1. What is spring?
2. Spring is an open source development framework for enterprise Java.
3. Spring is a proprietary framework.
4. Spring is a development framework for .Net applications.
5. Spring is a development framework for PHP based applications.
6. Which of the following is correct assertion about spring?
7. Spring enables developers to develop enterprise-class applications using POJOs.
8. Spring is organized in a modular fashion.
9. Testing an application written with spring is simple because environment-dependent code is moved into this framework.
10. All of above.
11. What is Dependency Injection?
12. It is a design pattern which implements Inversion of Control for software applications.
13. It is one of the spring module.
14. It is a technique to get dependencies of any project.
15. It is used to promote tight coupling in code.
16. Which of the following is correct about dependency injection?
17. It helps in decoupling application objects from each other.
18. It helps in deciding the dependencies of objects.
19. It stores objects states in database.
20. It stores object states in file system.
21. What AOP stands for?
22. Aspect Oriented Programming
23. Any Object Programming
24. Asset Oriented Programming
25. Asset Oriented Protocol
26. What is true about cross-cutting concerns?
27. The functions that span multiple points of an application are called cross cutting concerns.
28. Cross-cutting concerns are conceptually separate from the application's business logic.
29. Logging is one of the examples of cross cutting concerns.
30. All of the above.
31. Which are the modules of core container?
32. Beans, Core, Context, SpEL
33. Core, Context, ORM, Web
34. Core, Context, Aspects, Test
35. Bean, Core, Context, Test
36. Which are the modules of Data Access/ integration layer?
37. JDBC, ORM, OXM, JMS, Transactions
38. JDBC, ORM, OXM, JMS
39. JDBC, ORM, Web, Beans
40. JDBC, ORM, OXM, JMS
41. Which are the modules of Web layer?
42. WebSocket, Servlet, Web, Portlet
43. WebSocket, Servlet, Web-MVC, Web
44. HTML, JSP, WEB, Portlet
45. HTML, Servlet, WEB, Portlet
46. Which of the statement is not correct?
47. Core and beans modules provide the fundamental parts of the framework, including

Dependency Injection feature.

1. The SpEL module provides a powerful Expression Language for querying and manipulating an

object graph atruntime.

1. Aspects module provides integration with AspectJ.
2. None of the above.
3. Which of the statement is correct?
4. The JDBC module provides a JDBC-abstraction layer thatremoves the need to do tedious JDBC related coding.
5. The ORM module provides integration layers for popular object-relational mapping APIs, including JPA, JDO, Hibernate, and iBatis.
6. The Java Messaging Service JMS module contains features for producing and consuming messages.
7. All of the above.
8. Which of the statement is correct?
9. The AOP module provides aspect-oriented programming implementation allowing you to define method-interceptors and pointcuts to cleanly decouple code that implements functionality that should be separated.
10. The Aspects module provides integration with AspectJ - Which is again a powerful and mature aspect oriented programming (AOP) framework.
11. The Instrumentation module provides class instrumentation support and class loader implementations to be used in certain application servers.
