* Question On String

1)What is String? String in java is an object that represent sequence of character value. String is a special class We can create object of string without new keyword. **Java String** class provides a lot of methods to perform operations on strings such as compare(), concat(), equals(), split(), length(), replace(), compareTo(), intern(), substring() etc The java.lang.String class is used to create a string object.

### 2)How to create a string object?

### There are two ways of creating string object

### 1)String Literal :

### String str=”My name is Aparna”; This is object of string by using string literal String literal is created by using double quote. Each time you 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

### Example= String s1=”Welcome”;

### Strings2=”Welcome”; //it does not create new object it just return

### //reference to the pool instance

### only one object will be created. Firstly, JVM will not find any string object with the value "Welcome" in string constant pool that is why it will create a new object. After that it will find the string with the value "Welcome" in the pool, it will not create a new object but will return the reference to the same instance.

### 2)By new keyword : whenever we create object of string using new keyword it always create object in heap memory area wheather it is duplicate or not and it will create object in string constant pool if three is no object is already created

### Q.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.What is string constant pool? **String objects are stored in a special memory area known as the "string constant pool". A string constant pool is a saperate place where the program are stored String Pool in Java is a special storage space in Java Heap memory where string literals are stored**.

**String Pool, also known as SCP (String Constant Pool), is a special storage space in Java heap memory that is used to store unique string objects. Whenever a string object is created, it first checks whether the String object with the same string value is already present in the String pool or not, and if it is available, then the reference to the string object from the string pool is returned. Otherwise, the new string object is added to the string pool, and the respective reference will be returned.**

**State the difference between the new operator and without a new operatorin a string.**  
**Ans:**  
String string=” Software Testingo”  
Whenever we create an object of String object with a double quote, it allocates the memory in the string literal pool in the indexing manner.  
String string=new String(“Software Testingo”);  
Whenever we create the String object with the new operator, it allocates memory for the String object in the heap area in the indexing manner.

Q.Why the string immutable in java?(Imp question)

In java string object are immutable means unmodifiable and changeable.Once the string object are created its states and value cant change. If we change value new objct will be created. String is immutable, **its value can't be changed otherwise any hacker could change the referenced value to cause security issues in the application**. Since String is immutable, it is safe for multithreading. String pool is possible because string is immutable and it increase security because any hacker cant change its value and it is used for storing it is used for storing sensitive information such as database username ND PASSWORD SINCE IT IS IMMUTABLE  The String is immutable in Java because of many reasons like security, caching, synchronization and concurrency, and class loading.

Ex: String s1=”sachin”; its value doesn’t change but if we want Sachin in capital letter then we will use toUppercase method it will store in another variable it will create new object

### Q.Why String class is Final in Java?

The reason behind the String class being final is because no one can override the methods of the String class. So that it can provide the same features to the new String objects as well as to the old ones.

### Q. **What is the difference between equals() and “==” in java?**

### == is used to check reference or memory address of the objects whether they point to the same location or not, and equals() method is used to compare the contents of the object

Q. In Java, how can two strings be compared?(Equal method)

**String Equals Method**: In this method, the strings are compared based on the values within them. If the values of the two strings are the same, it returns true; otherwise, it returns false. This method is case-sensitive

//The Java String class equals() method compares the two given strings

//based on the content of the string. If any character is not matched, it returns false.

//If all characters are matched, it returns true.

//The String equals() method overrides the equals() method of the Object class

Method of String

* **split()**: Split/divide the string at the specified regex.
* **compareTo()**: Compares two strings on the basis of the Unicode value of each string character.
* **compareToIgnoreCase()**: Similar to compareTo, but it also ignores case differences.
* **length()**: Returns the length of the specified string.
* **substring()**: Returns the substring from the specified string.
* **equalsIgnoreCase()**: Compares two strings ignoring case differences.
* **contains()**: Checks if a string contains a substring.
* **trim()**: Returns the substring after removing any leading and trailing whitespace from the specified string.
* **charAt()**: Returns the character at specified index.
* **toLowerCase()**: Converts string characters to lower case.
* **toUpperCase()**: Converts string characters to upper case.
* **concat()**: Concatenates two strings.

### Q. In Java, how do you convert a string to an integer and vice versa?

 The parseInt() method allows you to convert a String into an integer and the toString() method allows you to convert an Integer into a String

Q. **What is the difference between String and String Buffer in java?**

 The main difference between String and StringBuffer in Java is that the String is immutable while StringBuffer is mutable. This means you can modify a StringBuffer object once you have created it without creating any new objects.   
String class uses String constant pool. Wheareas StringBuffer uses Heap memory

**Q.What is StringBuffer Class?**

Java StringBuffer class is used to create mutable (modifiable) String objects. The StringBuffer class in Java is the same as String class except it is mutable i.e. it can be changed. Its value and state can be changed

#### **Java StringBuffer class is thread-safe i.e. multiple threads cannot access it simultaneously. So it is safe and will result in an order.**

Example: **public** **class** StringBufferEx {

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

StringBuffer s1=**new** StringBuffer("Hello");

s1.append("Java");//now original string is changed

System.***out***.println(s1);

}

}

### Q.What is a mutable String?

### A String that can be modified or changed is known as mutable String. StringBuffer and StringBuilder classes are used for creating mutable strings

### Q.What is StringBuilderClass?

Java StringBuilder class is used to create mutable (modifiable) String. The Java StringBuilder class is same as StringBuffer class except that it is non-synchronized. I

**Q.Difference between StringBuffer and StringBuilder?**

|  |  |  |
| --- | --- | --- |
| 1) | StringBuffer is *synchronized* i.e. thread safe. It means two threads can't call the methods of StringBuffer simultaneously. | StringBuilder is *non-synchronized* i.e. not thread safe. It means two threads can call the methods of StringBuilder simultaneously. |
| 2) | StringBuffer is *less efficient* than StringBuilder. | StringBuilder is *more efficient* than StringBuffer. |
| 3) | StringBuffer was introduced in Java 1.0 | StringBuilder was introduced in Java 1.5 |

Q.How to create an immutable class?

* declare the class as final so it cannot be extended
* all class members should be private so they cannot be accessed outside of class
* shouldn't contain any **setter** methods to change the value of class members
* the **getter** method should return the copy of class members
* class members are only initialized using constructor
* Example:
* **public** **class** ImmutableClassByFinal {
* **public** **static** **void** main(String[] args) {
* Immutable obj=**new** Immutable("Aparna",1992);
* System.***out***.println("Name: " + obj.getName());
* System.***out***.println("Date: " + obj.getDate());
* }
* }
* **final** **class** Immutable{
* // private class members
* **private** String name;
* **private** **int** date;
* **public** Immutable(String name,**int** date) {
* **this**.name=name;
* **this**.date=date;
* }
* // getter method returns the copy of class members
* **public** String getName() {
* **return** name;
* }
* **public** **int** getDate() {
* **return** date;
* }
* }

.