**Part 2 – Spring Framework Reference Documentation**Chapter 1. The IoC Container [[https://docs.spring.io/spring/docs/5.2.x/spring-framework-reference/core.html (Links to an external site.)](https://docs.spring.io/spring/docs/5.2.x/spring-framework-reference/core.html)]  
This chapter covers the Spring Framework implementation of the Inversion of Control (IoC) principle. IoC is also known as dependency injection (DI). It is a process whereby objects define their dependencies (that is, the other objects they work with) only through constructor arguments, arguments to a factory method, or properties that are set on the object instance after it is constructed or returned from a factory method. The container then injects those dependencies when it creates the bean. This process is fundamentally the inverse (hence the name, Inversion of Control) of the bean itself controlling the instantiation or location of its dependencies by using direct construction of classes or a mechanism such as the Service Locator pattern.

**Part 3 – Reading Assignment**Chapter 1. Springing into action    [https://www.safaribooksonline.com/library/view/spring-in-action/9781617291203/kindle\_split\_009.html (Links to an external site.)](https://www.safaribooksonline.com/library/view/spring-in-action/9781617291203/kindle_split_009.html)  
Chapter 2. Wiring beans  
    [https://www.safaribooksonline.com/library/view/spring-in-action/9781617291203/kindle\_split\_010.html (Links to an external site.)](https://www.safaribooksonline.com/library/view/spring-in-action/9781617291203/kindle_split_010.html)  
If the links are not accessible, you may refer to the PDF version:  
[https://github.com/andriyanov-roman/planet/blob/master/Craig%20Walls%20-%20Spring%20in%20Action%204th%20Edition%20-%202014.pdf (Links to an external site.)](https://github.com/andriyanov-roman/planet/blob/master/Craig%20Walls%20-%20Spring%20in%20Action%204th%20Edition%20-%202014.pdf)

**Part 4 - Programming Assignment**Redo HW3-Part 6 using Spring MVC – Map all the controllers using ControllerClassNameHandlerMapping.

**Part 5 - Programming Assignment**Redo HW3-Part 7 using Spring MVC – Map all the controllers using BeanNameUrlHandlerMapping.

**Part 6 - Programming Assignment**Redo HW3-Part 8 using Spring MVC – Use Annotated Controllers

**Part 7 - Programming Assignment**Create a utility class that has a get method returning a string (it could be any random message). Use Spring’s IoC container to inject this into the handler method, and simply retrieve and pass this message to the view page. Use context to create an instance of the utility class. Do not use new operator. Set the Bean scope to singleton, and see if you will get the same instance. You could just print the Object ID to the console to see if the same object is used with other requests.

**Part 8 - Programming Assignment**Redo Part 7 of this assignment, and use annotations (@Autowired). Set the Bean scope to singleton, and see if you will get the same instance. You could just print the Object ID to the console to see if the same object is used with other requests.

**Part 9 - Programming Assignment**Redo Part 8 of this assignment, and use annotations. Set the Bean scope to request, and see if you will get the same instance. You could just print the Object ID to the console to see if the same object is used with other requests.