                                       Spring MCQS

1. Which method is used to gracefully shutdown all the bean processes after closing the spring container?

* [**A.**](http://javascript:%20void(0)/) shutdownHook
* [**B.**](http://javascript:%20void(0)/) destory method
* [**C.**](http://javascript:%20void(0)/) none of the mentioned
* [**D.**](http://javascript:%20void(0)/) all of the mentioned

2. Which one is the default scope of the beans?

* [**A.**](http://javascript:%20void(0)/) Prototype
* [**B.**](http://javascript:%20void(0)/) Session
* [**C.**](http://javascript:%20void(0)/) Request
* [**D.**](http://javascript:%20void(0)/) Singleton

3. Which attribute is used to specify classname of the bean?

* [**A.**](http://javascript:%20void(0)/) name
* [**B.**](http://javascript:%20void(0)/) id
* [**C.**](http://javascript:%20void(0)/) class
* [**D.**](http://javascript:%20void(0)/) constructor-arg

4. Declaring bean form object properties can be done using:-

* [**A.**](http://javascript:%20void(0)/) PropertyPathFactoryBean
* [**B.**](http://javascript:%20void(0)/) util:constant
* [**C.**](http://javascript:%20void(0)/) None of the mentioned
* [**D.**](http://javascript:%20void(0)/) All of the mentioned

5. What are the ways to create custom Property Editors?

* [**A.**](http://javascript:%20void(0)/) implement PropertyEditor interface
* [**B.**](http://javascript:%20void(0)/) extend PropertyEditorSupport Class
* [**C.**](http://javascript:%20void(0)/) none of the mentioned
* [**D.**](http://javascript:%20void(0)/) all of the mentioned

6. Which package is used for periodic work

* [**A.**](http://javascript:%20void(0)/) java.lang.Thread
* [**B.**](http://javascript:%20void(0)/) java.util.TimerTask
* [**C.**](http://javascript:%20void(0)/) java.util.Timer
* [**D.**](http://javascript:%20void(0)/) java.util.concurrent

7. Method used to process bean before initialization callback

* [**A.**](http://javascript:%20void(0)/) scope
* [**B.**](http://javascript:%20void(0)/) postProcessAfterInitialization()
* [**C.**](http://javascript:%20void(0)/) postProcessBeforeInitialization()
* [**D.**](http://javascript:%20void(0)/) it’s own constructor

8. Which attribute is used to set the scope of the bean?

* [**A.**](http://javascript:%20void(0)/) setScope
* [**B.**](http://javascript:%20void(0)/) scope
* [**C.**](http://javascript:%20void(0)/) getScope
* [**D.**](http://javascript:%20void(0)/) none of the mentioned

9. Which attribute is used to specify classname of the bean?

* [**A.**](http://javascript:%20void(0)/) name
* [**B.**](http://javascript:%20void(0)/) id
* [**C.**](http://javascript:%20void(0)/) class
* [**D.**](http://javascript:%20void(0)/) constructor-args

10. Which tag is also allowed by static field?

* [**A.**](http://javascript:%20void(0)/) util:constant
* [**B.**](http://javascript:%20void(0)/) list
* [**C.**](http://javascript:%20void(0)/) set
* [**D.**](http://javascript:%20void(0)/) constructor-args

11 Validator field is not assigned to any bean, but rather a factory class of the type.

* [**A.**](http://javascript:%20void(0)/) javax.validation.Validator
* [**B.**](http://javascript:%20void(0)/) javax.validation.ValidatorFactory
* [**C.**](http://javascript:%20void(0)/) javax.validation.ValidatorFactor
* [**D.**](http://javascript:%20void(0)/) none of the mentioned

12. Handler method’s annotation.

* [**A.**](http://javascript:%20void(0)/) @Before
* [**B.**](http://javascript:%20void(0)/) @Controller
* [**C.**](http://javascript:%20void(0)/) @After
* [**D.**](http://javascript:%20void(0)/) @RequestMapping

13. Alternative way to resolve locales.

* [**A.**](http://javascript:%20void(0)/) AcceptHeaderLocale
* [**B.**](http://javascript:%20void(0)/) AcceptHeader
* [**C.**](http://javascript:%20void(0)/) AcceptHeaderLocaleResolver
* [**D.**](http://javascript:%20void(0)/) SessionLocaleResolver

14. View beans from a resource bundle is loaded by.

* [**A.**](http://javascript:%20void(0)/) ResourceBundleViewResolver
* [**B.**](http://javascript:%20void(0)/) ResourceBundleView
* [**C.**](http://javascript:%20void(0)/) ResourceViewResolver
* [**D.**](http://javascript:%20void(0)/) None of the mentioned

15. To load root application context at the start up.

* [**A.**](http://javascript:%20void(0)/) ContextListener
* [**B.**](http://javascript:%20void(0)/) ContextLoader
* [**C.**](http://javascript:%20void(0)/) ContextLoaderListener
* [**D.**](http://javascript:%20void(0)/) ContextEventListener

16. Connection of database with Apache derby server requires.

* [**A.**](http://javascript:%20void(0)/) Derby client.jar
* [**B.**](http://javascript:%20void(0)/) Spring JDBC support
* [**C.**](http://javascript:%20void(0)/) None of the mentioned
* [**D.**](http://javascript:%20void(0)/) All of the mentioned

17. DAO methods require access to the session factory, which can be injected:-

* [**A.**](http://javascript:%20void(0)/) a setter method
* [**B.**](http://javascript:%20void(0)/) constructor argument
* [**C.**](http://javascript:%20void(0)/) none of the mentioned
* [**D.**](http://javascript:%20void(0)/) all of the mentioned

18. Interface whose instances can be obtained from a SessionFactory instance.

* [**A.**](http://javascript:%20void(0)/) Session
* [**B.**](http://javascript:%20void(0)/) Session Factory
* [**C.**](http://javascript:%20void(0)/) All of the mentioned
* [**D.**](http://javascript:%20void(0)/) None of the mentioned

19. To access a database running on the Derby server, you have to add:-

* [**A.**](http://javascript:%20void(0)/) Derby client library
* [**B.**](http://javascript:%20void(0)/) Tomcat client library
* [**C.**](http://javascript:%20void(0)/) All of the mentioned
* [**D.**](http://javascript:%20void(0)/) None of the mentioned

20. SimpleJdbcTemplate offers a convenient batch update method in the form of:-

* [**A.**](http://javascript:%20void(0)/) Vector
* [**B.**](http://javascript:%20void(0)/) Set
* [**C.**](http://javascript:%20void(0)/) Map
* [**D.**](http://javascript:%20void(0)/) List