**1). Describe the Collections type hierarchy, what are the main interfaces and what are the difference between them?**

In collection hierarchy it consist of several interfaces some main interface are Collection, List, set, Queue some other interfaces area sorted set, Dequeue, etc.

Collection:

is the basic and root interface of all collections.It is placed at the top of the collection hierarchy in java.It provides the basic operations for adding and removing elements in the collection. It extends the Iterable interface. The iterable interface has only one method called iterator(). The function of the iterator method is to return the iterator object. Using this iterator object, we can iterate over the elements of the collection. [List](https://www.scientecheasy.com/2020/09/java-list-interface.html/), Queue, and [Set](https://www.scientecheasy.com/2020/09/java-set.html/) have three component which extends the Collection interface.

List :

List elements are arranged sequentially ordered., List maintains an order of elements in which we add elements, and the same sequence we will get while retrieving elements. We can insert elements into the list at any location. The list allows to storing duplicate elements in Java. [Array List](https://www.scientecheasy.com/2020/09/arraylist-in-java.html/), [vector](https://www.scientecheasy.com/2020/09/vector-in-java.html/), and [Linked List](https://www.scientecheasy.com/2020/09/java-linkedlist.html/) are three concrete subclasses that implement the list interface.

Set Interface:

It is used to store the collection of unique elements, Set interface does not maintain any order while storing elements and while retrieving, we may not get the same order as we put elements.  All the elements in a set can be in any order. Set does not allow any duplicate elements. Hash Set, LinkedHashSet, Tree Set classes implements the set interface.

Queue Interface:

queue is an ordered of the homogeneous group of elements in which new elements are added at one end and elements are removed from the other end. This interface represents a special type of list whose elements are removed only from the head.

**2). Describe the various implementations of the map interface and their use case differences.**

Map and SortedMap is the interfaces in Map and it have three classes HashMap, LinkedHashMap, and TreeMap.

A Map doesnot allow duplicate keys, but you can have duplicate values we can go through the HashMap and LinkedHashMap allow null keys and values, but TreeMap doesn't allow any null key or value.

HashMap class implements the Map interface it allows us to store key and value pair, where keys should be unique. If you try to insert the duplicate key, it will be replace the element of the corresponding key. It is easy to perform operations using the key index like updation, deletion

LinkedHashMap class is Hashtable and Linked list implementation of the Map interface, with predictable iteration order. It inherits HashMap class and implements the Map interface.

Java TreeMap class is a red-black tree based implementation and it stores key-value pairs in sorted order.

**3).what are the difference between hashset and tree set.**

**Hashset:**

It does not provide a guarantee to sort the data.

only an element can be null.

faster than TreeSet. It allows only heterogeneous value.

**Treeset:**

It provides a guarantee to sort the data.

It does not allow null elements.

Slower then tree set. It allows only homogeneous value.

**4).how is hasmap implemented in java? How does its implememtaion use hash code and equals methods of objects ?**

**What is the time complexity of putting and getting an element from such structure?**

**5).what is fail fast and failsafe iterator?**

modification means adding, removing any element from collection while a thread is iterating over that collection. Iterator on ArrayList, HashMap classes are some examples of fail-fast Iterator.  
Fail-Safe iterators don’t throw any exceptions if a collection is structurally modified while iterating over it. This is because, they operate on the clone of the collection, not on the original collection and that’s why they are called fail-safe iterators.

**6).how to convert string to stream of chars?**

The method – chars() –is used to convert string to stream ofchars with which we can obtain an instance of *Stream* from a String object.

**IntStream** checkStringinput = CheckString.chars();

**7).explain spliterator?**

An object for traversing and partitioning elements of a source. The source of elements covered by a Spliterator could be, for example, an array, a [Collection](https://docs.oracle.com/javase/8/docs/api/java/util/Collection.html), an IO channel, or a generator function.A Spliterator may traverse elements individually.

**8).List the string operation with streams?**

Joining ,Splitting, Testing.

9). What is spring core?

The Spring Framework provides a comprehensive programming and configuration model for modern Java-based enterprise applications, Spring Framework is based on two design principles - Dependency Injection and Aspect Oriented Programming.

10).explain dependency injenction/ how to achive it?

Dependency injection is a programming technique that makes a class independent of its dependencies. It achieves that by decoupling the usage of an object from its creation. This helps you to follow SOLID’s dependency inversion and single responsibility principles.

Dependency injection is a programming technique that makes a class independent of its dependencies. It achieves that by decoupling the usage of an object from its creation. This helps you to follow SOLID’s dependency inversion and single responsibility principles.

11).how can we inject beans in a string?

12).can we have more then one spring configuration files in one project? How will spring get know whether it is configuration filw or component file or bean.

Those files have mentioned some annotations,spring is to identify those files searching like these annotations @configuration @component @bean .

13).explain string builder and string buffer.

[StringBuffer Class](https://www.geeksforgeeks.org/stringbuffer-class-in-java/): StringBuffer is a peer class of String that provides much of the functionality of strings. The string represents fixed-length, immutable character sequences while StringBuffer represents growable and writable character sequences.

[StringBuilder Class](https://www.geeksforgeeks.org/stringbuilder-class-in-java-with-examples/): Similar to StringBuffer, the StringBuilder in Java represents a mutable sequence of characters. Since the String Class in Java creates an immutable sequence of characters, the StringBuilder class provides an alternative to String Class, as it creates a mutable sequence of characters.

14). Difference between bean factory and application context?

Bean factory:

It is a fundamental container that provides the basic functionality for managing beans.

It is suitable to build standalone applications.

It supports only Singleton and Prototype bean scopes.

Application context:

It is an advanced container that extends the BeanFactory that provides all basic functionality and adds some advanced features.

It is suitable to build Web applications, integration with AOP modules, ORM and distributed applications.

It supports all types of bean scopes such as Singleton, Prototype, Request, Session etc.

15).explain

Spring AOP

Aspect oriented programming(AOP) as the name suggests uses aspects in programming. It can be defined as the breaking of code into different modules, also known as [modularisation](https://www.geeksforgeeks.org/modular-approach-in-programming/), where the aspect is the key unit of modularity.

IOC

Spring IoC (Inversion of Control) Container is the core of [Spring Framework](https://www.geeksforgeeks.org/introduction-to-spring-framework/). It creates the objects, configures and assembles their dependencies, manages their entire life cycle. The Container uses Dependency Injection(DI) to manage the components that make up the application.

POJO

POJO stands for Plain Old Java Object. It is an ordinary Java object, not bound by any special restriction other than those forced by the Java Language Specification and not requiring any classpath. POJOs are used for increasing the readability and re-usability of a program.

@Component

POJO stands for Plain Old Java Object. It is an ordinary Java object, not bound by any special restriction other than those forced by the Java Language Specification and not requiring any classpath. POJOs are used for increasing the readability and re-usability of a program.

@Componentscan

One of the most important annotations in spring is @ComponentScan which is used along with the @Configuration annotation to specify the packages that we want to be scanned. @ComponentScan without arguments tells Spring to scan the current package and all of its sub-packages.

@bean

One of the most important annotations in spring is the @Bean annotation which is applied on a method to specify that it returns a bean to be managed by Spring context. Spring Bean annotation is usually declared in Configuration classes methods.

16).

Answer

**public** **class** CountryCode {

**public** **static** **void** main(String[] args) {

Map<Integer,String> pa = **new** HashMap<Integer,String>();

pa.put(577201,"India");

pa.put(577301,"ShriLanka");

pa.put(577401,"Nepal");

pa.put(577418,"Japan");

pa.put(577427,"Buthan");

System.***out***.println("Lopping only Keys \n");

Collection<Integer> key =pa.keySet();

key.forEach(k-> System.***out***.println(k));

System.***out***.println("Lopping only Values \n");

Collection<String> values =pa.values();

values.forEach(v-> System.***out***.println(v));

System.***out***.println("ENTRIES\n");

Set<Entry<Integer, String>> entries = pa.entrySet();

**for**(Entry<Integer,String> entry:entries)

{

System.***out***.println(entry.getKey()+" : " +entry.getValue());

}

}

17).