**String**

In Java String is an Array of characters or object that represent sequence of char values.

String is a final class which implements **Serializable, Comparable<String>, CharSequence Interfaces**.

String class provide so many methods for perform an operations on strings such as concat(), compare(), equal() etc.

We can create a string using **String, StringBuffer, StringBuilder class** which implements **CharSequence interface**.

We can create String in Java by two way

1. Using String Literals – String str = “ABC”;
2. Using New Keyword – String str = new String(“ABC”);

Now Using Literals for creating Strings then it will store this String into String Constant Pool which is present in Heap memory.

String Constant Pool is memory area in Heap memory for storing a string literals.

Whenever we will create a string by literals the firstly JVM will check this String is present in String Constant Pool or not if.

If string is present in pool then reference is return for that instance.

If string is not present in pool then new string will create in this pool.

For Example

String str1 = “ABC”;

String str2 = “ABC”;

Here we create String str1, It JVM will first check in pool this “ABC” is present or not. Initially pool is empty the it will store this string in String constant pool.

Now we create String str2 now again JVM will check this “ABC” in pool but now this time this String is present in pool so it will return this reference to it.

There for those both are pointing to the same reference and because of that hashcode is same for both.

HashCode is an integer value that is associated with each object in Java.

Now if using new Keyword

String str1 = new String(“ABC”);

String str1 = new String(“ABC”);

Using new Keyword JVM will create new string object in heap memory and then literals will placed in poll.

This is known as Immutability of String.

Advantage of making String Immutable in Java is

1. Memory Efficiency.
2. Security for Data.

Why String class is final – because no one can override the method of string class.

Three ways a compare a string in Java.

1. equals() – compare the value of that string objects
2. == operator – it compare the reference of that object.
3. compareTo() – it will compare Lexicographically of strings and return 0 if equal, 1 if greater and -1 if smaller.

**toString()** Method – If we want to represent any object as string toString() method will return the string representation of object.

**StringBuffer and StrinBuilder**

Both those classes is use for creating mutable String object in Java.

So many methods are present in this classes for manipulating to Original String.

| **Feature** | **StringBuffer** | **StringBuilder** |
| --- | --- | --- |
| Thread Safety | Thread-safe | Not thread-safe |
| Synchronization | Methods are synchronized, leading to slower performance in multi-threaded environments | Not synchronized, faster performance in single-threaded environments |
| Mutable | Mutable (contents can be changed) | Mutable (contents can be changed) |
| Performance | Slower due to synchronization overhead | Faster due to lack of synchronization overhead |
| Introduced in | Introduced in Java 1.0 | Introduced in Java 5.0 |

**How to create Immutable Class in Java?**

To create an immutable object in Java, you should follow these steps:

Step 1: Declare all fields of the class as final and private.

Step 2: Initialize the fields in the constructor.

Step 3: Do not provide setter methods for changing the state of the object.

| **Comparable** | **Comparator** |
| --- | --- |
| Use for Single Sorting | Use for multiple Sorting |
| Having compareTo() method | Having compare() method |
| Affect original class | Doesn’t affect original class |
| Java.lang | Java.util |
| Collections.sort(List) | Collections.sort(List,Comparator) |