Sending Messages

Message Type	Syntax
simple message	`[receiver message]`
named arguments	`[receiver message name1:arg1 name2:arg2]`
anonymous arguments	`[receiver message :arg1 :arg2 ]`
typical use	`[receiver message:arg1 name2:arg2]`

A receiver is an object, a class, `self` or `super`.

Class Definition

```
@interface classname (category) <protocols> : superclass
{
@public
@protected
@private
    instance variables
}
method declarations
@end
```

The default visibility of instance variables is `@protected`.

Method Declaration

Method Type	Syntax
instance method	`- (return type) name :(type) arg1 :(type) arg2`
class method	`+(return type) name :(type) arg1 :(type) arg2`

Method Implementation

```
@import "declarations.h"
@implementation classname : superclass
-(return type) methodname :(type) arg1
{
    method implementation
}
@end
```

Protocols

```
@protocol protocolname <protocol list>
    method declarations
@end
```

Exception Handling

```
@try
{
}
@catch (exception)
{
}
@finally
{
}

@throw(exception);  // throw an exception
@throw();  // re-throw current exception
```

Built-in & pre-defined types

Type	Description
`BOOL`	boolean (YES/NO)
`char`	C char
`double`	C double
`float`	C float
`id`	an object
`int`	C int
`long`	C long
`short`	C short
`signed`	C signed
`unsigned`	C unsigned
`void`	no type

Thread Synchronization

```
@synchonized(sync object)
{
     // synchronized code
}
```

Special Names

Name	Description
`nil`	the "null" object
`self`	refers to the current object in a method
`super`	refers to the superclass in a method

Properties (Objective-C 2.0)

Properties automatically create setter and getter methods for a class attribute. They are declared in the method declaration section.

Declaration

```
@property(attributes) type name;
```

Attributes

Attribute	Description
`readonly`	read-only
`readwrite`	read-write (default)
`assign`	simple assignment (default)
`retain`	`retain` method is invoked upon assignment
`copy`	use a copy of the object for assignment
`nonatomic`	not thread-safe
`getter=name`	explicitly name getter
`setter=name`	explicitly name setter

`assign`, `copy` and `retain` are mutually exclusive. The default attributes are `assign` and `readwrite`. In addition to the normal method invocation syntax, dot syntax may be used to access properties: `object.property`

Implementation

```
@synthesize value = variable;
@dynamic value;
```

Fast Enumeration (Objective-C 2.0)

```
for (type variable in expression)
{
}
```

Common Messages

Message	Description
`alloc`	allocate object memory
`init`	initialize object instance
`retain`	increment object's reference count
`release`	decrement object's reference count

Legend

```
code syntax
optional code syntax
```