IOS - OBJECTIVE C

http://www.tutorialspoint.com/ios/ios_objective_c.htm

The language used in iOS development is objective C. It is an object-oriented language and hence, it would be easy for those who have some background in object-oriented programming languages.

Interface and Implementation

In Objective C, the file where the declaration of class is done is called the **interface file** and the file where the class is defined is called the **implementation file**.

A simple interface file **MyClass.h** would look like the following –

```
@interface MyClass:NSObject{
// class variable declared here
}
// class properties declared here
// class methods and instance methods declared here
@end
```

The implementation file **MyClass.m** would be as follows –

```
@implementation MyClass
// class methods defined here
@end
```

Object Creation

Object creation is done as follows –

```
MyClass  *objectName = [[MyClass alloc]init] ;
```

Methods

Method is declared in Objective C as follows –

```
-(returnType)methodName:(typeName) variable1 :(typeName)variable2;
```

An example is shown below.

```
-(void)calculateAreaForRectangleWithLength:(CGfloat)length
andBreadth:(CGfloat)breadth;
```

You might be wondering what the **andBreadth** string is for; actually it's an optional string, which helps us read and understand the method easily, especially at the time of calling. To call this method in the same class, we use the following statement –

```
[self calculateAreaForRectangleWithLength:30 andBreadth:20];
```

As said above, the use of andBreadth helps us understand that breadth is 20. Self is used to specify that it's a class method.

Class Methods

Class methods can be accessed directly without creating objects for the class. They don't have any variables and objects associated with it. An example is shown below.

```
+(void)simpleClassMethod;
```

It can be accessed by using the class name $let'sassumetheclassnameasMyClass$ as follows –

```
[MyClass simpleClassMethod];
```

Instance Methods

Instance methods can be accessed only after creating an object for the class. Memory is allocated to the instance variables. An example instance method is shown below.

```
-(void)simpleInstanceMethod;
```

It can be accessed after creating an object for the class as follows −

```
MyClass  *objectName = [[MyClass alloc]init] ;
[objectName simpleInstanceMethod];
```

Important Data Types in Objective C

S.N.	Data Type
1	**NSString** It is used for representing a string.
2	**CGfloat** It is used for representing a floating point value *normalfloatisalsoallowedbutit′sbettertouseCGfloat*.
3	**NSInteger** It is used for representing integer.
4	**BOOL** It is used for representing Boolean *YESorNOareBOOLtypesallowed*.

Printing Logs

NSLog - used for printing a statement. It will be printed in the device logs and debug console in release and debug modes respectively. For example,

```
NSlog(@"");
```

Control Structures

Most of the control structures are same as in C and C++, except for a few additions like for in statement.

Properties

For an external class to access the class, variable properties are used. For example,

```
@property(nonatomic , strong) NSString *myString;
```

Accessing Properties

You can use dot operator to access properties. To access the above property, we will do the following.

```
self.myString = @"Test";
```

You can also use the set method as follows −

```
[self setMyString:@"Test"];
```

Categories

Categories are used to add methods to the existing classes. By this way, we can add method to classes for which we don't have even implementation files where the actual class is defined. A sample category for our class is as follows −

```
@interface MyClass(customAdditions)
- (void)sampleCategoryMethod;
@end

@implementation MyClass(categoryAdditions)

-(void)sampleCategoryMethod{
   NSLog(@"Just a test category");
}
```

Arrays

NSMutableArray and NSArray are the array classes used in objective C. As the name suggests, the former is mutable and the latter is immutable. An example is shown below.

```
NSMutableArray *aMutableArray = [[NSMutableArray alloc]init];
[anArray addObject:@"firstobject"];
NSArray *aImmutableArray = [[NSArray alloc]
initWithObjects:@"firstObject",nil];
```

Dictionary

NSMutableDictionary and NSDictionary are the dictionary classes used in objective C. As the name suggests, the former is mutable and the latter is immutable. An example is shown below.

```
NSMutableDictionary*aMutableDictionary = [[NSMutableArray alloc]init];
[aMutableDictionary setObject:@"firstobject" forKey:@"aKey"];
NSDictionary*aImmutableDictionary= [[NSDictionary alloc]initWithObjects:[NSArray
arrayWithObjects:
@"firstObject",nil] forKeys:[ NSArray arrayWithObjects:@"aKey"]];
```