**List down Data Types in Java (with wrapper class name, memory size, range) in a tabular format.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Type** | **Wrapper Class** | **Byte** | **Range** |
| byte | Byte | 1 | Stores whole numbers from -128 to 127 |
| short | Short | 2 | Stores whole numbers from -32,768 to 32,767 |
| int | Integer | 4 | Stores whole numbers from -2,147,483,648 to 2,147,483,647 |
| long | Long | 8 | Stores whole numbers from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 |
|  |  |  |  |
| float | Float | 4 | Stores fractional numbers. Sufficient for storing 6 to 7 decimal digits |
| double | Double | 8 | Stores fractional numbers. Sufficient for storing 15 decimal digits |
|  |  |  |  |
| char | Character | 2 | Stores a single character/letter or ASCII values |
| String | String | Depend on value (Each character has 2 Bytes) |  |
| bool | Boolean | 1 bit | Stores true or false values |

**Write the concepts discussed about strings in the class**

**String** is a sequence of characters, for e.g. “Hello” is a string of 5 characters. In java, string is an immutable object which means it is constant and can cannot be changed once it has been created. In this tutorial we will learn about String class and String methods in detail along with many other Java String tutorials.

There are two ways to create a String in Java

1. String literal
2. Using new keyword

**Research and write what is String pool in java.**

String Pool is a storage area in Java heap.

String allocation, like all object allocation, proves to be a costly affair in both the cases of time and memory. The JVM performs some steps while initializing string literals to increase performance and decrease memory overhead. To decrease the number of String objects created in the JVM, the String class keeps a pool of strings.

Each time a string literal is created, the JVM checks the string literal pool first. If the string already exists in the string pool, a reference to the pooled instance returns. If the string does not exist in the pool, a new String object initializes and is placed in the pool.

After learning the concept theoretically, let me tell you how does a String pool work in Java step by step with the help of simple instances!

**How Does String pool work in Java?**

When you create a new string like this:

String s1 = "Rachel"

JVM automatically checks if the same value exists in the string constant pool or not.

* if yes, it occupies the already existing value.
* If no, it creates a new string by itself and adds it to the string pool.

If you want to halt this behavior, create a string using the new operator:

String s1 = new String("Rachel")

Now, if you are willing to add this string to the string literal pool, Java provides you with a method called, intern() method; you can call native intern() method like this:

S1.intern();

Now, I will show you the implementation and working of the string pool through an example.

But before that, a short reminder!

As you know if you’re comparing 2 objects using == operator it compares addresses in the memory.

So we will compare the strings using == to be completely sure that it’s the same object or not.

Now, let’s hop onto our implementation process.