|  |  |  |
| --- | --- | --- |
| **Q 1** | Which of the following is NOT the feature of Spring IoC?  A - Dependency Injection  B - Configurable Factory  C - Loose Coupling and Autowiring  D - Object Relational Mapping | S |
| **Q2.** | Observe the code...  <bean id=*"resourceBean2"* class=*" GlobalInvestment"*>  <property name=*"headOfficeAddress"* ref=*"address"*/>  </bean>  What is 'headOfficeAddress' and 'address' in above code?  [A -](javascript:void(0);) Both are field names  [B -](javascript:void(0);) The first is field name and second is bean reference name  [C - The](javascript:void(0);) first is property name and second is bean reference name  [D - Both](javascript:void(0);) are bean reference names | S |
| **Q 3** | @Component ("resourceBean")  @Scope("singleton")  **public** **class** GlobalInvestment {  **private** @Value ("GI Pvt. Ltd.")String firmName;  Observing above code, which statement is TRUE?  [A -](javascript:void(0);) Bean will be declared as Configuration Bean  [B -](javascript:void(0);) Scope of the bean is 'Singleton' in JVM  [C - The](javascript:void(0);) instance field in an object will be initialized by a given string  [D - The](javascript:void(0);) @Value annotation is applicable on property and not on private fields. | S |
| Q4. | What may be the purpose of method postProcessAfterInitialization() of InstantiationAwareBeanPostProcessor?  [A - To](javascript:void(0);) add new beans programmatically in the context  [B -](javascript:void(0);) To apply final touches to the created bean like wrapping in proxy  [C - To](javascript:void(0);) validate initialized state of a bean otherwise set defaults  D - None of these | C |
| Q5. | What kind of scope a stateful bean should be given?  [A - prototype](javascript:void(0);)  [B - global-session](javascript:void(0);)  [C - session](javascript:void(0);)  [D - singleton](javascript:void(0);) | S |
| **Q 6** | What is the correct difference from following between @Resource and @Autowired  A - The first is standard and second is third party annotation  [B - The](javascript:void(0);) first does lookup default 'by type' and second does injection default 'by name'.  [C -](javascript:void(0);) The first needs @Qualifier for while second does not  [D - The](javascript:void(0);) first takes care of duplicate beans while second does not | C |
| Q 7 | Which implementation gives us handle to shut down of IoC container?  [A - AnnotationConfig](javascript:void(0);)ApplicationContext  [B -](javascript:void(0);)ConfigurableApplicationContext  [C - Bean](javascript:void(0);)Factory  [D -](javascript:void(0);) ClasspathXmlApplicationContext | S |
| Q 8 | Which annotation out of following will create singleton beans on demand?  A - @Eager(false)  [B -](javascript:void(0);) @Eager(true)  [C -](javascript:void(0);) @Lazy(true)  [D -](javascript:void(0);) @Lazy(false) | S |
| Q 9 | What is the purpose of destroy()/PreDestroy for a bean?  A - A code which creates new bean for next code to execute  [B -](javascript:void(0);) A code which calls Garbage Collection to make heap space free  [C -](javascript:void(0);) A code which destroys bean  [D -](javascript:void(0);) To ensure graceful retirement for a bean | S |
| Q 10 | Which out of following is NOT sub-annotation of @Component  [A - @Bean](javascript:void(0);)  B - @Service  [C - @Repository](javascript:void(0);)  [D - @Controller](javascript:void(0);) | S |
| Q 11 | Let a 'resBean' has been declared as...  @Component("resBean")  **public** **class** GlobInvest {  **private** @Value ("GI Pvt. Ltd.")String fName;  Let the 'report' bean has been declared as...  @Service("report")  **public** **class** Report {  **private** @Value ("#{resBean.fName}") String firmName;  Assume necessary properties are written in 'GlobInvest'. Which statement out of following is true?  [A -](javascript:void(0);) The field 'firmName' will store null as Report bean does not have dependency set with 'GlobInvest'.  B - The field 'firmName' is initialized with value 'GI Pvt. Ltd.".  [C - The](javascript:void(0);) @Value does not work on 'field' but on property. So compile time error.  [D - The](javascript:void(0);) @Value does not work on 'private' field thus run time error. | C |
