## Map ADT
8.1: Design and document algorithms that use the Map abstract data type  
Mean: 95.00%  
Progress: 100.00%

8.2: Describe unordered list-based data structures that implement the Map abstract data type  
Mean: 88.80%  
Progress: 76.00%

8.3: Describe ordered list-based data structures that implement the Map abstract data type  
Mean: 85.26%  
Progress: 73.46%

8.4: Analyze runtime performance of algorithms that use the Map abstract data type  
Mean: 80.00%  
Progress: 83.33%

## Tree & Binary Tree ADTs
9.1: Design and document algorithms that use the Tree abstract data type  
Mean: 89.70%  
Progress: 87.57%

9.2: Describe linked data structures that implement the Tree abstract data type  
Mean: 82.57%  
Progress: 95.24%

9.3: Design and document algorithms that use the Binary Tree abstract data type  
Mean: 100.00%  
Progress: 100.00%

9.4: Describe linked data structures that implement the Binary Tree abstract data type  
Mean: 79.44%  
Progress: 100.00%

9.5: Describe array-based data structures that implement the Binary Tree abstract data type  
Mean: 100.00%  
Progress: 100.00%

9.6: Describe the algorithms for preorder, postorder, inorder, and levelorder traversals  
Mean: 91.33%  
Progress: 100.00%

9.7: Classify binary trees as proper, perfect, and/or complete  
Mean: 91.33%  
Progress: 100.00%

9.8: Analyze runtime performance of algorithms that use the Tree abstract data type  
Mean: 78.33%  
Progress: 100.00%

9.9: Analyze runtime performance of algorithms that use the Binary Tree abstract data type  
Mean: 96.05%  
Progress: 100.00%

## Binary Search Trees
10.1: Insert entries into a binary search tree  
Mean: 90.47%  
Progress: 100.00%

10.2: Lookup entries within a binary search tree  
Mean: 98.62%  
Progress: 100.00%

10.3: Remove entries from a binary search tree  
Mean: 90.95%  
Progress: 100.00%

10.4: Describe the relationship between the number of nodes in a binary search tree and the height of the binary search tree  
Mean: 90.44%  
Progress: 100.00%

10.5: Describe how to rotate nodes in a binary search tree  
Mean: 90.10%  
Progress: 80.00%

## AVL Trees
11.1: Describe the height-balance property for AVL trees  
Mean: 98.28%  
Progress: 100.00%

11.2: Insert entries into an AVL tree  
Mean: 98.88%  
Progress: 100.00%

11.3: Lookup entries within an AVL tree  
Mean: 98.62%  
Progress: 100.00%

11.4: Remove entries from an AVL tree  
Mean: 100.00%  
Progress: 90.00%

11.5: Describe the relationship between the number of nodes in an AVL tree and the height of the AVL tree  
Mean: 98.84%  
Progress: 50.00%

## Splay Trees
12.1: Describe the splay operation when working with splay trees  
Mean: 96.55%  
Progress: 100.00%
12.2: Insert entries into a splay tree

MEAN=79.70%

12.3: Lookup entries within a splay tree

MEAN=72.77%

12.4: Remove entries from a splay tree

MEAN=85.88%

12.5: Explain the amortized cost of insert, lookup, and delete in a splay tree

MEAN=87.11%

12.6: Describe the relationship between the number of nodes in a splay tree and the height of the splay tree

MEAN=71.55%

(2,4) Trees

13.1: Describe the differences among 2-nodes, 3-nodes, and 4-nodes

MEAN=91.44%

13.2: Insert entries into a (2,4) Tree

MEAN=92.89%

13.3: Lookup entries within a (2,4) Tree

MEAN=94.39%

13.4: Remove entries from a (2,4) Tree

MEAN=84.40%

13.5: Describe the relationship between the number of nodes in a (2,4) tree and the height of the (2,4) tree

MEAN=71.66%

Red-Black Trees

14.1: Explain the properties of a red-black tree – root property, leaf property, red property, black-depth property

MEAN=96.67%

14.2: Lookup entries within a red-black tree

MEAN=98.62%

14.3: Remove entries from a red-black tree

MEAN=74.66%

14.4: Describe the relationship between the number of nodes in a red-black tree and the height of the red-black tree

MEAN=75.00%

Hash Functions

15.1: Explain types of hash codes, including additive hashing, polynomial hashing, and cyclic shifting

MEAN=95.69%

15.2: Explain types of compression functions, including the division method, multiply-and-divide method, and golden ratio method

MEAN=85.06%

15.3: Choose appropriate hash functions to minimize collisions

MEAN=79.31%

Hash Tables & Collision Resolution

16.1: Describe chaining strategies for collision resolution, including separate chaining and coalesced chaining

