**Question: 1** What is an after returning advice? Select a unique answer.  
  
  
**Correct Answer:** Advice to be executed after a join point completes without throwing an exception

**Question: 2** Select method´s signatures that match with the following pointcut:   
execution(\* com.test.service..\*.\*(\*))  
  
  
**Correct Answer:** void com.test.service.MyServiceImpl#transfert(Money amount)

**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() {};  
  
  
**Correct Answer:** Public methods

**Question: 4** Which one is the correct statement about AOP proxy?  
  
  
**Correct Answer:** AOP proxies are created by Spring in order to implement the aspect contracts

**Question: 5** What is a pointcut?  
  
  
**Correct Answer:** An expression to identify joinpoints

**Question: 6** What is an after throwing advice? Select a unique answer.  
  
  
**Correct Answer:** Advice to be executed if a method exits by throwing an exception

**Question: 7** What are the unique correct answers about Spring AOP support?  
  
  
**Correct Answer:** A point cut could select methods that have a custom annotation

**Question: 8** What is an advice? Select a unique answer.  
  
  
**Correct Answer:** An action taken by an aspect at a particular join point

**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(..))  
  
  
  
**Correct Answer:** No joint point is defined  
  
**Description:** Considering 2 classes AccountServiceImpl and ClientServiceImpl. Any of these 2

**Question: 10** Using the Spring AOP framework, what are the joinpoint methods of the following pointcut   
expressions?   
execution(public \* \*(..))  
  
**Correct Answer:** The execution of all public method