Comparing Strings (quoted text) in Java is somewhat strange.

To see this, create a Program whose "main" code is:

```java
while( true ) {
    out.print("Type a String: ");
    String data = cin.nextString();
    out.println("comparison = " + (data == "GOOD"));
    out.println("comparison equals = " + data.equals("GOOD"));
}
```

And run it - typing GOOD sometimes and other words other times. Carefully notice that the two outputs can be different.
The problem is that Java provides TWO comparison expressions:

First there is:

```
data == "GOOD"
```

Even when data refers to "GOOD" this expression will be FALSE when the two GOODs are created from different sources.

And then there is:

```
data.equals("GOOD")
```

But this expression WILL be true provided that the text message referred to by the variable data:

1. is exactly FOUR characters long
2. is ALL uppercase
3. has just the letters G, O, O, and D
4. in THAT order.
The moral of the story is:

Beginners should use equals to compare **Strings**

```
stringOne.equals(stringTwo)
```

But use `==` to compare **numbers**:

```
numberOne == numberTwo
```
The second moral of the story is:

**Beginners should use `!` with `equals` to dis-compare Strings**

**Note**

`!` is "NOT"

**But use `!=` to dis-compare numbers:**

```
numberOne != numberTwo
```