CoGrammar

Tutorial: Debugging

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

Software Engineering Lecture Housekeeping

- The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly. **(FBV: Mutual Respect.)**
- No question is daft or silly - **ask them!**
- There are **Q&A sessions** midway and at the end of the session, should you wish to ask any follow-up questions. Moderators are going to be answering questions as the session progresses as well.
- If you have any questions outside of this lecture, or that are not answered during this lecture, please do submit these for upcoming Open Classes. You can submit these questions here: **Open Class Questions**

Software Engineering Lecture Housekeeping cont.

- For all **non-academic questions**, please submit a query: **www.hyperiondev.com/support**
- Report a **safeguarding** incident: **www.hyperiondev.com/safeguardreporting**
- We would love your **feedback** on lectures: **Feedback on Lectures**

Reminders!

Upcoming deadlines, GLH requirements, etc.

Guided Learning Hours

By now, ideally you should have 7 GLHs per week accrued. Remember to attend any and all sessions for support, and to ensure you reach 112 GLHs by the close of your Skills Bootcamp.

Milestone 2

Mid-Course Progress
Software Engineering: Finish 14 tasks by week 8.

Lecture Objectives

1. **Debugging**
    a. **Hypothesis-driven debugging**
    b. **Debugging in VS code**

Debugging

Debugging is the process of finding and fixing issues or "bugs" in your code. You use clues, like error messages or unexpected behavior, to figure out what went wrong. Debugging helps make your programs work smoothly and teaches you a lot about how to write better code.

Hypothesis-Driven Debugging

- Hypothesis-Driven Debugging is an approach to troubleshooting and fixing issues in computer programs by formulating educated guesses or hypotheses about the potential causes of the problem.
- Instead of making random changes, we observe symptoms, create hypotheses, test them, analyze the results, and iterate until they identify and resolve the root cause of the issue.

Hypothesis-Driven Debugging

The key steps involved in this approach:

- Identify the problem or failure.
- Formulate a hypothesis about the cause.
- Design and conduct experiments to test the hypothesis.
- Analyze the results and refine the hypothesis if necessary.
- Repeat the process until the issue is resolved.

Benefits of Hypothesis-Driven Debugging

The advantages of using this approach in debugging:

- Provides a structured and systematic approach to problem-solving.
- Helps in narrowing down the potential causes of the issue.
- Saves time and effort by focusing on relevant experiments.
- Facilitates learning from debugging experiences and improving future debugging skills.

CoGrammar

Questions around Debugging

Session Break

CoGrammar

CoGrammar

Thank you for joining