**Collections-Exceptions Interview   
Preparations   
Question**

### What are exceptions in java and how do you handle them?

Exceptions occur when there is an issue with the program. There are two types of exceptions: **Checked exceptions** (aka Compile time) and **Unchecked exceptions** (aka Runtime). All checked exceptions must be handled otherwise the program will not compile. Unchecked exceptions happen during runtime so if they occur the program will be interrupted during execution making these exceptions are more difficult to handle.

Exceptions can be handled using the try/catch block which lets you define the actions taken if specific exceptions occur. It is also possible to use the keyword throws in the method signature to allow the program to compile and accept that an exception may occur, but this is not handling the exception.

1. **Difference between Error and Exception in Java?**
   1. Both Error and Exception are derived from Throwable in Java.
   2. Error represent errors which generally cannot be handled. They are issues outside of the code
      1. For example: OutOfMemoryError, NoClassDefFoundError
   3. Exception represent problems coming from the code which can be handled.
      1. For example: IOException, NullPointerException
   4. Exceptions are divided in two categories checked and unchecked Exceptions. Checked Exceptions must be handled before execution, but Unchecked Exception occur during runtime
   5. Errors are something developer are not required to do anything with
   6. In general Errors are beyond anyone’s control, but Exceptions can be guessed and handled
2. **Difference between throw and throws in Java?**
   1. throw and throws are two keywords related to Java Exceptions
   2. throw keyword is used to throw an exception explicitly and throws keyword is used to allow an exception to be thrown if it occurs
   3. throw keyword is used in a code block while throws is used in method signature to allow the exception.
   4. throw keyword can also be used to break a switch statement without using break keyword

### Difference between RuntimeException and CheckedException in Java?

* 1. Checked Exceptions must be handled before execution, but Unchecked Exception occur during runtime

Common exceptions: NullPointerExceptio, ArrayIndexOutOfBound, ClassNotFoundException, IOException.

1. **What is the collection framework in java?**

The collection framework is a group of interfaces and concrete classes that are different data structures used to solve different problem.

 **java.util.Collection** - interface which defines the basis actions of a collection

 **Set** *(Unique things)* - DOES NOT ALLOW DUPLICATES. Classes that Implement Set:

* **HashSet:** Used when you don't want any duplicates and you don't care about order when you iterate through
  + Unordered and Unsorted
* **LinkedHashSet:** Ordered version of HashSet and use over HashSet when you care about iteration order

### SortedSet

* **TreeSet**: Elements will be in ascending order, according to the natural order of the elements
  + Can also customize constructor to implement your own rules of the natural order

 **List** *(list of things)* - cares about the index. Classes that implement List:

* **ArrayList:** Fast iteration and fast random access and ordered (by index)
* **LinkedList:** Ordered by index position and elements are doubly-linked to one another
  + It is a good choice for implementing stack and queue
  + Iterates more slowly than ArrayList but fast insertion and deletion
* **Vector**: Same as ArrayList BUT vector methods are synchronized (thread-safe)

 **java.util.Collections** - a class that holds static utility methods for use with collections; Includes add, remove, contains, size, and iterator, etc.

1. **What are Maps?**

**Map** *(things with unique ID):*

Important: none of the Map-related classes and interfaces extend form Collection. The implementation classes of Map are thought of “collections”, not Collection. Classes that implement Map:

* **HashMap:**  Unsorted and unordered data structure that allows one null KEY and multiple null values in a collection. Works by entries, which are key/value pairs

### Hashtable

* + Same as HashMap BUT HashTable methods are synchronized (REMEMBER. ONLY METHODS ARE SYNCHRONIZED, NOT CLASSES OR VARIABLES)
  + Hashtable won't let you have anything NULL (NO NULLS AT ALL)

### LinkedHashMap

* + Maintains insertion order (or optionally, access order)
  + Slower than HashMap for adding/removing elements but FASTER ITERATION
* **SortedMap:**
  + TreeMap: Keys are sorting in natural order