In this chapter, you will have an overview of the Spring framework. Spring’s architecture is divided into multiple modules in a hierarchical fashion. You will learn about the major functions of each module and see a summary of the highlights of Spring 2.0 and 2.5. Spring is not just an application framework but also a platform that hosts several related projects, which are called the Spring Portfolio projects. This chapter will give you an outline of their features.

Before you start with Spring, you have to install it on your local development machine. The Spring framework’s installation is very simple. However, to get the most out of Spring, you have to understand the structure of its installation directory and what’s contained in each subdirectory.

The Spring development team has created an Eclipse plug-in called Spring IDE to make developing Spring applications easier. You will learn about how to install this IDE and use its bean-supporting features.

Upon finishing this chapter, you will have a solid understanding of Spring’s overall architecture and major features. You will also be able to install the Spring framework and Spring IDE on your machine.

2-1. Introducing the Spring Framework

The Spring framework (http://www.springframework.org/) is a comprehensive Java/Java EE application framework hosted by SpringSource (http://www.springsource.com/), which was formerly known as Interface21. Spring addresses many aspects of Java/Java EE application development, and it can help you to build high-quality, high-performance applications more quickly.

The heart of the Spring framework is a lightweight IoC container that is able to add enterprise services to simple Java objects declaratively. Spring makes extensive use of an excellent programming methodology—AOP (aspect-oriented programming)—to provide these services to its components. Within the Spring IoC container’s scope, components are also called beans.

The Spring framework itself incorporates many design patterns, including the GoF (Gang of Four) object-oriented patterns and Sun’s core Java EE patterns. By using the Spring framework, you will be lead to use industry best practices to design and implement your applications.

Spring is not designed to compete with existing technologies in particular areas. On the contrary, it integrates with many leading technologies to make them easier to use. That makes Spring an appropriate solution in many usage scenarios.
Introducing Spring’s Modules

The architecture of the Spring framework is divided into modules, as shown in Figure 2-1. Spring’s module assembling is so flexible that your applications can build on top of their different subsets in different usage scenarios.

In Figure 2-1, the modules are organized in hierarchical fashion, with the upper modules depending on the lower ones. As you can see, the Core module lies at the very bottom since it’s the foundation of the Spring framework.

**Core**: This module provides core functions of the Spring framework. It provides a basic Spring IoC container implementation called `BeanFactory`. The basic IoC container features will be introduced in Chapter 3.

**Context**: This module builds on top of the Core module. It extends the Core module’s functions and provides an advanced Spring IoC container implementation called `ApplicationContext`, which adds features such as internationalization (I18N) support, event-based communication, and resource loading. These advanced IoC container features will be covered in Chapter 4.

**AOP**: This module establishes an aspect-oriented programming framework, which is referred to as Spring AOP. AOP is another of Spring’s fundamental concepts besides IoC. Chapters 5 and 6 will cover both the classic and new Spring AOP approaches, and also integrating AspectJ with Spring.