12. All of the above.
13. What types of Dependency injection does spring supports?
14. Constructor based, Setter based
15. Constructor based, Setter based, Getter Based
16. Setter based, Getter based, Properties based
17. Constructor based, Setter based, Properties based
18. Which are the IoC containers in Spring?
19. BeanFactory, ApplicationContext
20. BeanFactory, ApplicationContext, IocContextFactory
21. BeanFactory, BeanContext, IocContextFactory
22. BeanFactory, ApplicationContext, BeanContext
23. Which is the correct implementation class of BeanFactory?
24. XmlBeanFactory
25. ClassPathBeanFactory
26. FileSystemBeanFactory
27. AdvancedBeanFactory
28. Which are the correct implementation classes of ApplicationContext?
29. FileSystemXmlApplicationContext, ClassPathXmlApplicationContext, WebXmlApplicationContext
30. FileSystemApplicationContext, ClassPathApplicationContext, WebApplicationContext
31. AdvancedApplicationContext, FileApplicationContext
32. FileSystemApplicationContext, ClassPathApplicationContext
33. Which of the following stands true for spring beans?
34. Spring beans are managed by the Spring IoC container.
35. Spring beans are instantiated, assembled, and otherwise managed by a Spring IoC container.
36. Spring beans are simple POJOs.
37. All of the above.
38. Which is the way to provide configuration metadata to spring?
39. XML Based configuration file.
40. Annotation based configuration.
41. Java based configuration.
42. All of the above.
43. What is bean scope?
44. Bean scope forces Spring to produce a new bean instance as per the scope defined.
45. Bean scope defines the accessibility of bean in a java class.
46. Bean scope defines the accessibility of bean in a java package.
47. Bean scope defines the accessibility of bean in a web application.
48. What is singleton scope?
49. This scopes the bean definition to a single instance per Spring IoC container.
50. This scopes the bean definition to a single instance per HTTP Request.
51. This scopes the bean definition to a single instance per HTTP Session.
52. This scopes the bean definition to a single instance per HTTP Application/ Global session.
53. What is prototype scope?
54. This scopes a single bean definition to have any number of object instances.
55. This scopes the bean definition to a single instance per HTTP Request.
56. This scopes the bean definition to a single instance per HTTP Session.
57. This scopes the bean definition to a single instance per HTTP Application/ Global session.
58. What is request scope?
59. This scopes a bean definition to an HTTP request.
60. This scopes the bean definition to Spring IoC container.
61. This scopes the bean definition to HTTP Session.
62. This scopes the bean definition HTTP Application/ Global session.
63. What is session scope?
64. This scopes a bean definition to an HTTP session.
65. This scopes the bean definition to Spring IoC container.
66. This scopes the bean definition to HTTP request.
67. This scopes the bean definition to HTTP Application/ Global session.
68. What is global-session scope?
69. This scopes a bean definition to an HTTP Application/ Global session.
70. This scopes the bean definition to Spring IoC container.
71. This scopes the bean definition to HTTP request.
72. This scopes the bean definition to HTTP Session.
73. What is default scope of bean in Spring framework?
74. singleton
75. prototype
76. request
77. session
78. How can you inject Java Collection in Spring?
79. Using list, set, map or props tag.
80. Using lit, set, map or collection tag.
81. Using list, set, props or collection tag.
82. Using list, collection, map or props tag.
83. What is true about <list> collection configuration elements?
84. This helps in wiring a list of values, allowing duplicates.
85. This helps in wiring a list of values but without any duplicates.
86. This can be used to inject a collection of name-value pairs where name and value can be of any type.
87. This can be used to inject a collection of name-value pairs where the name and value are both Strings.
88. What is true about <set> collection configuration elements?
89. This helps in wiring a list of values, allowing duplicates.
90. This helps in wiring a list of values but without any duplicates.
91. This can be used to inject a collection of name-value pairs where name and value can be of any type.
92. This can be used to inject a collection of name-value pairs where the name and value are both Strings.
93. What is true about <map> collection configuration elements?
94. This helps in wiring a list of values, allowing duplicates.
95. This helps in wiring a list of values but without any duplicates.
96. This can be used to inject a collection of name-value pairs where name and value can be of any type.
97. This tag is not supported.
98. What is true about <props> collection configuration elements?
99. This helps in wiring a list of values, allowing duplicates.
100. This helps in wiring a list of values but without any duplicates.
101. This can be used to inject a collection of name-value pairs where name and value can be of any type.
102. This can be used to inject a collection of name-value pairs where the name and value are both Strings.
103. What is bean autowiring?
104. Autowiring lets Spring resolve collaborators (other beans) for your bean by inspecting the contents of the BeanFactory without using <constructor-arg> and <property> elements.
105. Autowiring injects values in spring beans.
106. Autowiring injects one bean into another.
107. Autowiring helps in wiring a list of values, allowing duplicates.
108. Which are the different modes of autowiring?
109. no, byName, byType, constructor, autodetect
110. no, byName, byType, constructor, autocorrect
111. byName, byContent, constructor, autodetect
112. byName, byContent, setter, autodetect
113. What is no mode of autowiring?
114. Default setting which means no autowiring and you should use explicit bean reference for wiring.
115. Autowiring by property name.
116. Spring first tries to wire using autowire by constructor, if it does not work, Spring tries to autowire by byType.
117. Similar to byType, but type applies to constructor arguments.
118. What is byName mode of autowiring?
119. Default setting which means no autowiring and you should use explicit bean reference for wiring.
120. Autowiring by property name. Spring tries to match and wire its properties with the beans defined by the same names in the configuration file.