MEAN=91.27%

16.2: Describe open-addressing strategies for collision resolution, including linear probing and double hashing

MEAN=91.33%

16.3: Calculate load factor for a given hash table

MEAN=96.55%

16.4: Describe the performance of hash tables that use chaining and open addressing strategies

MEAN=89.39%

16.5: Explain how to delete an entry from a hash table

MEAN=97.81%
Map ADT

**Objective 8.1:** Design and document algorithms that use the Map abstract data type

- Written Workshop Activity Questions
  - MEAN=95.00%
  - 100.00%

**Objective 8.2:** Describe unordered list-based data structures that implement the Map abstract data type

- In-class Exercises
  - MEAN=90.00%
  - 50.00%
- Written Workshop Activity Questions
  - MEAN=87.94%
  - 93.33%

**Objective 8.3:** Describe ordered list-based data structures that implement the Map abstract data type

- In-class Exercises
  - MEAN=78.33%
  - 50.00%
- Written Workshop Activity Questions
  - MEAN=88.03%
  - 82.85%

**Objective 8.4:** Analyze runtime performance of algorithms that use the Map abstract data type

- In-class Exercises
  - MEAN=88.33%
  - 100.00%
- Written Workshop Activity Questions
  - MEAN=75.43%
  - 75.00%
Tree & Binary Tree ADTs

**Objective 9.1:** Design and document algorithms that use the Tree abstract data type

- Written Workshop Activity Questions
  - MEAN=89.70%
  - 87.57%

**Objective 9.2:** Describe linked data structures that implement the Tree abstract data type

- In-class Exercises
  - MEAN=86.65%
  - 100.00%

- Written Workshop Activity Questions
  - MEAN=81.07%
  - 92.86%

**Objective 9.3:** Design and document algorithms that use the Binary Tree abstract data type

- Written Workshop Activity Questions
  - MEAN=100.00%

**Objective 9.4:** Describe linked data structures that implement the Binary Tree abstract data type

- In-class Exercises
  - MEAN=85.55%
  - 100.00%

- Written Workshop Activity Questions
  - MEAN=83.68%
  - 100.00%

**Objective 9.5:** Describe array-based data structures that implement the Binary Tree abstract data type

- In-class Exercises
  - MEAN=85.55%
  - 100.00%

- Written Workshop Activity Questions
  - MEAN=73.33%
  - 100.00%

**Objective 9.6:** Describe the algorithms for preorder, postorder, inorder, and levelorder traversals

- In-class Exercises
  - MEAN=89.17%
  - 100.00%

- Written Workshop Activity Questions
  - MEAN=100.00%

**Objective 9.7:** Classify binary trees as proper, perfect, and/or complete

- In-class Exercises
  - MEAN=91.33%
  - 100.00%

**Objective 9.8:** Analyze runtime performance of algorithms that use the Tree abstract data type

- Written Workshop Activity Questions
  - MEAN=78.33%
  - 100.00%

**Objective 9.9:** Analyze runtime performance of algorithms that use the Binary Tree abstract data type

- Written Workshop Activity Questions
  - MEAN=96.05%
  - 100.00%
Binary Search Trees

**Objective 10.1:** Insert entries into a binary search tree

- In-class Exercises: MEAN=91.11%
- Written Workshop Activity Questions: MEAN=96.55%

- In-class Exercises: 100.00%
- Written Workshop Activity Questions: 100.00%

**Objective 10.2:** Lookup entries within a binary search tree

- Written Workshop Activity Questions: MEAN=98.62%

- In-class Exercises: 100.00%
- Written Workshop Activity Questions: 100.00%

**Objective 10.3:** Remove entries from a binary search tree

- In-class Exercises: MEAN=93.33%
- Written Workshop Activity Questions: MEAN=94.57%

- In-class Exercises: 100.00%
- Written Workshop Activity Questions: 100.00%

**Objective 10.4:** Describe the relationship between the number of nodes in a binary search tree and the height of the binary search tree

- In-class Exercises: MEAN=98.33%
- Written Workshop Activity Questions: MEAN=96.46%

- In-class Exercises: 100.00%
- Written Workshop Activity Questions: 100.00%

**Objective 10.5:** Describe how to rotate nodes in a binary search tree

- In-class Exercises: MEAN=93.97%
- Written Workshop Activity Questions: MEAN=95.68%

- In-class Exercises: 75.00%
- Written Workshop Activity Questions: 100.00%
AVL Trees

Objective 11.1: Describe the height-balance property for AVL trees
In-class Exercises
MEAN=98.28%
100.00%

