**Question: 1** What is an after returning advice? Select a unique answer.  
  
**Your Answer:** None  
  
**Correct Answer:** Advice to be executed after a join point completes without throwing an exception  
  
**Description:** None

**Question: 2** Select method´s signatures that match with the following pointcut:   
execution(\* com.test.service..\*.\*(\*))  
  
**Your Answer:** None  
  
**Correct Answer:** void com.test.service.MyServiceImpl#transfert(Money amount)  
  
**Description:** None

**Question: 3** Using the Spring AOP framework, what is the visibility of the method matches by the following join   
point?   
@Pointcut("execution(\* \*(..))")   
private void anyOperation() {};  
  
**Your Answer:** None  
  
**Correct Answer:** Public methods  
  
**Description:** Due to the proxy-based nature of Spring´s AOP framework, protected methods are by definition not   
intercepted, neither for JDK proxie nor for CGLIB proxies. As a consequence, any given pointcut will be matched against public methods only! To intercept private and protected methods, AspecJ weaving should be used instead of the Spring´s proxy-bases AOP framework.

**Question: 4** Which one is the correct statement about AOP proxy?  
  
**Your Answer:** None  
  
**Correct Answer:** AOP proxies are created by Spring in order to implement the aspect contracts  
  
**Description:** 1. An object created by the AOP framework in order to implement the aspect contracts   
2. If the target object does not implement any interfaces then a CGLIB proxy will be created.   
You could also use CGLIB proxy instead of JDK dynamic proxy   
3. If the target object does not implement any interfaces then a CGLIB proxy will be created.   
4. When CGLIB proxy is used, final methods cannot be advised, as they cannot be overridden.

**Question: 5** What is a pointcut?  
  
**Your Answer:** None  
  
**Correct Answer:** An expression to identify joinpoints  
  
**Description:** None

**Question: 6** What is an after throwing advice? Select a unique answer.  
  
**Your Answer:** None  
  
**Correct Answer:** Advice to be executed if a method exits by throwing an exception  
  
**Description:** 1. A before advice could throw an exception   
2. An after throwing advice is executed if a method exits by throwing an exception   
3. An advice that executes before a join point is named a before advice   
4. Spring supports after throwing advices

**Question: 7** What are the unique correct answers about Spring AOP support?  
  
**Your Answer:** None  
  
**Correct Answer:** A point cut could select methods that have a custom annotation  
  
**Description:** The @annotation designator enables to select methods that are annotated by a given   
annotation

**Question: 8** What is an advice? Select a unique answer.  
  
**Your Answer:** None  
  
**Correct Answer:** An action taken by an aspect at a particular join point  
  
**Description:** None

**Question: 9** Considering 2 classes AccountServiceImpl and ClientServiceImpl. Any of these 2 classes inherits from   
each other. What is the result of the pointcut expressions?   
execution(\* \*..AccountServiceImpl.update(..))   
&& execution(\* \*..ClientServiceImpl.update(..))  
  
**Your Answer:** None  
  
**Correct Answer:** No joint point is defined  
  
**Description:** Considering 2 classes AccountServiceImpl and ClientServiceImpl. Any of these 2 classes inherits from   
each other. What is the result of the pointcut expressions?   
execution(\* \*..AccountServiceImpl.update(..))   
&& execution(\* \*..ClientServiceImpl.update(..))   
Poincut expression could not satisfied both first and second execution point. Do not confuse the &&   
operator and || operator.

**Question: 10** Using the Spring AOP framework, what are the joinpoint methods of the following pointcut   
expressions?   
execution(public \* \*(..))  
  
**Your Answer:** None  
  
**Correct Answer:** The execution of all public method  
  
**Description:** The execution of all public method

[Finish](https://www.javatpoint.com/examaccess)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

|  |  |
| --- | --- |
|  |  |