Semi-Structured Interview Protocol

Green: Questions specific to Group01 (Informed)
Blue: Questions specific to Group02 (Non-Informed)
Time limit: 15 minutes

RQ1: How do programmers comprehend code?

1. Do you find the initial code that you debugged makes sense? Why or why not?
   Do you find the code generated by Copilot makes sense? Why or why not?
2. **[G02]** ⭐ Do you find that the code snippets you debugged were actually generated by Copilot during your programming?
   What is your current opinion on the quality of Copilot-generated code?
3. How do you differentiate between code written by humans and code generated by Copilot?
   Specifically, please describe your opinion on the differences between human-written and Copilot-generated code in the following four aspects:
   - Functionality
   - Complexity
   - Readability
   - Coding style

RQ2: What are the strategies for debugging?

- **General debugging strategy**
  Can you describe the process you typically follow when debugging code?
- **Validation**
  When attempting to locate the error, which part of the code do you examine?
  Additionally, which part of the code is particularly challenging to validate for [specific task]?
  [Flexible - IDE Features] Could you discuss your usage of IDE features during the bug-validating process?
- **Repairing**
  After identifying an error, what steps will you take to repair it?
  Additionally, which part of the code is particularly challenging to repair for [specific task]?
  [Flexible - GitHub Copilot] Could you discuss your usage of the GitHub Copilot during the bug-repairing process?
  [Flexible - IDE Features] Could you discuss your usage of IDE features during the bug-repairing process?
- **Comparison**

In your opinion, how does the process of detecting and repairing errors generated by Copilot differ from traditional code debugging?

RQ3: Does trust or overreliance on Copilot change over time during debugging?
In general, do you trust Copilot-generated code?
Has your level of trust in Copilot changed over time as you've used it more? Have you become more or less reliant on it?
how do you validate the generated code to ensure its correctness and reliability?

Over-Reliance
Have you ever found yourself in a situation where you relied on Copilot so heavily that you stopped thinking critically about the code you were debugging? If so, could you describe the outcome of this situation and how it affected your debugging process?
Under-Reliance
Could you describe a situation where you initially did not trust Copilot-generated code, but its solution turned out to be correct? How did this experience affect your level of trust in the tool and your approach to debugging with Copilot?
Proper Untrust
Have you ever faced a situation where Copilot suggested a solution that you knew to be incorrect? If so, could you describe how you addressed this issue and what steps you took to ensure the correctness of the code?
Proper Trust
Could you describe your approach to balancing the use of Copilot for assistance and your own debugging skills to ensure the correctness of the code? How do you determine when to rely on Copilot and when to trust your own abilities?

[G01] Despite being aware of the potential for bugs in Copilot-generated code, what motivates you to continue using the tool during debugging? Could you describe the benefits that Copilot provides and how it enhances your overall debugging process?