Over the past 15 years, the programming world focused on the development paradigm of object oriented programming (OOP). Most modern development environments and languages implement OOP. Put simply, OOP forms the basis of everything you develop today.

You may be asking yourself why we waited until Chapter 5 to present OOP using Objective-C if it is the primary development style of today. The simple answer is that it is not an easy concept for new developers. We will spend this chapter going into detail about the different aspects of OOP and how this will affect your development.

Implementing OOP into your applications correctly will take some front-end planning but you will save yourself a lot of time throughout the life of your projects. OOP has changed the way development is done. In this chapter, we will look at what OOP is. OOP was initially discussed in the first chapter of this book, but we will go into more detail here. We will revisit what objects are and how they relate to physical objects we find in our world. We will also look into what classes are and how they relate to objects. We will also discuss steps you will need to take when planning your classes and some visual tools you can use to accomplish this. When you have read this chapter and have worked through the exercises, you will have a better understanding of what OOP is and why it is necessary for you as a developer.

At first, objects and object-oriented programming may seem difficult to understand, but the hope is that as we progress through this chapter, it will begin to make sense.
The Object

As was discussed in Chapter 1, OOP is based on objects. Some of our discussion about objects will be a review, but we will also go into more depth. An object is anything that can be acted upon. In order to better explain what a programming object is, we will first look at some items in the physical world around us. A physical object can be anything around you that you can touch or feel. Take, for example, a television. Some characteristics of a television include type (plasma, LCD, or CRT), size (40 inches), brand (Sony, Vizio), weight, and cost. Televisions also have functions. They can be turned on or off. You can change the channel, adjust the volume, and change brightness.

Some of these characteristics and functions are unique to televisions and some are not. For example, a couch in your house would probably not have the same characteristics as a television. You would want different information about a couch, such as material type, seating capability, and color. A couch might have only a few functions, such as converting to a bed.

Now let’s talk specifically about objects as they relate to programming. An object is a specific item. It can describe something physical like a book, or it could be something such as a window for your application. Objects have properties and methods. Properties describe certain things about an object such as location, color, or name. Conversely, methods describe actions the object can perform such as close or recalculate. In our example, a TV object would have type, size, and brand properties, while a Couch object would have properties such as color, material, and comfort level. In programming terms, a property is a variable that is part of an object. For example, a TV would use a string variable to store the brand and an integer to store the height.

Objects also have commands the programmer can use to control them. The commands are called methods. Methods are the way that other objects interact with a certain object. For example, with the television, a method would be any of the buttons on the remote control. Each of those buttons represents a way you can interact with your television. Methods can and often are used to change the values of properties, but methods do not store any values themselves.

As we described in Chapter 1, objects have a state, which is basically a snapshot of an object at any given point in time. A state would be the values of all of the properties at a specific time.

In previous chapters, we have used the example of the bookstore. A bookstore contains many different objects. It contains book objects that have properties such as title, author, page count, and publisher. It also contains magazines with properties such as title, issue, genre, and publisher. A bookstore also has some non-tangible objects such as a sale. A sale object would contain information about the books purchased, the customer, the amount paid, and the payment type. A sale object might also have some methods that calculate tax, print the receipt, or void the sale. A sale object does not represent a tangible object, but it is still an object and is necessary for creating an effective bookstore.