121. Spring first tries to wire using autowire by constructor, if it does not work, Spring tries to autowire by byType.
122. Similar to byType, but type applies to constructor arguments.
123. What is byType mode of autowiring?
124. Default setting which meas no autowiring and you should use explicit bean reference for wiring.
125. Autowiring by property name. Spring tries to match and wire its properties with the beans defined by the same names in the configuration file.
126. Spring first tries to wire using autowire by constructor, if it does not work, Spring tries to autowire by byType.
127. Autowiring by property type. Spring tries to match and wire a property if its type matches with exactly one of the beans name in configuration file.
128. What is constructor mode of autowiring?
129. Autowiring by property name. Spring tries to match and wire its properties with the beans defined by the same names in the configuration file.
130. Spring first tries to wire using autowire by constructor, if it does not work, Spring tries to autowire by byType.
131. Autowiring by property type. Spring tries to match and wire a property if its type matches with exactly one of the beans name in configuration file.
132. Similar to byType, but type applies to constructor arguments. If there is not exactly one bean of the constructor argument type in the container, a fatal error is raised.
133. What is autodetect mode of autowiring?
134. Similar to byType, but type applies to constructor arguments. If there is not exactly one bean

of the constructor argument type in the container, a fatal error is raised.

1. Autowiring by property name. Spring tries to match and wire its properties with the beans

defined by the same names in the configuration file.

1. Spring first tries to wire using autowire by constructor, if it does not work, Spring tries to autowire by byType.
2. Autowiring by property type. Spring tries to match and wire a property if its type matches with exactly one of the beans name in configuration file.
3. Can you inject null and empty string values in Spring?
4. Yes
5. No
6. How do you turn on annotation wiring?
7. Add <annotation-context:config /> to bean configuration.
8. Add <annotation-config /> to bean configuration.
9. Add <annotation-context-config /> to bean configuration.
10. Add <context:annotation-config/> to bean configuration.
11. What does @Required annotation mean?
12. This annotation indicates that bean property must be populated by the user.
13. This annotation indicates that bean property is required while saving the bean data to database.
14. This annotation simply indicates that the affected bean property must be populated at configuration time, through an explicit property value in a bean definition or through autowiring.
15. This annotation indicates that bean property is required while serializing the bean.
16. What is true about @Autowired annotation?
17. The @Autowired annotation can be used to autowire bean on the setter method.
18. This annotation provides more fine-grained control over where and how autowiring should be accomplished.
19. The @Autowired annotation can be used to autowire bean on the methods with arbitrary names and/or multiple arguments.
20. All of above.
21. What is ContextRefreshedEvent event?
22. This event is published when the Servlet Context is either initialized or refreshed.
23. This event is published when the HTTP Request is received.
24. This event is published when the HTTP Response is returned.
25. This event is published when the ApplicationContext is either initialized or refreshed.
26. What is ContextStartedEvent event?
27. This event is published when the Servlet Context is either initialized or refreshed.
28. This event is published when the HTTP Request is received.
29. This event is published when the ApplicationContext is started using the start() method on the ConfigurableApplicationContext interface.
30. This event is published when the HTTP Response is returned.
31. What is ContextStoppedEvent event?
32. This event is published when the Servlet Context is either initialized or refreshed.
33. This event is published when the ApplicationContext is stopped using the stop() method on the ConfigurableApplicationContext interface.
34. This event is published when the HTTP Request is received.
35. This event is published when the HTTP Response is returned.
36. What is ContextClosedEvent event?
37. This event is published when the Servlet Context is either initialized or refreshed.
38. This event is published when the HTTP Request is received.
39. This event is published when the HTTP Response is returned.
40. This event is published when the ApplicationContext is closed using the close() method on the ConfigurableApplicationContext interface.
