1. @ sign immediately precedes a constant **NSString** object, if no @ is specified, you are writing a constanct C-style string.
2. How to call methods defined in objective c.

myCar = [Car new];

[myCar getName];

[myCar setName:@”hello world”];

1. How to define one class

The @interface section describes the class and its methods, and the @implementation section describes the data (the instance variables that objects from the class will store) and contains the actual code that implements the methods declared in the interface section.

**@interface Fraction: NSObject //Parent class NSObject**

**-(void) print;**

**-(void) setNumerator: (int) n;**

**-(void) setDenominator: (int) m;**

**@end**

**// here – means it’s an instance method, + means it’s a class method**

**@implementation Fraction**

**{**

**int numerator;**

**int denominator;**

**}**

**-(void) print**

**{**

**NSLog(@”%i / %i ”, numerator, denominator);**

**}**

**-(void) setNumerator: (int) n**

**{**

**numerator = n;**

**}**

**-(void) setDenominator: (int) m**

**{**

**denominator = m;**

**}**

**@end**

When a method takes an argument, you also append a colon to the method name when referring to the method. Therefore, setNumerator: and setDenominator: is the correct way to identify these two methods, each of which takes a single argument.

1. dfdf
2. dfdf