



CORESPRINGV3.2^{Q&As}

Core-Spring (based on Spring 3.2)

Pass SpringSource CORESPRINGV3.2 Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

<https://www.passapply.com/corespringv3-2.html>

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by
SpringSource Official Exam Center

-  **Instant Download** After Purchase
-  **100% Money Back** Guarantee
-  **365 Days** Free Update
-  **800,000+** Satisfied Customers





QUESTION 1

State which type of object the JdbcTemplate can be used to query (select one)

- A. Simple types (int, long, String, etc)
- B. Generic Maps
- C. Domain Objects
- D. All of the above

Correct Answer: D

QUESTION 2

When injecting scalar/literal values into Spring beans, which of the following statements is true? (select one)

- A. Scalar values cannot be injected into setters or constructors with primitive type parameters
- B. Spring performs automatic type conversion for certain data types, such as String to int
- C. In XML Spring configuration, you can inject scalar values using the ref attribute of the tag
- D. All of the above

Correct Answer: B

QUESTION 3

You want to externalize constants from your Spring XML configuration file into a .properties file to your Spring beans. Which mechanism could you use? (select one)

- A. Use a PropertyEditor
- B. Use a PropertyPlaceholderConfigurer
- C. Use
- D. Use a BeanPostProcessor

Correct Answer: B

QUESTION 4

Using declarative transaction management, by default a transaction rolls back if:

- A. Any uncaught exception that inherits from Exception has been thrown



- B. Any uncaught exception that inherits from RuntimeException has been thrown
- C. Any uncaught Throwable that inherits from Throwable has been thrown
- D. Null is being returned by a non-void method

Correct Answer: B

QUESTION 5

Which of the following statements comparing traditional RMI with Spring RMI is NOT true? (select one)

- A. Traditional RMI requires the client to catch RemoteExceptions, but Spring's approach does not
- B. Traditional RMI requires implementing java.rmi.Remote, but Spring's approach does not
- C. Both approaches require you to extend the UnicastRemoteObject class on the server side
- D. Both approaches require method parameters and return values to implement java.io.Serializable

Correct Answer: C

[Latest CORESPRINGV3.2 Dumps](#)

[CORESPRINGV3.2 Practice Test](#)

[CORESPRINGV3.2 Braindumps](#)