“Constructor” Methods

* There are no constructor methods in Objective-C.
* In Objective C, an object is **initialized** by calling the init method immediately after it’s allocated.
* This is why instantiation is always a two-step process: allocate, and then initialize.
* There is also a class-level initialization method in Objective-C.
* In Objective-C init is the default initialization method.
* In Objective-C you can also define your own versions of Constructors to accept configuration parameters.
* There’s nothing special about custom initialization methods—they’re just normal instance methods, except the method name should always begin with init.

**Example:**

// Car.h

- (id)initWithModel:(NSString \*)aModel;

* To implement this method, you should follow the canonical initialization pattern shown in initWithModel: below.

**Example:**

// Car.m

- (id)initWithModel:(NSString \*)aModel {

self = [super init];

if (self) {

// Any custom setup work goes here

\_model = [aModel copy];

\_odometer = 0;

}

return self;

}

* The super keyword refers to the parent class, and again, the self keyword refers to the instance calling the method.
* Initialization methods should always return a reference to the object itself, and if it cannot be initialized, it should return nil.
* This is why we need to check if self exists before trying to use it