| Q 12 | Which of the following is a most correct assertion about spring?  [A - Spring enables developers to develop enterprise-class applications using POJOs.](javascript:void(0);)  [B - Spring is organized in a modular fashion.](javascript:void(0);)  [C - Spring](javascript:void(0);) is susceptible to testing and has special support to test code.  [D - All of above.](javascript:void(0);) | S |
| Q 13 | Let the 'GlobInvestment' has been declared as...  **public** **class** GlobalInvestment {  **private** Set<String> directorsPanel;  **public** Set<String> getDirectorsPanel() {  **return** directorsPanel;  }  **public** **void** setDirectorsPanel(Set<String> diPanel) {  **this**.directorsPanel = diPanel;  }  }  Choose nearest configuration for initializing 'directorsPanel'...  A - <bean id=*"resourceBean"* class=*"GlobalInvestment"*>  <property name=*"directorsPanel"*>  <set>  <value>Mr. Malhotra</value>  <value>Mr. Gihrotra</value>  </set>  </property>  </bean>  B - <bean id=*"resourceBean"* class=*"GlobalInvestment"*>  <property name=*"directorsPanel"*>  <list>  <value>Mr. Malhotra</value>  <value>Mr. Gihrotra</value>  </list>  </property>  </bean>  C - <bean id=*"resourceBean"* class=*"GlobalInvestment"*>  <property name=*"directorsPanel"*>  <map>  <entry key=*"director1"*>  <value>Mr. Malhotra</value>  </entry>  <entry key=*"director2"*>  <value>Mr. Gihrotra</value>  </entry>  </map>  </property>  </bean>  D - <bean id=*"resourceBean"* class=*"GlobalInvestment"*>  <property name=*"directorsPanel"*>  <props>  <prop key=*"director1"*>Mr. Malhotra</prop>  <prop key=*"director2"*>Mr. Gihrotra</prop>  </props>  </property>  </bean> | S |
| Q14 | Observe following bean definition...  @Repository("empDao")  **public** **class** EmpDao {  **private** IGlobInvestment resource;  @Resource  **public** **void** setResource(IGlobInvestment resource){  **this**.resource = resource;  }  Here, 'IGlobalInvestment' is an interface with implementations in two beans with names...'resourse1' and 'resource2'. What is most probable possibility out of following...  [A -](javascript:void(0);) The autowiring is not possible on interface. Above code thus will show compile time error.  [B - This](javascript:void(0);) situation needs @Autowired and not @Resourse  [C -](javascript:void(0);) The @Resource annotation does 'byName' autowiring if failed with 'byType'. So, autowiring will be done successfully.  [D -](javascript:void(0);) Will throw 'No Unique Bean Exception'. | C |
| Q15 | Observe following code...  @Component("globalInvest")  **public** **abstract** **class** GlobalInvestment {}  @Component("localInvest")  **public** **class** LocalInvestment **extends** GlobalInvestment {}  Which of the following statements is true while creating the context...  [A -](javascript:void(0);) The 'locaInvest' bean is created while 'globalInvest' not created being an 'abstract'.  [B -](javascript:void(0);) Context creation will fail while creating 'globalInvest' as it has been declared as 'abstract'.  [C -](javascript:void(0);) Spring context will override 'abstract'ness of 'globalInvest' and will create both beans.  [D -](javascript:void(0);) None of the bean will be created. | C |
| Q 16 | What is JdbcTemplate?  A - Its a DataSource complient of javax.sql.DataSource  [B -](javascript:void(0);) Its a connection pool feature given by Spring-JDBC  [C -](javascript:void(0);) Its a JDBC wrapper providing JDBC abstraction  [D -](javascript:void(0);) Its an implementation of ORM | S |
| Q 17 | Observe the code...  <bean id=*"resourceBean2"* class=*" packContext.GlobalInvestment"*>  <constructor-arg>  <value>Gobal Investment Pvt. Ltd.</value>  </constructor-arg>  <constructor-arg>  <value>Ladder of success and Delegency</value>  </constructor-arg>  </bean>  Which constructor of class GlobalInvestment out of following will be invoke for object creation?  [A -](javascript:void(0);)  **public** GlobalInvestment(String firmName, String companyMotto) {  **super**();  **this**.firmName = firmName;  **this**.companyMotto = companyMotto;  }  [B -](javascript:void(0);)  **public** GlobalInvestment(String firmName, int companyRank) {  **super**();  **this**.firmName = firmName;  **this**.companyRank = companyRank;  }  [C -](javascript:void(0);)  **public** GlobalInvestment(  }  [D -](javascript:void(0);)  **public** GlobalInvestment(String firmName, String companyMotto, **int** globalRank, Address headOfficeAddress) {  **super**();  **this**.firmName = firmName;  **this**.companyMotto = companyMotto;  **this**.globalRank = globalRank;  **this**.headOfficeAddress = headOfficeAddress;  } |  |
