

# Class `java.lang.Integer`

```
java.lang.Object
|
+----java.lang.Number
|
+----java.lang.Integer
```

---

public final class **Integer**  
extends [Number](#)

The Integer class is a wrapper for integer values. In Java, integers are not objects and most of the Java utility classes require the use of objects. Thus, if you needed to store an integer in a hashtable, you would have to "wrap" an Integer instance around it.

**Version:**

1.26, 10/04/95

**Author:**

Lee Boynton, Arthur van Hoff

---

## Variable Index

o **[MAX\\_VALUE](#)**

The maximum value an Integer can have.

o **[MIN\\_VALUE](#)**

The minimum value an Integer can have.

## Constructor Index

o **[Integer](#)**(int)

Constructs an Integer object initialized to the specified int value.

o **[Integer](#)**(String)

Constructs an Integer object initialized to the value specified by the String parameter.

# Method Index

- o **doubleValue()**  
Returns the value of this Integer as a double.
- o **equals()**(Object)  
Compares this object to the specified object.
- o **floatValue()**  
Returns the value of this Integer as a float.
- o **getInteger()**(String)  
Gets an Integer property.
- o **getInteger()**(String, int)  
Gets an Integer property.
- o **getInteger()**(String, Integer)  
Gets an Integer property.
- o **hashCode()**  
Returns a hashCode for this Integer.
- o **intValue()**  
Returns the value of this Integer as an int.
- o **longValue()**  
Returns the value of this Integer as a long.
- o **parseInt()**(String, int)  
Assuming the specified String represents an integer, returns that integer's value.
- o **parseInt()**(String)  
Assuming the specified String represents an integer, returns that integer's value.
- o **toString()**(int, int)  
Returns a new String object representing the specified integer in the specified radix.
- o **toString()**(int)  
Returns a new String object representing the specified integer.
- o **toString()**  
Returns a String object representing this Integer's value.
- o **valueOf()**(String, int)  
Assuming the specified String represents an integer, returns a new Integer object initialized to that value.
- o **valueOf()**(String)  
Assuming the specified String represents an integer, returns a new Integer object initialized to that value.

# Variables

## o MIN\_VALUE

```
public final static int MIN_VALUE
```

The minimum value an Integer can have. The lowest minimum value an Integer can have is 0x80000000.

## o MAX\_VALUE

```
public final static int MAX_VALUE
```

The maximum value an Integer can have. The greatest maximum value an Integer can have is 0x7fffffff.

## Constructors

### o Integer

```
public Integer(int value)
```

Constructs an Integer object initialized to the specified int value.

**Parameters:**

value – the initial value of the Integer

### o Integer

```
public Integer(String s) throws NumberFormatException
```

Constructs an Integer object initialized to the value specified by the String parameter. The radix is assumed to be 10.

**Parameters:**

s – the String to be converted to an Integer

**Throws:** NumberFormatException

If the String does not contain a parsable integer.

## Methods

### o toString

```
public static String toString(int i,  
                                int radix)
```

Returns a new String object representing the specified integer in the specified radix.

**Parameters:**

i – the integer to be converted

radix – the radix

**See Also:**

MIN\_RADIX, MAX\_RADIX

### o toString

```
public static String toString(int i)
```

Returns a new String object representing the specified integer. The radix is assumed to be 10.

**Parameters:**

i – the integer to be converted

**o parseInt**

```
public static int parseInt(String s,  
                           int radix) throws NumberFormatException
```

Assuming the specified String represents an integer, returns that integer's value. Throws an exception if the String cannot be parsed as an int.

**Parameters:**

s – the String containing the integer

radix – the radix to be used

**Throws:** NumberFormatException

If the String does not contain a parsable integer.

**o parseInt**

```
public static int parseInt(String s) throws NumberFormatException
```

Assuming the specified String represents an integer, returns that integer's value. Throws an exception if the String cannot be parsed as an int. The radix is assumed to be 10.

**Parameters:**

s – the String containing the integer

**Throws:** NumberFormatException

If the string does not contain a parsable integer.

**o valueOf**

```
public static Integer valueOf(String s,  
                               int radix) throws NumberFormatException
```

Assuming the specified String represents an integer, returns a new Integer object initialized to that value. Throws an exception if the String cannot be parsed as an int.

**Parameters:**

s – the String containing the integer

radix – the radix to be used

**Throws:** NumberFormatException

If the String does not contain a parsable integer.

**o valueOf**

```
public static Integer valueOf(String s) throws NumberFormatException
```

Assuming the specified String represents an integer, returns a new Integer object initialized to that value. Throws an exception if the String cannot be parsed as an int. The radix is assumed to be 10.

**Parameters:**

s – the String containing the integer

**Throws:** NumberFormatException

If the String does not contain a parsable integer.

**o intValue**

```
public int intValue()
```

Returns the value of this Integer as an int.

**Overrides:**

intValue in class Number

**o longValue**

```
public long longValue()
```

Returns the value of this Integer as a long.

**Overrides:**

longValue in class Number

**o floatValue**

```
public float floatValue()
```

Returns the value of this Integer as a float.

**Overrides:**

floatValue in class Number

**o doubleValue**

```
public double doubleValue()
```

Returns the value of this Integer as a double.

**Overrides:**

doubleValue in class Number

**o toString**

```
public String toString()
```

Returns a String object representing this Integer's value.

**Overrides:**

toString in class Object

**o hashCode**

```
public int hashCode()
```

Returns a hashCode for this Integer.

**Overrides:**

hashCode in class Object

**o equals**

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

Compares this object to the specified object.

**Parameters:**

obj – the object to compare with

**Returns:**

true if the objects are the same; false otherwise.

**Overrides:**

equals in class Object

**o getInteger**

```
public static Integer getInteger(String nm)
```

Gets an Integer property. If the property does not exist, it will return 0.

**Parameters:**

nm – the property name

**o getInteger**

```
public static Integer getInteger(String nm,  
                                int val)
```

Gets an Integer property. If the property does not exist, it will return val. Deals with Hexadecimal and octal numbers.

**Parameters:**

nm – the String name

val – the Integer value

**o getInteger**

```
public static Integer getInteger(String nm,  
                                Integer val)
```

Gets an Integer property. If the property does not exist, it will return val. Deals with Hexadecimal and octal numbers.

**Parameters:**

nm – the property name

val – the integer value