41. What is RequestHandledEvent:event?
42. This event is published when the Servlet Context is either initialized or refreshed.
43. This event is published when the HTTP Request is received.
44. This event is published when the HTTP session is initialized or refreshed.
45. This event is published when the HTTP Request is serviced.
46. What is aspect?
47. Aspect is a way to do the dependency injection.
48. A module which has a set of APIs providing cross-cutting requirements.
49. Aspect is used to log information of application.
50. Aspectrepresents properties of spring based application.
51. What is Join point?
52. This represents a point in your application which joins two objects.
53. This represents a point in your object where you join values.
54. This represents a point in your object where you join injected values.
55. This represents a point in your application where you can plug-in AOP aspect.
56. What is Advice?
57. This is the way to instruct object to behave in certain manner.
58. This is used to inject values in objects.
59. This is the actual action to be taken either before or after the method execution.
60. This is not invoked during program execution by Spring AOP framework.
61. What is Pointcut?
62. This represents a point in your application where you can plug-in AOP aspect.
63. This is a set of one or more joinpoints where an advice should be executed.
64. This is used to inject values in objects.
65. This is invoked during program execution by Spring AOP framework.
66. What is Introduction?
67. An introduction represents a point in your application where you can plug-in AOP aspect.
68. This is used to inject values in objects.
69. This is not invoked during program execution by Spring AOP framework.
70. An introduction allows you to add new methods or attributes to existing classes.
71. What is Target object?
72. A represents a object in your application where you can plug-in AOP aspect.
73. The object being advised by one or more aspects, this object will always be a proxy object, also referred to as the advised object.
74. This is used to inject values in objects.
75. This is not invoked during program execution by Spring AOP framework.
76. What is Weaving?
77. Weaving is the process of injecting values in objects to create an advised object.
78. Weaving is the process of linking aspects with other application types or objects to create an advised object.
79. This is used to inject values in objects.
80. Weaving is used to check object dependencies.
81. What are the different points where weaving can be applied?
82. Compile time, load time
83. Compile time, run time
84. Run time
85. Compile time, load Time, Run time
86. What are the types of advice?
87. then, after, after-returning, after-throwing, around
88. When, after, after-returning, around
89. Where, after, after-returning, after-throwing, around
90. Before, after, after-returning, after-throwing, around
91. How before advice works?
92. Run advice before a class loads.
93. Run advice before a method execution.
94. Run advice before http response is to be returned.
95. Run advice before http request is to be processed.
96. How after-returning advice works?
97. Run advice after a class loads only if class loads successfully.
98. Run advice after a method execution only if method completes successfully.
99. Run advice after http response is returned only if http response is success.
100. Run advice after http request is processed with no exception.
101. How after-throwing advice works?
102. Run advice after a method execution only if method exits by throwing an exception.
103. Run advice after a class loads only if class throws exception during load time.
104. Run advice after http response is returned with error status.
105. Run advice after http request is processed and an exception occurred.
106. How around advice works?
107. Run advice before and after the advised method is invoked.
108. Run advice before and after a class is loaded.
109. Run advice before and after http response is returned.
110. Run advice before and after http request is processed.
111. Which of the following aspect implementation spring supports?
112. XML Schema based aspect implementation
113. @AspectJ based aspect implementation
114. Both of above.
115. None of above.
116. What are the types of the transaction management Spring supports?
117. Programmatic transaction management
118. Declarative transaction management
119. Both of above.
120. None of above.
121. What is Spring MVC framework?
122. Spring MVC framework is Model-Value-Class architecture and used to bind model data with values.
123. The Spring web MVC framework provides model-view-controller architecture and ready components that can be used to develop flexible and loosely coupled web applications.