| Q 18 | The below DataSource configuration ideally should go to...  <bean id=*"ds"*  class=*"org.springframework.jdbc.datasource.DriverManagerDataSource"*>  <property name=*"driverClassName"* value=*"oracle.jdbc.OracleDriver"*/>  <property name=*"url"* value=*"jdbc:oracle:thin:@localhost:1521:XE"*/>  <property name=*"username"* value=*"scott"*/>  <property name=*"password"* value=*"tiger"*/>  </bean>  A - The web.xml  [B -](javascript:void(0);) The servlet-context.xml  [C -](javascript:void(0);) The hibernate.cfg.xml  [D -](javascript:void(0);) The applicationContext.xml | S |
| Q 19 | A meaning of following transactional attribute on a method...  @Transactional(propagation=Propagation.***MANDATORY***)  A - Method needs a transaction but participate in propagated transaction. Can't create its own.  [B -](javascript:void(0);) Method needs a transaction. Either participate in propagated transaction otherwise creates its own.  [C -](javascript:void(0);) Method can run without transaction if transaction is not propagated.  [D -](javascript:void(0);) Always create a new transaction for a method thus method can run in new transaction. | C |
| Q 20 | The method transferAmount(Account from, Account to, float amount) withdraws an 'amount' from an Account and deposits it to another Account. For withdrawal and for deposit, methods like withdraw(Account from, Amount) and deposit(Account to, Amount) have been defined with @Required transactional attribute. Expecting method transferAmount() to start its own transaction, which of the following transactional attribute should be used?  A - Propagation.MANDATORY  [B -](javascript:void(0);) Propagation.REQUIRES\_NEW  [C -](javascript:void(0);) Propagation.NEVER  [D -](javascript:void(0);) Propagation.SUPPORTS | C |
| Q 21 | @Component ("resourceBean")  **public** **class** GlobalInvestment {  **private** @Value ("GI Pvt. Ltd.")String firmName;  @Value("L and T Infotech.")  **public** **void** setFirmName(String firmName){  **this**.firmName = firmName;  }  Observing above code, what will be the final initialized value of 'firmName'?  [A -](javascript:void(0);) Its Null  [B -](javascript:void(0);) Its ‘L and T Infotech.'  [C -](javascript:void(0);) Its 'GI Pvt. Ltd.'  [D -](javascript:void(0);) Its 'resourceBean'. | C |
| Q 22 | What may be the purpose of method annotated with @PostConstruct?  [A -](javascript:void(0);) To apply common logic of initialization for all beans.  [B -](javascript:void(0);) To apply common logic of instantiation for all beans.  [C - To](javascript:void(0);) execute cleaning code while shutting down of container  D - None of these | S |
| Q 23 | Which out of following is a valid reason for the 'byType' autowiring may fail but 'byName' will be successful?  A - If there are more than one beans of same id.  [B -](javascript:void(0);) If there is no bean defined in context of given type.  [C -](javascript:void(0);) If there are two beans of same type but different ids  [D -](javascript:void(0);) If there are two beans of different type as well as different ids. | C |
| Q 24 | What is the valid difference between BeanFactory and ApplicationContext?  [A -](javascript:void(0);) BeanFactory does not allow Dependency Injection while ApplicationContext allows.  [B -](javascript:void(0);) BeanFactory does not allow Multiple context while ApplicationContext allows.  [C - Bean](javascript:void(0);)Factory does not allow declarative bean approach while ApplicationContext allows.  [D -](javascript:void(0);) BeanFactory does not allow 'Parent-Child' bean relationship while ApplicationContext allows. | S |
