Spring core Part 2

Q 1 - How can you inject Java Collection in Spring?
A - Using list, set, map or props tag.
B - Using lit, set, map or collection tag.
C - Using list, set, props or collection tag.
D - Using list, collection, map or props tag.

Q 2 - What is true about <list> collection configuration elements?
A - This helps in wiring a list of values, allowing duplicates.
B - This helps in wiring a list of values but without any duplicates.
C - This can be used to inject a collection of name-value pairs where name and value can be of any type.
D - This can be used to inject a collection of name-value pairs where the name and value are both Strings.

Q 3 - What is true about <set> collection configuration elements?
A - This helps in wiring a list of values, allowing duplicates.
B - This helps in wiring a list of values but without any duplicates.
C - This can be used to inject a collection of name-value pairs where name and value can be of any type.D - This can be used to inject a collection of name-value pairs where the name and value are
both Strings.

Q 4 - What is true about <map> collection configuration elements?
A - This helps in wiring a list of values, allowing duplicates.
B - This helps in wiring a list of values but without any duplicates.
C - This can be used to inject a collection of name-value pairs where name and value can be of any type.
D - This tag is not supported.

Q 5 - What is true about <props> collection configuration elements?
A - This helps in wiring a list of values, allowing duplicates.
B - This helps in wiring a list of values but without any duplicates.
C - This can be used to inject a collection of name-value pairs where name and value can be of any type.
D - This can be used to inject a collection of name-value pairs where the name and value are both Strings.

Q 6 - What is bean autowiring?

A - Autowiring lets Spring resolve collaborators otherbeans for your bean by inspecting the contents
of the BeanFactory without using <constructor-arg> and <property> elements.

B - Autowiring injects values in spring beans.

C - Autowiring injects one bean into another.

D - Autowiring helps in wiring a list of values, allowing duplicates.

Q 7 - Which are the different modes of autowiring?

A - no, byName, byType, constructor, autodetect

B - no, byName, byType, constructor, autocorrect

C - byName, byContent, constructor, autodetect

D - byName, byContent, setter, autodetect

Q 8 - What is no mode of autowiring?

A - Default setting which means no autowiring and you should use explicit bean reference for wiring.

B - Autowiring by property name.

C - Spring first tries to wire using autowire by constructor, if it does not work, Spring tries to autowire by byType.

D - Similar to byType, but type applies to constructor arguments.

Q 9 - What is byName mode of autowiring?

A - Default setting which means no autowiring and you should use explicit bean reference for wiring.

B - Autowiring by property name. Spring tries to match and wire its properties with the beans defined by the same names in the configuration file.

C - Spring first tries to wire using autowire by constructor, if it does not work, Spring tries to autowire by byType.

D - Similar to byType, but type applies to constructor arguments.

Q 10 - What is byType mode of autowiring?

A - Default setting which meas no autowiring and you should use explicit bean reference for wiring.

B - Autowiring by property name. Spring tries to match and wire its properties with the beans defined by the same names in the configuration file.

C - Spring first tries to wire using autowire by constructor, if it does not work, Spring tries to autowire by byType.

D - 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.

Q 11 - What is constructor mode of autowiring?
A - Autowiring by property name. Spring tries to match and wire its properties with the beans defined by the same names in the configuration file.
B - Spring first tries to wire using autowire by constructor, if it does not work, Spring tries to autowire by byType.
C - 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.
D - 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.

Q 12 - What is autodetect mode of autowiring?
A - 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.
B - Autowiring by property name. Spring tries to match and wire its properties with the beans defined by the same names in the configuration file.
C - Spring first tries to wire using autowire by constructor, if it does not work, Spring tries to autowire by byType.
D - 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.

Q 13
Can you inject null and empty string values in Spring?
A - Yes
B – No

Q 14 - How do you turn on annotation wiring?
A - Add <annotation-context:config /> to bean configuration.
B - Add <annotation-config /> to bean configuration.
C - Add <annotation-context-config /> to bean configuration.
D - Add <context:annotation-config/> to bean configuration.

Q 15- What does @Required annotation mean?
A - This annotation indicates that bean property must be populated by the user.

B - This annotation indicates that bean property is required while saving the bean data todatabase.
C - 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.
D - This annotation indicates that bean property is required while serializing the bean.

Q 16 - What is true about @Autowired annotation?
A - The @Autowired annotation can be used to autowire bean on the setter method.
B - This annotation provides more fine-grained control over where and how autowiring should be
accomplished.
C - The @Autowired annotation can be used to autowire bean on the methods with arbitrary names and/or multiple arguments.
D - All of above.

Q 17 - What is ContextRefreshedEvent event?
A - This event is published when the Servlet Context is either initialized or refreshed.
B - This event is published when the HTTP Request is received.
C - This event is published when the HTTP Response is returned.
D - This event is published when the ApplicationContext is either initialized or refreshed.

Q 18 - What is ContextStartedEvent event?
A - This event is published when the Servlet Context is either initialized or refreshed.
B - This event is published when the HTTP Request is received.
C - This event is published when the ApplicationContext is started using the start method on the ConfigurableApplicationContext interface.
D - This event is published when the HTTP Response is returned.

Q 19 - What is ContextStoppedEvent event?
A - This event is published when the Servlet Context is either initialized or refreshed.
B - This event is published when the ApplicationContext is stopped using the stop method on the
ConfigurableApplicationContext interface.
C - This event is published when the HTTP Request is received.
D - This event is published when the HTTP Response is returned.

Q 20 - What is ContextClosedEvent event?
A - This event is published when the Servlet Context is either initialized or refreshed.

B - This event is published when the HTTP Request is received.
C - This event is published when the HTTP Response is returned.
D - This event is published when the ApplicationContext is closed using the close method on the ConfigurableApplicationContext interface.

Q 21 - What is RequestHandledEvent:event?
A - This event is published when the Servlet Context is either initialized or refreshed.
B - This event is published when the HTTP Request is received.
C - This event is published when the HTTP session is initialized or refreshed.
D - This event is published when the HTTP Request is serviced.

Q 22 - What is aspect?
A - Aspect is a way to do the dependency injection.
B - A module which has a set of APIs providing cross-cutting requirements.
C - Aspect is used to log information of application.
D - Aspect represents properties of spring based application.

Q 23 - What is Join point?
A - This represents a point in your application which joins two objects.
B - This represents a point in your object where you join values.
C - This represents a point in your object where you join injected values.
D - This represents a point in your application where you can plug-in AOP aspect.

Q 24 - What is Advice?
A - This is the way to instruct object to behave in certain manner.
B - This is used to inject values in objects.
C - This is the actual action to be taken either before or after the method execution.
D - This is not invoked during program execution by Spring AOP framework.

Q 25 - What is Pointcut?
A - This represents a point in your application where you can plug-in AOP aspect.
B - This is a set of one or more joinpoints where an advice should be executed.
C - This is used to inject values in objects.
D - This is invoked during program execution by Spring AOP framework.