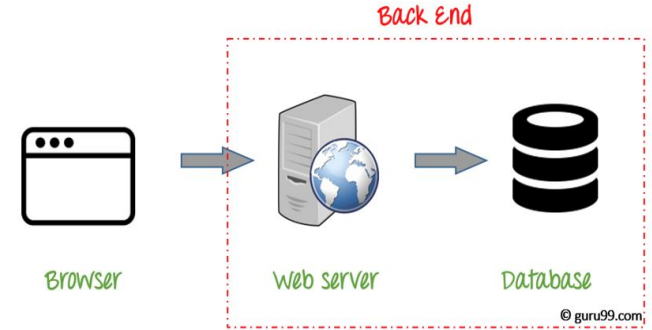
Disclaimer:

- Sessions can be heavy!
- Lots of acronyms.
- Instructor doesn't know everything.

Backend Engineering

- Design, build and maintain server-side web applications.

- Common terms: Client-server architecture, networking, APIs, web frameworks, platform, micro-service, databases, web fundamentals, operating systems, etc.



- Tech Stack: Java, PHP, .NET, C#, Ruby, Python, REST, AWS, Node, SQL, NoSQL, etc.
- Other domains: Front end engineering, full stack engineering, design & user experience, mobile development, devOps engineering, machine learning, etc.

Code Debugging

- Run the code step-by-step.
- Understand the program flow and data flow.
- Find problems in the code: errors and exceptions.
- Find, fix and test for bugs.
- Test the functionality (business logic).



Code Debugging

Code debugging is an important skill:

- All code has bugs. For quality code, we want the application to be bug-free.
- Bugs can introduce serious flaws in the functionality and cause serious effects.
- Functionality (business logic) should work as designed and expected.
- Best practices for coding support bug-free development.
- External integration development should be bug-free and tested thoroughly.
- Understand someone else's code.
- Explain your code to someone else.
- Explaining application functionalities (business logic) to others.
- Essential skill for engineers to design, write, execute and test quality code.

Code Debugging

When debugging:

- Observe the error messages and exceptions. Review the stack trace.

```
Exception in thread "main" java.lang. NullPointerException:  
Cannot invoke "java.util.HashSet.contains(Object)" because " roots" is null at  
com.wcodesf.backendstudygroup.codedebugging.ReplaceWords.replaceWords  
InSentence(ReplaceWords.java: 58) at  
com.wcodesf.backendstudygroup.codedebugging.ReplaceWords.main(Replace  
Words.java:38)
```

- Write clean code. Use best practices (can be language-specific).
- “DRY” principle: don’t repeat yourself – don’t repeat the code.
- “KISS” principle: keep it simple stupid – write simple code that is easy to read & test.
- Ask for help: from a co-worker, the internet, books, etc.

Code Debugging

How to debug code (with examples):

- Print statements after running the application. Logging statements using loggers.
- Insert assert statements in the functions to check for correct conditions.

```
for (int i = 1; i <= word.length(); ++i) {  
    prefix = word.substring(0, i);  
    if (words.contains(prefix)) {  
        System.out.println(i); // i=3  
        System.out.println(word.length()); // word="cattle" and length=6  
        logger.info(prefix); // prefix="cat"  
        assert i != 0;  
        break;  
    }  
}  
Output:  
3  
6  
Apr 02, 2022 1:45:27 AM com.wvcodesf.backendstudygroup.codedebugging.ReplaceWords  
replaceWordsInSentence  
INFO: cat
```

- IDEs has debugger to debug through the code and view objects, data and errors.
- Unit testing.

Code Debugging

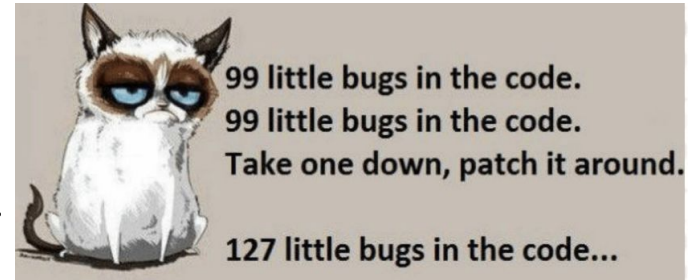
Code debugging tools:

- IDE (Integrated Development Environment) provides debugging functionality. Examples: IntelliJ IDEA, Eclipse, MS Visual Code Studio, etc. Add breakpoints to statements and the debugger will debug step-by-step.
- Frontend code: TypeScript auto-detects errors and bad syntax.
- In the browser: `console.log()` or `console.debug()` to debug frontend code.
- node.js has a debugger that does stepping and code inspections.
- Python has a built-in debugger called `pdb`.
- Java has a built-in command-line debugger called `jdb`.
- Sentry logging prints out useful debugging information about the application.
- More: GNU Debugger `gdb`, CodeLite for C/C++, Xcode debugger for iOS, etc.

Code Debugging

Debugging as an engineer at a workplace:

- Debugging code locally for development.
- Debugging code locally for testing.
- Debugging code locally for integration (external).
- Fix production bugs.
- Pair programming with another engineer.
- Understand fellow engineer's code.
- Explain your code to fellow engineers.
- Explaining application functionalities (business logic) to others.
- When setting up a new project, debugging & running tests is the best way to learn!



Code Debugging

Demo: Java Code in IntelliJ IDE with Program Flow and Data Flow.



Code Debugging

Key Takeaways:

- Run the code step-by-step.
- Understand the program flow and data flow.
- Find problems in the code: errors and exceptions.
- Find, fix and test for bugs.
- Test the functionality (business logic).
- All code has bugs. For quality code, we want the application to be bug-free.
- Write clean code. Use best practices (can be language-specific).
- Understand existing applications, explain code functionality and write good tests.
- IDEs, and languages have in-built debuggers & independent debugger tools.
- Essential skill for engineers!
- Ask for help: from a co-worker, the internet, books, etc.

Backend Engineering

References:

- [What is Debugging?](#)
- [Debugger](#)

Backend Study Group:

- [Presentations](#) on GitHub and session recordings are found on [WWCode YouTube channel](#).
- Upcoming sessions:
 - November 17th, 2022 about [Intro to Distributed Systems](#).
 - December 8th, 2022 about [Git and Version Control System](#)

Women Who Code:

- [Technical Tracks](#) and [Digital Events](#) for more events.
- Join the [Digital mailing list](#) for updates about WWCode.
- Contact us at: contact@womenwhocode.com
- Join our [Slack](#) workspace and join `#backend-study-group`!

You can unmute and talk or use the chat.

