**Question: 1** What is true about Spring security configuration and the security namespace?  
  
**Your Answer:** None  
  
**Correct Answer:** The patterns declared into the intercept-url tag are analyzed from up to bottom. Winning is the first that matches.  
  
**Description:** If more than one intercept-url matches, the top one is used

**Question: 2** Where do you cannot declare Spring MVC controller? Select correct answer.  
  
**Your Answer:** None  
  
**Correct Answer:** Into the web.xml file of the web application  
  
**Description:** In the web.xml, you may declarer and a ContextLoaderListener and a DispatcherServlet that   
are in charge to load XML Spring configuration files. But you cannot declare controllers   
directly in these file.

**Question: 3** How could you secure MVC controller with Spring Security? Select a unique answer.  
  
**Your Answer:** None  
  
**Correct Answer:** All of the above  
  
**Description:** None

**Question: 4** What is the purpose of the @ManageResource annotation?  
  
**Your Answer:** None  
  
**Correct Answer:** Identify a Spring bean as a JMX MBean  
  
**Description:** @ManageResources identify a Spring bean as a JMX MBean

**Question: 5** What are the possible mechanisms provided by Spring Security to store user details?  
  
**Your Answer:** None  
  
**Correct Answer:** All of the above  
  
**Description:** None

**Question: 6** What is the method that is not provided by the JmsTemplate Spring class?  
  
**Your Answer:** None  
  
**Correct Answer:** onMessage  
  
**Description:** The onMessage method does not exist.

**Question: 7** What could not return a Spring MVC controller? Select a single answer.  
  
**Your Answer:** None  
  
**Correct Answer:** An absolute path to the view  
  
**Description:** Spring does not allow to return an absolute path to the view

**Question: 8** What is exact about the HttpInvokerServiceExporter?  
  
**Your Answer:** None  
  
**Correct Answer:** Has to run into a HTPP server as Jetty  
  
**Description:** HttpInvokerServiceExporter requires a HTTP web server to process incoming http request.   
Tomcat or Jetty is possible candidates. Spring also supports the Oracle/Sun´s JRE 1.6 HTTP   
server.

**Question: 9** What is the easiest method to write a unit test?  
  
**Your Answer:** None  
  
**Correct Answer:** @RequestMapping("/displayAccount")   
String displayAccount(@RequestParam("accountId") int id, Model model)  
  
**Description:** This method is not dependent of the servlet API. Id of the account to display may be directly   
passed through the call stack. Thus test methods are simplified.

**Question: 10** What do you have to do even if you are using the RMI Spring Remoting support?   
  
**Your Answer:** None  
  
**Correct Answer:** Implements the Serializable interface  
  
**Description:** Object that are transferred via RMI are serializabled/unserializabled. So they have to   
implement the Serializable interface.