**Java\_String\_Concept**

**Q.1. Why are the object immutable in Java?**

Because Java uses the concept of String literal. Suppose there are 5 reference variables, all refer to one object "Hello". If one reference variable changes the value of the object, it will be affected by all the reference variables. That is why String objects are immutable in Java.

**Q.2. How to create a string object?**

1. By Literal 🡺 String s = “Hello”;

2. By new key word 🡺 String s = new String (“Hello”);

**Q.3. Why Java uses the concept of String Literal?**

To make Java more memory efficient (because no new objects are created if it exists already in the string constant pool).

**Q.4. Why Char array is preferred over String for storing password?**

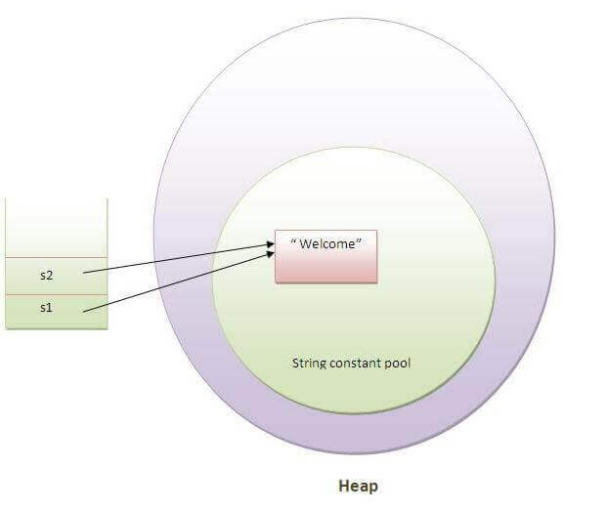
String is immutable in Java and stored in String pool. Once it’s created it stays in the pool until unless garbage collected, so even though we are done with password it’s available in memory for longer duration and there is no way to avoid it. It’s a security risk because anyone having access to memory dump can find the password as clear text.  
If we use a char array to store password, we can set it to blank once we are done with it. So, we can control for how long it’s available in memory that avoids the security threat with String.

**Q.5.** **How do you check if two Strings are equal in Java?**

There are two ways to check if two Strings are equal or not – using “==” operator or using equals method. When we use “==” operator, it checks for the value of String as well as the reference but in our programming, most of the time we are checking equality of String for value only. So, we should use the equals method to check if two Strings are equal or not.  
There is another function equalsIgnoreCase that we can use to ignore case.

**Q.6. What is String pool in Java?**

String pool is the space reserved in the heap memory that can be used to store the strings. The main advantage of using the String pool is whenever we create a string literal; the JVM checks the "string constant pool" first. If the string already exists in the pool, a reference to the pooled instance is returned. If the string doesn't exist in the pool, a new string instance is created and placed in the pool. Therefore, it saves the memory by avoiding the duplicacy.



**Q.7. What does String intern () do?**

If we want to store any heap object inside the String Constant Pool Area forcefully then we will use intern () method. This method ensures that all the same Strings share the same memory. It can be used to return string memory, if it is created by new keyword. I t creates exact copy of heap string object in string constant pool. For example, creating string “Hello” 10 times using intern () method would ensure that there will be only one instance of “Hello” in the memory and all the 10 references point to the same instance.

**Q.8. What does String valueOf () method do?**

The String class valueOf () method converts given type such as int, long, float, double, Boolean, char and char array into String.

**Q. 9. What does String split () method do?**

The Java string split () method splits the given string against given delimiter/regular expression and **returns a char array.**

**Q. 10. What does String trim () method do?**

The String trim () method eliminates white spaces before and after string.

**Q. 11. What does String index Of () method do?**

The Java String index Of () method returns index of given character value or sub-string.

**Q.12. What does String charAt () method do?**

charAt () returns a char value at the given index number. Index number start from 0 and goes to n-1. n is the length of the string.

**Q. 12. What are ways to compare string in Java?**

There are three ways to compare string in java:

1. By equals () method ==> compare values of the string are equal or not.
2. By = = operator and ==> compare the reference of the string
3. By compare To () method ==> The Java String compare To () method is used for comparing two strings lexicographically.

Each character of both the strings is converted into a Unicode value for comparison.

If both the strings are equal then this method returns 0 else it returns positive or negative value.

The result is positive if the first string is lexicographically greater than the second string else the result would be

negative.

**Q. 13. Difference between String and String Buffer?**

|  |  |
| --- | --- |
| **String** | **String Buffer** |
| String is immutable | String Buffer is mutable. |
| String class uses string constant pool. | String Buffer uses heap memory. |
| Thread-safe | Thread-safe. |
| Provides synchronization. | Provide synchronization. |
| String is slow and consumes more memory when we concatenate too many strings because every time it creates new instance. | String Buffer is fast and consumes less memory when concatenate too many strings. |
| Object of String class is of fixed length i.e., read-only in nature. | The length of the String Buffer object can be increased when required i.e., dynamic in nature. |

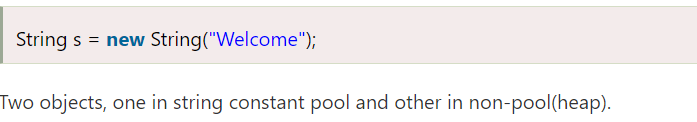
**Q. 14. Difference between String Buffer and String Builder?**

|  |  |
| --- | --- |
| **String Buffer** | **String Builder** |
| **String Buffer is synchronized i.e., thread safe. It means two threads can’t call the methods of String Buffer simultaneously.** | **String Builder is non-synchronized i.e., not thread safe. It means two threads can call the methods of String Builder simultaneously.** |
| String Buffer is less efficient than String Builder. | String Builder is more efficient than String Buffer. |
| String Buffer was introduced in Java 1.0. | String Builder was introduced in Java 1.5. |

**Q.15. Does String is thread-safe in Java?**

Since String is immutable in Java, it’s inherently thread-safe.

**Q.16. How many objects will be created in the following case?**



In such case, [JVM](https://www.javatpoint.com/jvm-java-virtual-machine) will create a new string object in normal (non-pool) heap memory, and the literal "Welcome" will be placed in the string constant pool. The variable s will refer to the object in a heap (non-pool).

### **Q. 17. What is the purpose of toString () method in Java?**

The toString () method returns the string representation of an object. If you print any object, java compiler internally invokes the toString () method on the object. So, overriding the toString () method, returns the desired output, it can be the state of an object, etc. depending upon your implementation. By overriding the toString () method of the Object class, we can return the values of the object, so we don't need to write much code.