124. Spring MVC framework is used for Transaction management for Web Applications.
125. Spring MVC framework is used for AOP for Web Applications.
126. What is @Controller annotation?
127. The @Controller annotation indicates that a particular class serves the role of a controller.
128. The @Controller annotation indicates how to control the transaction management.
129. The @Controller annotation indicates how to control the dependency injection.
130. The @Controller annotation indicates how to control the aspect programming.
131. What are the ways to access Hibernate by using Spring?
132. Inversion of Control with a Hibernate Template and Callback.
133. Extending HibernateDAOSupport and Applying an AOP Interceptor node.
134. Both of above.
135. None of above.
136. Which ORM Spring supports ?
137. Hibernate
138. iBatis
139. JPA
140. All of above.
141. Which of the following database is not supported using jdbcTemplate?
142. MySql
143. PostgresSql
144. NoSql
145. Oracle
146. How to get object of a service in spring framework?
147. Using new keyword
148. Using dependency injection
149. Which of the following is part of Data Access layer in Spring framework?
150. Beans
151. Aspects
152. JMS
153. Context
154. How to use ref keyword in beans.xml?
155. Using setter method only.
156. Using constructor argument only.
157. Using setter method and constructor argument both.
158. None of the above.
159. Core container has AOP as one of its module.
160. True
161. False
162. SpEL is part of core container.
163. False
164. True
165. Which class acts as IoC Container?
166. ServletContext
167. DispatcherServlet
168. C – ApplicationContext
169. None of the above
170. What stands true for spring framework?
171. Spring framework is a light weight framework.
172. Spring framework is a heavy weight framework.
173. Expression Language/ SpEL was introduced in which version of spring framework.
174. 1.0
175. 2.0
176. 3.0
177. 4.0
178. Can we integrate Struts with Spring.
179. Yes
180. No
181. By default a bean is lazily loaded.
182. True
183. False
184. By default a bean is eagerly loaded.
185. False
186. True
187. If a bean is scoped to HTTP request, scope is
188. session
189. global-session
190. prototype
191. request
192. If a bean is created once per Ioc Container, scope is
193. singleton
194. global-session
195. prototype
196. request
197. Thread scoped bean is introduced in which version of spring framework.
198. 1.0
199. 2.0
200. 3.0
201. 4.0
202. If a bean can be created any number of times, scope is
203. session
204. global-session
205. prototype
206. request
207. What is the scope of stateless bean?
208. global-session
209. singleton
210. prototype
211. request
212. What is the scope of stateful bean?
213. session
214. global-session
215. prototype
216. request
217. If a bean is scoped to HTTP session, scope is
218. global-session
219. session
220. prototype
221. request
222. How to handle shut down of IoC container?
223. Using shutdownHook()
224. Using shutdownHandler()
225. Using registerHook()
226. Using registerShutdownHook()
227. How bean life cycle can be controlled?
228. Using init() only
229. Using InitializingBean class only
230. Using DisposableBean class only
231. Using All of above
232. What is the scope of bean in portlet context?
233. session
234. global-session
235. prototype
236. request
237. How after advice works?
238. Run advice after a method execution regardless of its outcome.
239. Run advice after a class loads.
240. Run advice after http response is returned.
241. Run advice after http request is processed.
242. Which class is used to map a database row to a java object in spring?
243. ResultSet
244. RowMapper
245. RowSetMapper
246. ResultSetMapper
247. A bean must have id attribute in beans configuration file.
248. True
249. False
250. Which of the following class can be used to execute Sql queries in spring?
251. JdbcTemplate
252. JDBCHelper
253. DBHelper
254. DBTemplate
255. Which of the following class can be used to call Stored Procedures in spring?
256. SPHelper
257. JdbcTemplateCall
258. JdbcTemplate
259. SimpleJdbcCall
260. What is a DispatcherServlet?
261. DispatcherServlet is used for transaction management.
262. DispatcherServlet is used for AOP.
263. DispatcherServlet handles all the HTTP requests and responses.
264. DispatcherServlet is used for Dependency injection.
265. What is ACID in transactional management?
266. Accurate, Controlled, Isolation, Durability
267. Atomicity, Consistency, Independent, Done
268. Atomicity, Consistency, Isolation, Durability
269. Accurate, Controlled, Independent, Done
270. Where do you define DispatcherServlet?
271. In Beans configuration file.
272. web.xml file
273. Meta-inf/dispatcher.xml
274. Web-inf/ dispatcher.xml
275. What is true about BeanPostProcessor?
276. It is a concrete class.
277. It is an interface.
278. It is an abstract class.
279. None of the above.
280. What BeanPostProcessor does?
281. It processes beans once a bean is initialized.
282. It defines callback methods that you can implement to provide your own instantiation logic,
283. dependency-resolution logic etc.
284. It processes beans once a bean is loaded.
285. It processes beans once a bean exits.
286. Can be bean be configured to have an inner bean?
287. True
288. False
289. Can we inject value and ref both together in a bean?
290. True
291. False
292. Following class can be extended to create custom event in spring.
293. SpringEvent
294. Event
295. ApplicationEvent
296. None of above