# The String Class

String objects consist of a sequence of characters. However, the String class can be initialized like a primitive type, which we have done in previous lessons. For example:

final String USB\_PORT\_ADDRESS = "COM5";

is creating a String object.

A String object is immutable, meaning, once it is created, methods can’t be used to change them. You can however re-assign a String reference. In this case the String object is still immutable because the old reference has been discarded. For example:

String robot\_phrase = "I'm a robot";

robot\_phrase = "I'm the best robot";

You could also initialize a String object like has been done for other objects, such as:

String s1 = new String(); // An Empty Character Sequence

String s2 = new String("String Example");

In the example file, there are multiple instances of the plus symbol, “+”, used in System.out.println() method calls.

1. **Run the program and use the output to infer the functionality of the plus sign when used with Strings.**

|  |
| --- |

The String class also inherits the Object class’ methods like equals(). Comparing Strings with the equals method is different from the previously referenced equals operator “==”.

1. **Inspect the output of the program for the equals() method call outputs. In the line “USB\_PORT\_ADDRESS.equals(robotName)”, change the “USB\_PORT\_ADDRESS” to “ROBOT\_NAME” and rerun the program. What does the output tell you about the equals() method used on String objects?**

|  |
| --- |

1. **Add a new line to compare ROBOT\_NAME and robotName using the “==” operator. How does this affect the output? If you wanted to compare whether two String objects had the same contents, which would you use, the equals() method or the “==” operator?**

|  |
| --- |

As the example file shows, the String class has additional methods available for analyzing Strings. One such method, the substring() method, returns a subset of sequential characters from a String object.

The substring() method takes a starting index and non-inclusive ending index as inputs.

1. **Examine your robot’s name and add a new println() call to print the substring of the entire String. What happens if you try to increase the substring to use more characters then the String has?**

|  |
| --- |

The indexOf() method will return an index value of the first instance of a string in a string if detected, otherwise it will return -1.

Another String class method called length() returns the length of a String object.

1. **Write a System.out.println() line to display the length of your robot’s name.**

|  |
| --- |