**How String and StringBuffer differs?**

**Refer StringStringBufferTest.java**

**Case 1**

**String s = new String(“suganthna”); | StringBuffer sb = new SB(“suganthan”)**

**s.concat(“madhavan”) | sb.append(“madhavan”)**

**sop(s) //suganthan | sop(sb) //suganthan madhavan**

Once we create a String object we can’t perform any changes in the existing object. If we are trying to make changes a new object will be created(immutability).

**How many objects will be created when String s = new String(“Suganthan”) is executed?**

Two object will be created one at heap and another at String constant pool and **s is always pointing to heap object**.

**What and how many object will be created when String s = “Suganthan” is executed?**

One object will be created at String constant pool and s is point is SCP

**How many objects will be created when the below line of code executed?**

String s1 = new String(“Suganthan”);

String s2 = new String(“Suganthan”);

String s3 = “Suganthan”;

String s4 = “Suganthan”;

**In total 3 objects will be created 2 in heap and one in SCP.**

**How many objects will be created when the below line of code executed?**

String s1 = new String(“Suganthan”); //2 objects

s1.concat(“Software”); //2 objects

String s2 = s1.concat(“engineer”); //2 objects

s1 = s1.concat(“blore”); //2 objects

**List down all the constructor of String class?**

String s = new String();

String s = new String(String literal);

String s = new String(StringBuffer sb);

String s = new String(char[] chArray);

String s = new String(byte[] byArray);

**What will happen due to runtime change in the content?**

Because of runtime operation if there is a change in the content, then with those changes a new object will be create at the heap. If their is no change in the content, the existing object will be re-used and new object wont be created

Whether the object present in heap or SCP, if there is change in content new object will be created or existing object will be re-used.

**Refer StringRuntimeTest.java for more details**

**What is the difference between Final and Immutable?**

Final Objects we can’t perform re-assignment. Whereas immutabe objects means actual objects value can’t changed

Final applicable for variables but not for objects. Whereas immutability applicable for objects but not for variables.

By declaring reference variable as final we won’t get any immutability nature. Even though reference variable is the final, we can perform any type of change in the corresponding object but we can’t perform reassignment for that variable.

**Refer FinalTest.java for more details**

**What are wrapper class in Java?**

The objective of we can define several utility method which are required for primitives.

**Constructors of Wrapper class?**

Almost all wrapper class contain two constructor, one can take corresponding primitive as argument and other can take String as argument.

Integer integer = new Integer(10);

Integer integer = new Integer(“10”);

**What is method used to create wrapper class from primitive type or String?**

**valueOf(String str);**

**valueOf(primitive);**

**valueOf(String str, int radix);**

Integer integer = Integer.*valueOf*(10);

Integer integer1 = Integer.*valueOf*(**"10"**);

Integer integer2 = Integer.*valueOf*(**"10"**, 2);

Float flt = Float.*valueOf*(10.5f);

Float flt1 = Float.*valueOf*(**"10.5f"**);

System.***out***.println(flt+**", "**+flt1);

**Which method you can use to convert wrapper to primitive?**

We can use xxxValue() method

**String to primitive?**

ParseXXX?