1. What is the difference between declaring string as a literal and string as an object?

(A) String s = “Code Chef”; -> This is declaration of a string literal.

Whenever we declare string as a literal it calls an intern() method of String class where a string pool exists internally. It checks in the constant pool for the literal if it exists already. If it does, it will bind the literal to the same object rather than creating a new one.

String s = new String(“Code Chef”); -> This is a declaration of a string object.

Whenever a String object is created it will always create a new object even if there is an object with same literal.

String object creation is often referred as slower process than String literal because everytime there is new object created.

1. Why is output “True” for both the codes mentioned below?

String s1 = “hello”;

String s = “hello”;

System.out.println(s1.equals(s)); -> True

System.out.println(s1==s); ->True

(A) Firstly, s and s1 both are string literals.

In First Statement, .equals() method checks for the content of both the object references.

Since they are same, that’s why it is True.

In Second Statement, ‘==’ ooperator checks for the equivalency of reference. Since they both are String literals and they have same content, that’s why they point to same reference.

So, it is True.

1. What is the purpose of replaceAll() method of String class?

(A) The replaceAll() method replaces each substring of the String on which it is called if it matches the regular expression by the replacement String.

1. What is the usage of matches() method in String Class?

(A) The matches() method in java is used for checking if the regular expression passed matches the string or not. It return Boolean values.

1. What is intern() method in java?

(A)It returns the pooled string from the memory. It usually checks if the string content is same for the literals.

1. What is the purpose of subsequence method?

(A) The **Java.lang.String.subSequence()** is a built-in function in Java that returns a CharSequence. CharSequence that is a subsequence of this sequence. The subsequence starts with the char value at the specified index and ends with the char value at (end-1). The length (in chars) of the returned sequence is (end-start, so if start == end then an empty sequence is returned.

**Syntax: public CharSequence subSequence(int start, int end)**

**Parameters:**

**start** - This is the index from where the subsequence starts, it is inclusive.

**end** - This is the index where the subsequence ends, it is exclusive.