Object class in Java

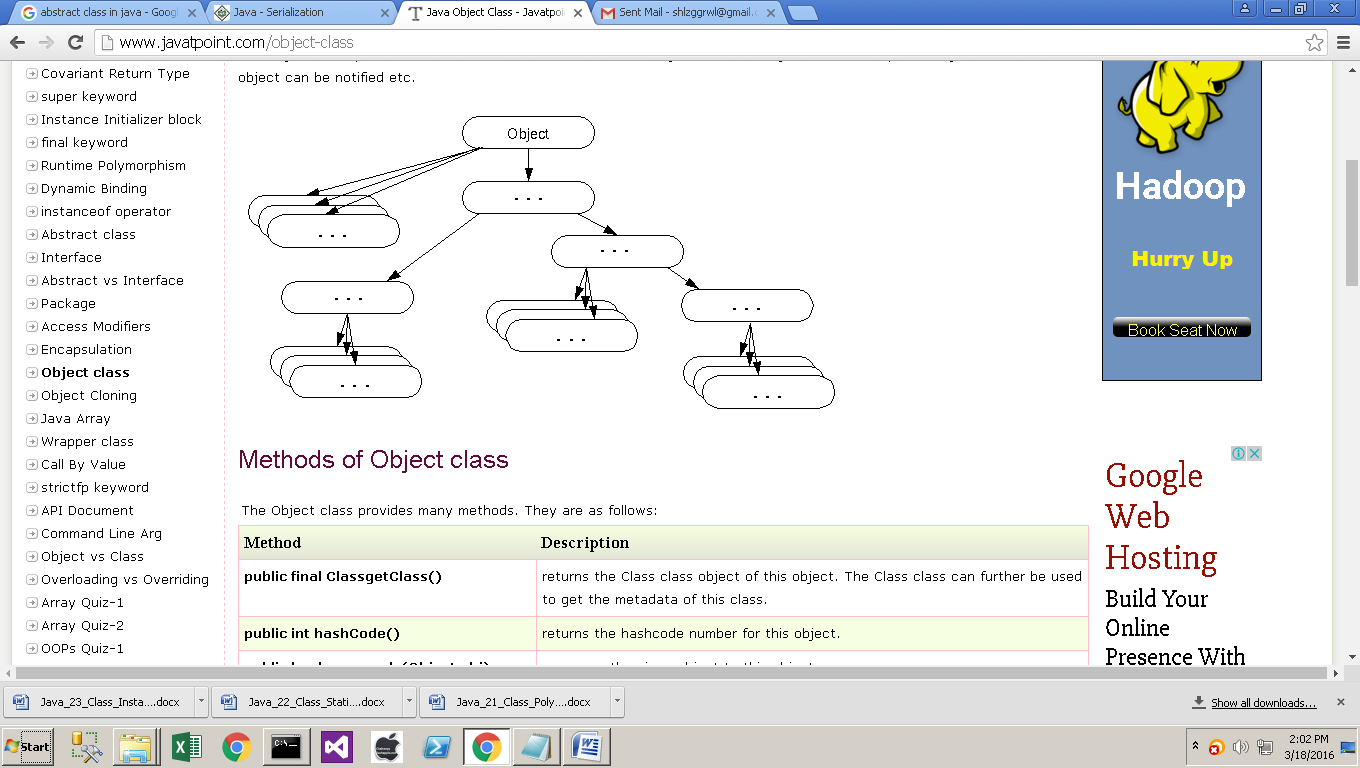
The Object class is the parent class of all the classes in java bydefault. In other words, it is the topmost class of java.

The Object class is beneficial if you want to refer any object whose type you don't know. Notice that parent class reference variable can refer the child class object, know as upcasting.

Let's take an example, there is getObject() method that returns an object but it can be of any type like Employee,Student etc, we can use Object class reference to refer that object. For example:

Object obj=getObject();//we don't what object would be returned from this method

The Object class provides some common behaviours to all the objects such as object can be compared, object can be cloned, object can be notified etc.



**Methods of Object class**

The Object class provides many methods. They are as follows:

|  |  |
| --- | --- |
| **Method** | **Description** |
| **public final ClassgetClass()** | returns the Class class object of this object. The Class class can further be used to get the metadata of this class. |
| **public int hashCode()** | returns the hashcode number for this object. |
| **public boolean equals(Object obj)** | compares the given object to this object. |
| **protected Object clone() throws CloneNotSupportedException** | creates and returns the exact copy (clone) of this object. |
| **public String toString()** | returns the string representation of this object. |
| **public final void notify()** | wakes up single thread, waiting on this object's monitor. |
| **public final void notifyAll()** | wakes up all the threads, waiting on this object's monitor. |
| **public final void wait(long timeout)throws InterruptedException** | causes the current thread to wait for the specified milliseconds, until another thread notifies (invokes notify() or notifyAll() method). |
| **public final void wait(long timeout,int nanos)throws InterruptedException** | causes the current thread to wait for the specified miliseconds and nanoseconds, until another thread notifies (invokes notify() or notifyAll() method). |
| **public final void wait()throws InterruptedException** | causes the current thread to wait, until another thread notifies (invokes notify() or notifyAll() method). |
| **protected void finalize()throws Throwable** | is invoked by the garbage collector before object is being garbage collected. |

**Object Cloning in Java**

The object cloning is a way to create exact copy of an object. For this purpose, clone() method of Object class is used to clone an object.

The java.lang.Cloneable interface must be implemented by the class whose object clone we want to create. If we don't implement Cloneable interface, clone() method generates CloneNotSupportedException.

The clone() method is defined in the Object class. Syntax of the clone() method is as follows:

protected Object clone() throws CloneNotSupportedException

**Why use clone() method ?**

The clone() method saves the extra processing task for creating the exact copy of an object. If we perform it by using the new keyword, it will take a lot of processing to be performed that is why we use object cloning.

**Advantage of Object cloning**

Less processing task.

**Example of clone() method (Object cloning)**

Let's see the simple example of object cloning

class Student18 implements Cloneable{

int rollno;

String name;

Student18(int rollno,String name){

this.rollno=rollno;

this.name=name;

}

public Object clone()throws CloneNotSupportedException{

return super.clone();

}

public static void main(String args[]){

try{

Student18 s1=new Student18(101,"amit");

Student18 s2=(Student18)s1.clone();

System.out.println(s1.rollno+" "+s1.name);

System.out.println(s2.rollno+" "+s2.name);

}catch(CloneNotSupportedException c){}

} }

**Output**:101 amit

101 amit

As you can see in the above example, both reference variables have the same value. Thus, the clone() copies the values of an object to another. So we don't need to write explicit code to copy the value of an object to another.

If we create another object by new keyword and assign the values of another object to this one, it will require a lot of processing on this object. So to save the extra processing task we use clone() method.