**Data Types**

1. **Id type :**

The id type is one of the data type. It is designed to be generic type which can hold any object. The id type is typically typed (cast) to a specific type after using the introspection methods. The id type is predefined as pointer type there’s no need to add asterisk.

**Syntax:**

Id some object;

2>**NSString:**

The Objective-C class for strings is NSString. Strings are typically created by direct assignment or by calling one of the NSString class methods. The NSString class provides an extensive set of APIs for working with strings, including methods for comparing, searching, and modifying strings. Just like other languages, strings are enclosed and defined by the use of “double quotes”, however in Objective-C an NSString also comes with the prefix of the @ sign.

Syntax:

NSString \*variable name = @”statement”;

E.g.:

NSString string1 \*stirng1 = @”This is NSString”;

Inherits from:

NSObject

NSString

**Comparing String:**

NSString uses the isEqualToString method to compare the two strings and also with this it uses hassuffix and hasprefix methods for partial comparison.

Eg:

NSString \*bus = @”KSRTC red bus”;

If ([bus isEqualToString:@”This is red bus”])

{

NSLog(@”The bus is red”);

}

If ([bus hasprefix:@”KSRTC”])

{

NSLog(@“it is the ksrtc bus of some color”);

}

If ([bus hassuffix:@”bus”])

{

NSLog(@“this is the bus”);

}

**Combining the String:**

NSString is an immutable type so whenever we concatenate the new strings will be created. It uses the two methods to combine the strings they are 1)stringByAppendingString and 2)stringByAppendingFormat.

Eg:

NSString \*fname = @”Harish”;

NSString \*lname = @”Bhardwaj”;

NSString \*name = [ fname stringByAppendingString lname ]

NSLog(@”%@”,name);//HarishBhardwaj

NSString \*name = [ fname stringByAppendingFormat lname ]

NSLog(@”%@”,name);//Harish Bharadwaj

**Searching String:**

The NSString search method returns the NSRange which defines a location and length field. The Location contains the beginning match of the string and length field has toatal number of characters present in the string. if no match was found, location will contain NSNotFound.

**Changing case:**

The NSString change case is used to convert the string from lower case to upper case and upper case to lower case.

Eg:

NSString \*name = @”NeoRays”;

NSLog(@”%@”, [name uppercaseStirng]);//NEORAYS

NSLog(@”%@”, [ name lowercaseString]);//neorays

**2>NSMutabaleString:**

NSObject

NSString

NSMutableString

The NSMutableString class is the mutable version of NSString. The NSMutableString will not create a new string for any changes made for the string. There are so many methods are there to support for the NSMutableString. The mutable string is created with StringewithString Class method. Which turns the NSString object into mutable String

Eg:

NSMutableString \*name = [NSMutableString StringwithString :@”Harish”];

After created a mutable string we can assign the new value for the string using setString method.

**3>NSArray:**

NSObject

NSArray

NSArray is Objective-C’s general-purpose array type. It represents an ordered collection of objects.  NSArray creates static arrays. Immutable arrays can be defined as literals using the @[] syntax.

Eg:

NSArray \*fruits = @ [@”mango”,@”apple”,@”orange”];

NSArray \*colors = @[NSArray arraywithobjects:@”yellow”,@”blue”@”red”];

NSLog(@”The fruits are %@”,fruits[0]);

NSLog(@”The colors are:%@”,[colors objectAtIndex:0]);

* **Comparing arrays:**

The arrays can be compared to check the equality of two different arrays. The array object use isEqualToArray method to compare the two arrays. If the two arrays elements are equal then it returns YES.

Eg:

NSArray \*fruits = @ [@”mango”,@”apple”,@”orange”];

NSArray \*colors = @[NSArray arraywithobjects:@”yellow”,@”blue”@”red”];

If([fruits isEqualToArray:colors])

{

NSLog(@”the given arrays are equal “);

}

**4>NSMutableArray:**

NSObject

NSArray

NSMutableArray

The easiest way to create mutable arrays is arrayWithObjects:method we can create the mutable empty array using ArrayWithCapacity: class method.

Eg:

NSMutableArray \*city = [NSMutableArray arrayWithObjects:@”hassn”,

@”banglore”,@”manglore”]);

* addObject and removeLastObject: adds the elements to the end of the array.
* insertObject:atIndex and removeObjectAtIndex: if we know the index of the object we use remove fromObjectatindex .if we don’t know the particular index means we will use the removeObject:method
* replaceObjectAtIndex:withObject: used to replace the content of the objects.

**5>NSDictionary:**

The NSDictionary represents the unordered collection of Objects. The NSDictionary Contains the key and a value pair. The NSDictionary is immutable class. The NSDictionary can be declared using the literals @{} syntax.

Eg: