Class java.lang.Class

```
java.lang.Object
   |
   +----java.lang.Class
```

---

public final class **Class**
extends Object

Class objects contain runtime representations of classes. Every object in the system is an instance of some Class, and for each Class there is one of these descriptor objects. A Class descriptor is not modifiable at runtime.

The following example uses a Class object to print the Class name of an object:

```
void printClassName(Object obj) {
    System.out.println("The class of " + obj +
                       " is " + obj.getClass().getName());
}
```

Version:
1.25, 08/23/95

Method Index

o **forName**(String)
Returns the runtime Class descriptor for the specified Class.
o **getClassLoader**()
Returns the Class loader of this Class.
o **getInterfaces**()
Returns the interfaces of this Class.
o **getName**()
Returns the name of this Class.
o **getSuperclass**()
Returns the superclass of this Class.
o **isInterface**()
Returns a boolean indicating whether or not this Class is an interface.
o **newInstance**()
Creates a new instance of this Class.

o **toString**()
Returns the name of this Class or this interface.

Methods

o **forName**

```
public static Class forName(String className) throws ClassNotFoundException
```

Returns the runtime Class descriptor for the specified Class. For example, the following code fragment returns the runtime Class descriptor for the Class named java.lang.Thread:

```
Class t = Class.forName("java.lang.Thread")
```

Parameters:
className – the fully qualified name of the desired Class
Throws: ClassNotFoundException
If the Class could not be found.

o **newInstance**

```
public Object newInstance() throws InstantiationException, IllegalAccessException
```

Creates a new instance of this Class.
Returns:
the new instance of this Class.
Throws: InstantiationException
If you try to instantiate an abstract class or an interface, or if the instantiation fails for some other reason.
Throws: IllegalAccessException
If the class or initializer is not accessible.

o **getName**

```
public String getName()
```

Returns the name of this Class.

o **getSuperclass**

```
public Class getSuperclass()
```

Returns the superclass of this Class.

o **getInterfaces**

```
public Class[] getInterfaces()
```

Returns the interfaces of this Class. An array of length 0 is returned if this Class implements no interfaces.

o **getClassLoader**

```
public ClassLoader getClassLoader()
```

Returns the Class loader of this Class. Returns null if this Class does not have a Class loader.
See Also:
ClassLoader

o **isInterface**

```
public boolean isInterface()
```

Returns a boolean indicating whether or not this Class is an interface.

o **toString**

```
public String toString()
```

Returns the name of this Class or this interface. The word "class" is prepended if it is a Class; the word "interface" is prepended if it is an interface.
Overrides:
toString in class Object