# NSObject

**Inherits From:**        NSObject is the root class

**Conforms To:**        NSCoding
                                      NSObject

**Declared In:**        foundation/NSObject.h
                                      foundation/NSRunLoop.h


## Initializing the Class

+ (void)**initialize**                                    Initializes the class before it's used (before it receives its first message).


## Creating and Destroying Instances

+ (id)**alloc**                                    Returns a new, uninitialized instance of the receiving class.
+ (id)**allocWithZone:**(NSZone *)*zone*        Returns a new, uninitialized instance of the receiving class in *zone*.
+ (id)**new**                                    Allocates a new instance of the receiving class, sends it an **init** message,
                                      and returns the initialized object returned by **init**.
- (id)**copy**                                    Invokes **copyWithZone:**.
- (void)**dealloc**                                    Deallocates the memory occupied by the receiver.
- (id)**init**                                    Implemented by subclasses to initialize a new object (the receiver)

immediately after memory for it has been allocated.

- (id)**mutableCopy**                         Invokes **mutableCopyWithZone:**.

## Identifying Classes

+ (Class)**class**                            Returns **self**.   Since this is a class method, it returns the class object.
+ (Class)**superclass**                       Returns the class object for the receiver's superclass.

## Testing Class Functionality

+ (BOOL)**instancesRespondToSelector:**(SEL)*aSelector*
                                              Returns YES if instances of the class are capable of responding to
                                              *aSelector* messages, and NO if they're not.

## Testing Protocol Conformance

+ (BOOL)**conformsToProtocol:**(Protocol *)*aProtocol*
                                              Returns YES if the receiving class conforms to *aProtocol*, and NO if it
                                              doesn't.

## Obtaining Method Information

+ (IMP)**instanceMethodForSelector:**(SEL)*aSelector*
                                              Locates and returns the address of the implementation of the *aSelector*
                                              instance method.
- (IMP)**methodForSelector:**(SEL)*aSelector*    Locates and returns the address of the receiver's implementation of the
                                              *aSelector* method, so that it can be called as a function.
- (NSMethodSignature *)**methodSignatureForSelector:**(SEL)*aSelector*

Returns an object that contains a description of the *aSelector* method, or **nil** if the *aSelector* method can't be found.

## Describing Objects

+ (NSString *)**description**

Subclasses override this method to return a human-readable string representation of the contents of the receiver.   NSObject's implementation simply prints the name of the receiver's class.

## Posing

+ (void)**poseAsClass:**(Class)*aClass*

Causes the receiving class to ªpose asº its superclass.

## Error Handling

- (void)**doesNotRecognizeSelector:**(SEL)*aSelector*

Handles *aSelector* messages that the receiver doesn't recognize.

## Sending Messages Determined at Run Time

+ (void)**cancelPreviousPerformRequestsWithTarget:**(id)*aTarget*
    **selector:**(SEL)*aSelector*
    **object:**(id)*anObject*

Cancels previous perform requests having the same target and argument (as determined by **isEqual:**), and the same selector. This method removes timers only in the current run loop, not all run loops.

- (void)**performSelector:**(SEL)*aSelector*
    **object:**(id)*anObject*
    **afterDelay:**(NSTimeInterval)*delay*

Sends an *aSelector* message to *anObject* after *delay*.   **self** and *anObject* are retained until after the action is executed.

**Forwarding Messages**

- (void)**forwardInvocation:**(NSInvocation *)*anInvocation*
Implemented by subclasses to forward messages to other objects.


**Archiving**

- (id)**awakeAfterUsingCoder:**(NSCoder *)*aDecoder*
Implemented by subclasses to reinitialize the receiver.   The NSObject
implementation of this method simply returns **self**.
- (Class)**classForArchiver**  Identifies the class to be used during archiving.    NSObject's
implementation returns the object returned by **classForCoder:**.
- (Class)**classForCoder**  Identifies the class to be used during serialization.   An NSObject returns its
own class by default.
- (id)**replacementObjectForArchiver:**(NSArchiver *)*anArchiver*
Allows an object to substitute another object for itself during archiving.
NSObject's implementation returns the object returned by
**replacementObjectForCoder:**.
- (id)**replacementObjectForCoder:**(NSCoder *)*anEncoder*
Allows an object to substitute another object for itself during serialization.
NSObject's implementation returns **self**.
+ (void)**setVersion:**(int)*version*  Sets the class version number to *version*.
+ (int)**version**  Returns the version of the class definition.