Tutorial: Debugging

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**

CoGrammar

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**

Debugging

Lecture Objectives

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

Hypothesis-Driven Debugging

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

Thank you for joining