Now that you've completed the section, let's see if you can answer these questions. The answer sheet is available as a PDF in the "Support Files" folder.

Section 9: Peeking under the Hood

1) You have two beans of the same type and you want to have one of them active based on a profile **test**, how can you solve this?
   a. Add **@Profile("test")** to the designated bean.
   b. Add **@Profile("test")** to the designated bean and **@Profile("!test")** to the other.
   c. Add **@Profile("test")** to the designated bean and **@Profile("-test")** to the other.
   d. Add **@Profile("test")** to the designated bean and **@Primary** to the other.
   e. Add **@Profile("test")** and **@Primary** to the designated bean.

2) Given that you have two beans of the same type with **@ConditionalOnProperty** and **matchIfMissing=true** and the given property is not set. What is the result?
   a. The property is missing. So, none of the beans is in the context.
   b. The first bean that is found is added to the context.
   c. Both beans are added to the context.
   d. Spring refuses to start because it has two beans of the same type in the context.

3) You are creating the Spring expression language based **@Value** annotation to a field and you want to get the value of the property **test.prop** and if this is not there you want the value of the property **fallback.prop** or the default value of **defaultvalue**. Which among the following is the correct expression?
   a. ${'test.prop':'fallback.prop':'defaultvalue'}
   b. ${'test.prop':${'fallback.prop':'defaultvalue'}}
   c. ${test.prop:${fallback.prop}:defaultvalue}
   d. ${test.prop:${fallback.prop:defaultvalue}}
   e. ${test.prop:${fallback.prop:'defaultvalue'}}

4) You have added a **@Value** annotation which calls a bean that fetches something from the database. When is Spring going to evaluate this and fetch the data?
   a. When the spring context is built
   b. When the bean is instantiated
   c. When the property is read
   d. You can't access the database from a SpEL expression

5) You have a bean that has the request scope and you want to autowire it in a regular service that has singleton scope. Which of the statements are true?
   a. The autowired field is null and only has a value when the service is processing a request.
   b. Autowiring will fail, because there is no request.
   c. Spring creates a proxy which returns the correct instance based on the request.
   d. You need to specify the proxy mode on the bean definition.
   e. You need to specify the proxy mode on the field definition.
   f. You don't need to specify the proxy mode anywhere. Spring does this for you.