| Q 25 | Let a 'resBean' has been declared as...  @Service  **public** **class** Operands {  @Value("150") **private** **int** value1;  @Value("200") **private** **int** value2;  Let the 'operator' bean has a field declaration as...  @Value("#{operands.value1 == operands.value2}") **private** **boolean** equalTest;  Assume other necessary syntactical things are all at their places. What will be the resultant value for field 'equalTest'?  [A -](javascript:void(0);) The @Value annotation can't be applied to private field so will show compile time error.  B - The expression language does not allow the way of expression written while initializing field 'equalTest'.  [C -](javascript:void(0);) The resultant value of a field is... FALSE  [D -](javascript:void(0);) The resultant value of a field is... TRUE | C |
| Q 26 | <bean id=*"bean"* class=*" packInnerBean.GlobalInvestment"*>  <property name=*"address"*>  <bean id=*"address"* class=*"packInnerBean.Address"*>  <property name=*"line1"*>  <value>207, Lok Center</value>  </property>  </bean>  </property>  </bean>  Assuming respective properties are already existing in respective bean, Which one is most suitable statement out of following?  [A -](javascript:void(0);) The 'address' bean is visible to client  [B -](javascript:void(0);) The 'address' bean is not visible to other beans  [C -](javascript:void(0);) A bean can't be declared within another bean  [D -](javascript:void(0);)The attribute 'inner=true' is missing for 'address' bean | C |
| Q 27 | Observe following code...  **public** **class** GlobInvest {}  <bean id=*"globInvest"* class=*"GlobInvest"* abstract=*"true"* >  </bean>  Which of the following statements is true while creating the context...  [A -](javascript:void(0);) Spring does not create a bean as it is declared as 'abstract'.  [B -](javascript:void(0);) Spring creates a bean as class has not been declared as 'abstract'  [C -](javascript:void(0);) Exception is thrown as class has not been declared as 'abstract'  [D -](javascript:void(0);) None of these | S |
| Q 28 | <bean id=*"ds"*  class=*"org.springframework.jdbc.datasource.DriverManagerDataSource"*>  <property name=*"driverClassName"* value=*"oracle.jdbc.OracleDriver"*/>  <property name=*"url"* value=*"jdbc:oracle:thin:@localhost:1521:XE"*/>  <property name=*"username"* value=*"scott"*/>  <property name=*"password"* value=*"tiger"*/>  </bean>  The most suitable statement for DriverManagerDataSource out of following is...  A - It is compliant with javax.sql.DataSource with single connection pool.  [B -](javascript:void(0);) It is oracle.jdbc.OracleDriver complient Data Source.  [C -](javascript:void(0);) It is java.sql.Connection bean support given by Spring API  [D -](javascript:void(0);) Its a database driver. Here it is Oracle Driver. | C |
| Q 29 | A meaning of following transactional attribute on a method...  @Transactional(propagation=Propagation.***REQUIRED***)  A - Method needs a transaction but participate in propagated transaction. Can't create its own.  [B -](javascript:void(0);) Method can run without transaction if transaction is not propagated.  [C -](javascript:void(0);) Always create a new transaction for a method thus method can run in new transaction.  [D -](javascript:void(0);) Method needs a transaction. Either participate in propagated transaction otherwise creates its own. | S |
| Q 30 | The method createNewAccount(Account newAccount) creates a new account in database by inserting a new record. This method is always to be executed in conjunction with another methods like newTransaction() atomically. Which of the following transactional attribute should be used for createNewAccount()?  A - Propagation.NOT\_SUPPORTED  [B -](javascript:void(0);) Propagation.REQUIRES\_NEW  [C -](javascript:void(0);) Propagation.MANDATORY  [D -](javascript:void(0);) Propagation.SUPPORTS | C |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SN** | **Key** | **SN.** | **Key** | **SN** | **Key** |
| 1 | **D** | 11 | **B** | 21 | **B** |
| 2 | **C** | 12 | **D** | 22 | **D** |
| 3 | **C** | 13 | **A** | 23 | **C** |
| 4 | **B** | 14 | **D** | 24 | **B** |
| 5 | **A** | 15 | **A** | 25 | **C** |
| 6 | **A** | 16 | **C** | 26 | **B** |
| 7 | **B** | 17 | **A** | 27 | **A** |
| 8 | **C** | 18 | **D** | 28 | **A** |
| 9 | **D** | 19 | **A** | 29 | **D** |
| 10 | **A** | 20 | **B** | 30 | **C** |