```
package JAVAC1;
```

Strings:- The String is immutable in Java because of the security, synchronization and concurrency, caching, and class loading. The reason of making string final is to destroy the immutability and to not allow others to extend it. The String objects are cached in the String pool, and it makes the String immutable, it means we cannot concat another value in between the code to the same variable of that particular String Variable.

```
public class StringClass {
        //Examples
        public static void main(String[] args) {
                //Example 1
                String message = "Welcome";
                //here if we try to concatinate something in this message it will not be added.
                message.concat(" to Java");
                System.out.println(message);
                //Here you can the answer is printed only Welcome.
                //Example 2
                String newString = message.concat(" To Java");
                System.out.println(newString);
                //here the output is Welcom To Java because we store that concatinated string in another string so,
                //it's clear that we cannot concat in same variable.
```

String Builder:-The only difference is StringBuilder is mutable while String is immutable, means we can concat values in the same variable.

```
                //Example 1
                StringBuilder b = new StringBuilder("Welcome");
                b.append(" To Java");
                System.out.println("B is: "+b);
                //String can be write in two ways
                //1. String Literal
                //2. String Using new Operator
                //examples are given above.
```

```
	public void StringPrint(){
		String n = "Welcome";
		System.out.println(n);
	}

	public void exampleTwo() {
		String n2 = new String("Example2");
		System.out.println(n2);
	}

}
```