Objective C NSArray

• Data storage and its retrieval is one the most important in any program.

• NSArray is Objective-C’s general-purpose array type.

**• NSArray** and its subclass **NSMutableArray** manage ordered collections of objects called arrays.

•  NSArray creates static arrays, and NSMutableArray creates dynamic arrays. You can use arrays when you need an ordered collection of objects

• The NSArray class is used for managing an ordered collection of objects.

• An ordered collection of objects is a grouping of objects that is expected to be maintained in the order in which they were stored.

• Typically, an ordered collection of objects is accessed either through enumeration or by index.

• NSArray is used to hold an immutable array of objects.

• Mutability helps to change the array in runtime a pre allocated array.

• If we use NSArray, we only replace the existing array and cannot change the contents of the existing array.

• NSArray is immutable, so you cannot dynamically add or remove items.

• The NSArray class is immutable — once it is created, you cannot modify its contents.

However, because Objective-C provides no mechanism for ensuring the immutability of the objects inside an array, if you access an element of an array, those objects can be modified.

• NSArray as well as all collections in Objective-C is zero based. This means the first element starts at index 0, and the last element has an index of one less than the length of the array.

• If you try to access an element outside of these index boundaries, you will get an exception.

Fig a): NSArray collection classes of the Foundation Framework.

Courtesy: Apple Documentation.

**Creating Arrays:**

Immutable arrays can be defined as literals using the @[] syntax.

**Example:**

NSArray \*germanMakes = @[@"Mercedes-Benz", @"BMW", @"Porsche",@"Opel", @"Volkswagen", @"Audi"];

NSArray \*ukMakes = [NSArray arrayWithObjects:@"Aston Martin",@"Lotus", @"Jaguar", @"Bentley", nil];

NSLog(@"First german make: %@", germanMakes[0]);

NSLog(@"First U.K. make: %@", [ukMakes objectAtIndex:0]);

Enumerating Arrays:

Fast-enumeration is the most efficient way to iterate over an NSArray, and its contents are guaranteed to appear in the correct order. It’s also possible to use the count method with a traditional for-loop to step through each element in the array:

**Example:**

NSArray \*germanMakes = @[@"Mercedes-Benz", @"BMW", @"Porsche", @"Opel", @"Volkswagen", @"Audi"];

**// With fast-enumeration**

for (NSString \*item in germanMakes) {

NSLog(@"%@", item);

}

**// With a traditional for loop**

for (int i=0; i<[germanMakes count]; i++) {

NSLog(@"%d: %@", i, germanMakes[i]);

}

**Important methods of NSArray are as follows**

• alloc/initWithObjects: Used to initialize an array with objects.

• objectAtIndex: Returns the object at specific index.

• count: Returns the number of objects

• lastObject: which returns the last element of the array. To find the index of a specific element, you can also use the method

• isEqual: message to each of the elements of the array, and returns the first element which returns YES.

Comparing Arrays:

Arrays can be compared for equality with the aptly namedisEqualToArray: method, which returns YES when both arrays have the same number of elements and every pair pass an isEqual:comparison. NSArray does not offer the same subset and intersection comparisons as NSSet.

NSArray \*germanMakes = @[@"Mercedes-Benz", @"BMW", @"Porsche", @"Opel", @"Volkswagen", @"Audi"];

NSArray \*sameGermanMakes = [NSArray arrayWithObjects:@"Mercedes-Benz" @"BMW", @"Porsche", @"Opel” @"Volkswagen", @"Audi", nil];

if ([germanMakes isEqualToArray:sameGermanMakes]) {

NSLog(@"Oh good, literal arrays are the same as NSArrays");

}

**Inherits From:**

NSObject

NSArray

NSMutableArray

**Import Statement:**

• @import Foundation;

**Availability:**

• Available in OS X v10.0 and later

**Advantages:**

•  It represents an ordered collection of objects, and it provides a high-level interface for sorting and otherwise manipulating lists of data.

**Disadvantage:**

• The NSArray class is immutable — once it is created, you cannot modify its contents.

**References:**

• https://developer.apple.com/library/mac/documentation/Cocoa/Reference/Foundation/Classes/NSArray\_Class/index.html#//apple\_ref/occ/cl/NSArray

• http://rypress.com/tutorials/objective-c/data-types/nsarray

• http://as.wiley.com/WileyCDA/Section/id-400181.html

• http://www.tutorialspoint.com/objective\_c/objective\_c\_data\_storage.htm