

## Oral Exam Version A

### Conceptual Questions (20 pts each)

1. Describe how collisions are handled in a hash map that uses a fixed-size array with linked list chains. Why is this method used, and what are its trade-offs?
2. Explain what properties a good hash function should have. How might poor hash function design affect the performance of a hash map?
3. Analyze the best-case and worst-case time complexity of searching for a key in a hash map. What conditions lead to each?

### Applied Use Case (40 pts)

Imagine you're designing an app for scheduling tutoring appointments in the QUAD. Would a hash map be a good choice for organizing appointments by student ID? Justify your answer.

## Oral Exam Version B

### Conceptual Questions (20 pts each)

1. What are the consequences of a poor hash function that results in clustering of keys in a few buckets?
2. How does resizing a hash map affect performance, and when should resizing be triggered?
3. As a hash map grows, how does the load factor affect the performance of operations?

### Applied Use Case (40 pts)

Suppose you're building a lost-and-found database for Willamette University where items are stored by location found (e.g., Sparks, Ford, Montag). What data structure(s) would you use to implement this and why?

## Oral Exam Version C

### Conceptual Questions (20 pts each)

1. Describe the trade-offs between using a hash map versus a linked list for storing key-value pairs.
2. What happens when two different keys hash to the same index? How is the lookup affected in a hash map with chaining?
3. Why is the load factor important in hash map design? How does it relate to performance?

### Applied Use Case (40 pts)

You're creating a meal-tracking app for Goudy that maps student IDs to their recent meal check-ins. Explain what data structure(s) you would use and how you would implement it. Discuss any potential challenges or limitations of your design.

## Oral Exam Version D

### Conceptual Questions (20 pts each)

1. If the hash function maps all keys to the same index, what is the resulting time complexity of operations? How can this be avoided?
2. Why might linked lists be chosen for chains in a hash map rather than arrays or sets?
3. What is the role of modular arithmetic in implementing a hash function? What could go wrong?

### Applied Use Case (40 pts)

If you were managing a campus-wide club registry app, where club names map to student leader contact info, how could a hash map help or hurt the system design?

## Oral Exam Version E

### Conceptual Questions (20 pts each)

1. Explain how a hash map differs from an array and a linked list in terms of indexing and lookup efficiency.
2. Under what conditions might a hash map become inefficient compared to a sorted array?
3. What steps are taken when inserting into an occupied bucket in a hash map with chaining? Describe the process.

### Applied Use Case (40 pts)

You are designing a system to manage students and faculty advisors. You are selecting data structures for the system to store and retrieve the students and the faculty advisors. The desired system should allow for efficient retrieval of a faculty advisor given a student ID. It should also allow for efficient retrieval of all students assigned to a faculty advisor. What data structure(s) would you choose for this task and why?