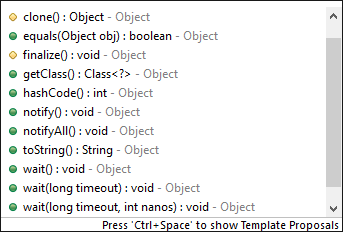
Object class:

 used in threading

It is part of java.lang package.

**toString():** this method return a string that is equals to

getClass().getName() + '@' + Integer.toHexString(hashCode())

This gives the fully qualified name of class concatenated with @ and hashcode

You can override the method in our class by following the overriding rules.

When you try to print the object, java by default appends toString() to it and executes it.

**Equals(Object obj):**

For a non null reference x and a reference variable y to compare they are equal or not we use equals method

**x.equals(y)**

Here X is non null reference because when a null reference compare with any reference it will generate false. The output of this method is Boolean value. If the two references equals then it return true else return false.

It is similar operation to == but == is applied on primitive data types and it compare the values stored in the identifiers. But when we applied on the reference variables it compare the values in the reference variables and reference values hold the address location where the values stored. So they never match each other because of address location until we assign same object to other reference. The equals() also compares the address stored in the reference variables

**String:**

By default every class has default packages and classes available to them without importing

Java.lang.\*;

Java.lang.Object;

Strings in java are immutable (we can’t modify the String objects)

The String reference variables are mutable (we can modify the reference variables)

Strings can be created in two ways

1. String literals ex: String s=”hello”;
2. Using new keyword ex: String s=new String();

JVM will create String poll of constants and update with String objects created through String literals.

Ex: Sting s1=”hello”

S1=”world”

When we create s1 initially with literal “hello” JVM check it is available on the String pool of constants if available the available address is assigned to s1 reference next time when we assign “world” literal then JVM check for it is not available then save in a location then the address is assigned to the s1.

Here the String is class which values is never be changed because world literal not replacing the hello literal. S1 is reference variable and its values i.e., address of the literals are modified.

With the help of new keyword also the same achieved but the String is created outside in the heap memory not in String pool of constants and they created like objects.

Sting class override the equals () by its own style. So when you use equals () on two string objects instead of comparing the references it compare the objects.

Assignment:

1. Bootstrap apply to register page, login page
2. Bootstrap to display table with 3 columns, and 10 rows even reduce the browser it should be generate a horizontal scroll bar not alter the table structure.