---

# Class java.lang.Object

```
java.lang.Object
```

---

public class **Object**

The root of the Class hierarchy. Every Class in the system has Object as its ultimate parent. Every variable and method defined here is available in every Object.

**See Also:**
Class
**Version:**
1.25, 08/18/95

---

## Constructor Index

o **Object**()

## Method Index

o **clone**()
Creates a clone of this Object.
o **copy**(Object)
Copies the contents of the specified Object into this Object.
o **equals**(Object)
Compares two Objects for equality.
o **getClass**()
Returns the Class of this Object.
o **hashCode**()
Returns a hashcode for this Object.
o **notify**()
Notifies a single waiting thread on a change in condition of another thread.
o **notifyAll**()
Notifies all of the threads waiting for a condition to change.
o **toString**()
Returns a String that represents the value of this Object.
o **wait**(long)

Causes a thread to wait until it is notified or the specified timeout expires.

- **wait**(long, int)
  More accurate wait.
- **wait**()
  Causes a thread to wait forever until it is notified.

# Constructors

- **Object**

```
public Object()
```

# Methods

- **getClass**

```
public final Class getClass()
```

Returns the Class of this Object. Java has a runtime representation for classes– a descriptor of type Class– which the method getClass() returns for any Object.

- **hashCode**

```
public int hashCode()
```

Returns a hashcode for this Object. Each Object in the Java system has a hashcode. The hashcode is a number that is usually different for different Objects. It is used when storing Objects in hashtables. Note: hashcodes can be negative as well as positive.

**See Also:**

Hashtable

- **equals**

```
public boolean equals(Object obj)
```

Compares two Objects for equality. Returns a boolean that indicates whether this Object is equivalent to the specified Object. This method is used when an Object is stored in a hashtable.

**Parameters:**

obj – the Object to compare with

**Returns:**

true if these Objects are equal; false otherwise.

**See Also:**

Hashtable

- **copy**

```
protected void copy(Object src)
```

Copies the contents of the specified Object into this Object. The contents of an Object are defined as the values of its instance variables. The parameter src must be of the same Class as this Object.

**Parameters:**

src – the Object whose contents are copied into the current object

**Throws:** ClassCastException

If obj is not of the same type as this Object.

**See Also:**

clone

o **clone**

```
protected Object clone()
```

Creates a clone of this Object. A new instance is allocated and the copy() method is called to copy the contents of this Object into the clone.

**Returns:**

a copy of this Object.

**Throws:** OutOfMemoryError

If there is not enough memory.

**See Also:**

copy

o **toString**

```
public String toString()
```

Returns a String that represents the value of this Object. It is recommended that all subclasses override this method.

o **notify**

```
public final void notify()
```

Notifies a single waiting thread on a change in condition of another thread. The thread effecting the change notifies the waiting thread using notify(). Threads that want to wait for a condition to change before proceeding can call wait().

*The method notify() can only be called from within a synchronized method.*

**Throws:** InternalError

If the current thread is not the owner of the Object's monitor.

**See Also:**

wait, notifyAll

o **notifyAll**

```
public final void notifyAll()
```

Notifies all of the threads waiting for a condition to change. Threads that are waiting are

generally waiting for another thread to change some condition. Thus, the thread effecting a change that more than one thread is waiting for notifies all the waiting threads using the method notifyAll(). Threads that want to wait for a condition to change before proceeding can call wait().

*The method notifyAll() can only be called from within a synchronized method.*
**Throws:** InternalError
If the current thread is not the owner of the Object's monitor.
**See Also:**
wait, notify

o **wait**

```
public final void wait(long timeout) throws InterruptedException
```

Causes a thread to wait until it is notified or the specified timeout expires.

*The method wait() can only be called from within a synchronized method.*
**Parameters:**
timeout – the maximum time to wait in milliseconds
**Throws:** InternalError
If the current thread is not the owner of the Object's monitor.

o **wait**

```
public final void wait(long timeout,
                       int nanos) throws InterruptedException
```

More accurate wait. *The method wait() can only be called from within a synchronized method.*
**Parameters:**
timeout – the maximum time to wait in milliseconds
nano – additional time, in nanoseconds range 0–999999
**Throws:** InternalError
If the current thread is not the owner of the Object's monitor.

o **wait**

```
public final void wait() throws InterruptedException
```

Causes a thread to wait forever until it is notified.

*The method wait() can only be called from within a synchronized method*
**Throws:** InternalError
If the current thread is not the owner of the Object's monitor.

---