OBJECTIVE-C CATEGORIES

http://www.tutorialspoint.com/objective_c/objective_c_categories.htm

Sometimes, you may find that you wish to extend an existing class by adding behavior that is useful only in certain situations. In order add such extension to existing classes, Objective-C provides **categories** and **extensions**.

If you need to add a method to an existing class, perhaps, to add functionality to make it easier to do something in your own application, the easiest way is to use a category.

The syntax to declare a category uses the @interface keyword, just like a standard Objective-C class description, but does not indicate any inheritance from a subclass. Instead, it specifies the name of the category in parentheses, like this:

```
@interface ClassName (CategoryName)

@end
```

Characteristics of category

- A category can be declared for any class, even if you don't have the original implementation source code.
- Any methods that you declare in a category will be available to all instances of the original class, as well as any subclasses of the original class.
- At runtime, there's no difference between a method added by a category and one that is implemented by the original class.

Now, let's look at a sample category implementation. Let's add a category to the Cocoa class NSString. This category will make it possible for us to add a new method getCopyRightString which helps us in returning the copyright string. It is shown below.

```
#import <Foundation/Foundation.h>

@interface NSString(MyAdditions)

+(NSString *)getCopyRightString;

@end

@implementation NSString(MyAdditions)

+(NSString *)getCopyRightString{
    return @"Copyright TutorialsPoint.com 2013";
}

@end


int main(int argc, const char * argv[])
{

   NSAutoreleasePool * pool = [[NSAutoreleasePool alloc] init];
   NSString *copyrightString = [NSString getCopyRightString];
   NSLog(@"Accessing Category: %@",copyrightString);
   [pool drain];
    return 0;
}
```

Now when we compile and run the program, we will get the following result.

```
2013-09-22 21:19:12.125 Categories[340:303] Accessing Category: Copyright
TutorialsPoint.com 2013
```

Even though any methods added by a category are available to all instances of the class and its subclasses, you'll need to import the category header file in any source code file where you wish to use the additional methods, otherwise you'll run into compiler warnings and errors.

In our example, since we just have a single class, we have not included any header files, in such a case we should include the header files as said above.