Objective 11.2: Insert entries into an AVL tree
In-class Exercises
MEAN=86.21%
100.00%
Written Workshop Activity Questions
MEAN=86.52%
100.00%

Objective 11.3: Lookup entries within an AVL tree
Written Workshop Activity Questions
MEAN=98.62%
100.00%

Objective 11.4: Remove entries from an AVL tree
In-class Exercises
MEAN=89.66%
100.00%
Written Workshop Activity Questions
MEAN=80.71%
80.00%

Objective 11.5: Describe the relationship between the number of nodes in an AVL tree and the height of the AVL tree
In-class Exercises
MEAN=93.10%
100.00%
Written Workshop Activity Questions
MEAN=28.57%
0.00%
Splay Trees

Objective 12.1: Describe the splay operation when working with splay trees
- In-class Exercises: MEAN=96.55%
- Written Workshop Activity Questions: 100.00%

Objective 12.2: Insert entries into a splay tree
- In-class Exercises: MEAN=77.59%
- Written Workshop Activity Questions: 100.00%

Objective 12.3: Lookup entries within a splay tree
- In-class Exercises: MEAN=58.62%
- Written Workshop Activity Questions: 100.00%

Objective 12.4: Remove entries from a splay tree
- In-class Exercises: MEAN=65.52%
- Written Workshop Activity Questions: 100.00%

Objective 12.5: Explain the amortized cost of insert, lookup, and delete in a splay tree
- In-class Exercises: MEAN=90.10%
- Written Workshop Activity Questions: 0.00%

Objective 12.6: Describe the relationship between the number of nodes in a splay tree and the height of the splay tree
- In-class Exercises: MEAN=90.10%
- Written Workshop Activity Questions: 0.00%
(2,4) Trees

**Objective 13.1:** Describe the differences among 2-nodes, 3-nodes, and 4-nodes

In-class Exercises

Written Workshop Activity Questions

**Objective 13.2:** Insert entries into a (2,4) Tree

In-class Exercises

Written Workshop Activity Questions

**Objective 13.3:** Lookup entries within a (2,4) Tree

Written Workshop Activity Questions

**Objective 13.4:** Remove entries from a (2,4) Tree

In-class Exercises

Written Workshop Activity Questions

**Objective 13.5:** Describe the relationship between the number of nodes in a (2,4) tree and the height of the (2,4) tree

In-class Exercises

Written Workshop Activity Questions
Red-Black Trees

Objective 14.1: Explain the properties of a red-black tree ~ root property, leaf property, red property, black-depth property

In-class Exercises: MEAN=96.67%

Objective 14.2: Lookup entries within a red-black tree

Written Workshop Activity Questions: MEAN=98.62%

Objective 14.3: Remove entries from a red-black tree

In-class Exercises: MEAN=76.44%

Written Workshop Activity Questions: MEAN=71.11%

Objective 14.4: Describe the relationship between the number of nodes in a red-black tree and the height of the red-black tree

In-class Exercises: MEAN=100.00%

Written Workshop Activity Questions: MEAN=50.00%
### Hash Functions

**Objective 15.1:** Explain types of hash codes, including additive hashing, polynomial hashing, and cyclic shifting

<table>
<thead>
<tr>
<th>In-class Exercises</th>
<th>Written Workshop Activity Questions</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEAN=84.89%</td>
<td>MEAN=87.41%</td>
</tr>
<tr>
<td>50.00%</td>
<td>100.00%</td>
</tr>
</tbody>
</table>

**Objective 15.2:** Explain types of compression functions, including the division method, multiply-and-divide method, and golden ratio method

<table>
<thead>
<tr>
<th>In-class Exercises</th>
<th>Written Workshop Activity Questions</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEAN=89.66%</td>
<td>MEAN=100.00%</td>
</tr>
<tr>
<td>100.00%</td>
<td>100.00%</td>
</tr>
</tbody>
</table>

**Objective 15.3:** Choose appropriate hash functions to minimize collisions

<table>
<thead>
<tr>
<th>In-class Exercises</th>
<th>Written Workshop Activity Questions</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEAN=64.44%</td>
<td>MEAN=87.50%</td>
</tr>
<tr>
<td>87.50%</td>
<td>100.00%</td>
</tr>
</tbody>
</table>
Objective 16.1: Describe chaining strategies for collision resolution, including separate chaining and coalesced chaining

Objective 16.2: Describe open-addressing strategies for collision resolution, including linear probing and double hashing

Objective 16.3: Calculate load factor for a given hash table

Objective 16.4: Describe the performance of hash tables that use chaining and open addressing strategies

Objective 16.5: Explain how to delete an entry from a hash table