

CS-636: Computer Science

Practice Exam

Generated on June 27, 2025

Difficulty: Medium

Important Notes:

- Answer all questions to the best of your ability
 - Show all work for partial credit
 - Use additional paper if needed
 - Time limit: 90 minutes
-

1. Explain the difference between a compiler and an interpreter, providing examples of programming languages that utilize each and discussing the advantages and disadvantages of each approach in terms of execution speed and debugging.

2. Describe the concept of a deadlock in concurrent programming. Explain the four necessary conditions for a deadlock to occur and provide a specific example scenario involving two threads competing for resources. How can deadlocks be prevented or resolved?

3. What is the purpose of a hash table (hash map)? Explain how hash functions and collision handling mechanisms contribute to its efficient operation. Discuss the time complexity of common hash table operations (insertion, deletion, search) in the average and worst cases.

4. Compare and contrast object-oriented programming (OOP) and procedural programming paradigms. Provide examples of programming languages that exemplify each paradigm, highlighting the strengths and weaknesses of each approach for specific types of programming tasks (e.g., game development versus data analysis).

5. Describe the functionality of a binary search tree (BST). Explain how insertion and search operations work in a BST, and discuss the time complexity of these operations in the best, average, and worst-case scenarios. How can self-balancing algorithms improve the performance of a BST?

6. What are the key differences between IPv4 and IPv6 addressing schemes? Discuss the reasons for the transition from IPv4 to IPv6, including the limitations of IPv4 and the advantages offered by IPv6. Explain the concept of subnetting in the context of IP addressing.

7. Explain the concept of a software design pattern, giving examples of at least three common design patterns (e.g., Singleton, Factory, Observer). For each example, describe its purpose, structure, and when it would be an appropriate solution to a software design problem.

8. Describe the role of a relational database management system (RDBMS). Explain the concepts of normalization and ACID properties in the context of database design and transaction management. Discuss the advantages and disadvantages of using an RDBMS compared to other data storage solutions, such as NoSQL databases.