7.4 Sets

A **Set** is a collection that cannot contain duplicate elements. It models the mathematical set abstraction.

One of the implementations of the Set is the **HashSet** class.

Example:

```
import java.util.HashSet;

public class MyClass {
 public static void main(String[ ] args) {
   HashSet<String> set = new HashSet<String>();
   set.add("A");
   set.add("B");
   set.add("C");
   System.out.println(set);
 }
}
//  Output: [A, B, C]
```

Note:

You can use the **size**() method to get the number of elements in the HashSet.

Q: What is the output of this code?

```
import java.util.HashSet;
class A {
  public static void main(String[ ] args) {
  HashSet<String> set = new HashSet<String>();
  set.add("A");
  set.add("B");
  set.add("C");
  System.out.println(set.size());
  }
}
```

LinkedHashSet

The HashSet class does not automatically retain the order of the elements as they're added. To order the elements, use a **LinkedHashSet**, which maintains a linked list of the set's elements in the order in which they were inserted.

What is hashing?

A hash table stores information through a mechanism called hashing, in which a key's informational content is used to determine a unique value called a hash code.
So, basically, each element in the HashSet is associated with its unique hash code.

Note:

You've learned about the various collection types that are available in Java, including **Lists**, **Maps**, and **Sets**. The choice of which one to use is specific to the data you need to store and manipulate.

Q: Which two of the following statements are true? [Select all that apply]

- ❏ Set can contain duplicate values
- ❏ HashSet does not retain order
- ❏ LinkedHashSet does not retain order
- ❏ Set contains only unique values