

Name: _____ Date: _____

Handout: Java HashSets

Warm Up

Imagine you have a program that tracks attendees who check in for an event using a list. One individual mistakenly checks in twice – now you have a duplicate entry!

How can we ensure there are no duplicates at all?

Your ideas:

How might we solve this problem with an `ArrayList`?

Your ideas:

What is a Set?

Does it allow duplicates?

Does it maintain insertion order?

Example 1: Creating HashSet Objects

```
Set<String> attendees = new HashSet<>();  
// Adding items to the HashSet.  
attendees.add("Angel");  
attendees.add("Alice");  
attendees.add("Bob");  
attendees.add("Alice");  
  
System.out.println(attendees);
```

What will the output be?

Predicted Output:

Actual Output:

Example 2: Removing Duplicates

```
List<String> attendees = Arrays.asList("Angel", "Alice", "Bob", "Alice");  
Set<String> uniqueAttendees = new HashSet<>(attendees);  
  
System.out.println("Original: " + attendees);  
System.out.println("Unique: " + uniqueAttendees);
```

What will the output be?

Predicted Output:

Actual Output:

Useful HashSet Methods

Fill the method descriptions during the lesson:

<code>add(item)</code>	
<code>contains(item)</code>	
<code>remove(item)</code>	
<code>size()</code>	

Limitations and Alternatives

HashSets are _____. (ordered / unordered)

How many `null` values are allowed? _____.

To maintain insertion order, use _____ instead of **HashSet**.

To have duplicate values, use _____ instead of **HashSet**.

Additional Notes

Check for Understanding

What happens if I add the same element twice to a `HashSet`?

Which `Set` would I use if I wanted to maintain insertion order?

Other than the examples described in this lecture, what's one real-world example where you'd use a `Set` instead of a `List`? Share with the class after.

Real Life Uses

Fill in with examples provided by others during discussion at the end of class.

1.

2.

3.

What questions do you have?

1.

2.

3.
