42 Lecture - CS504

Important Subjective

1. What is debugging, and why is it an important process in software development?

Answer: Debugging is the process of identifying and fixing defects or errors in software code. It is crucial as it ensures the software runs correctly and efficiently, delivering a reliable and bug-free product.

2. Describe the steps you would take to debug a software issue reported by a user.

Answer: The steps would typically involve reproducing the issue, analyzing logs or error messages, using a debugger to trace the code, identifying the root cause, and implementing the necessary fix.

3. What are the common types of software defects encountered during debugging?

Answer: Common types include syntax errors, runtime errors, and logic errors. Syntax errors result from incorrect code structure, while runtime errors occur during code execution. Logic errors lead to incorrect program behavior.

4. How does a debugger help in the debugging process?

Answer: A debugger allows developers to inspect the code execution, set breakpoints to pause the program, check variable values, and trace program flow, enabling them to identify and resolve defects efficiently.

5. What is the role of unit testing in debugging?

Answer: Unit testing helps identify defects at the unit level, ensuring that individual components work as expected. It aids in detecting issues early in the development process, making debugging more manageable.

6. What strategies can you use to effectively debug a complex issue in the code?

Answer: Strategies include dividing the problem into smaller parts, using logging to track code flow and variable values, and consulting with other team members for fresh perspectives.

7. Explain the difference between debugging in a development environment and debugging in a production environment.

Answer: Debugging in a development environment involves using a debugger and tools to trace code and identify defects, whereas debugging in a production environment requires analyzing logs and error reports without disrupting the live system.

8. How does code refactoring contribute to the debugging process?

Answer: Code refactoring improves code readability and maintainability, making it easier to debug. By restructuring the code, it reduces the likelihood of defects and aids in identifying and fixing issues.

9. Why is it essential to thoroughly test the fixed code after debugging?

Answer: Thorough testing ensures that the defect has been successfully resolved and that no new issues have been introduced during the debugging process, providing confidence in the code's stability.

10. How can you prevent recurring defects and minimize the need for debugging in future releases?

Answer: To prevent recurring defects, conduct code reviews, adopt best coding practices, use automated testing, and maintain proper documentation. Continuous improvement helps minimize the need for debugging in the future.