<https://dzone.com/articles/java-unit-testing-interview>

**Java Unit Testing Interview Questions**

The article presents some of the frequently asked interview questions in relation with unit testing with Java code. Please suggest other questions tthat you came across and I shall include in the list below.

1. What is unit testing? Which unit testing framework did you use? What are some of the common Java unit testing frameworks?   
   **Ans**: Read the definition of Unit testing on [Wikipedia page for unit testing](http://en.wikipedia.org/wiki/Unit_testing). Simply speaking, unit testing is about testing a block of code in isolation. There are two popular unit testing framework in Java named as Junit, TestNG.
2. In SDLC, When is the right time to start writing unit tests?   
   **Ans**: Test-along if not test-driven; Writing unit tests towards end is not very effective. Test-along technique recommends developers to write the unit tests as they go with their development.
3. With Junit 4, do we still need methods such as setUp and tearDown?   
   **Ans**: No. This is taken care with help of @Before and @After annotations respectively
4. What do following junit test annotations mean?   
   **Ans**: Following is a list of frequently used JUnit 4 annotations:@Test (@Test identifies a test method)  
   @Before (Ans: @Before method will execute before every JUnit4 test)@After (Ans: @After method will execute after every JUnit4 test)@BeforeClass (Ans: @BeforeClass method will be executed before JUnit test for a Class starts)@AfterClass (Ans: @AfterClass method will be executed after JUnit test for a Class is completed)@Ignore (@Ignore method will not be executed)
5. How do one do exception handling unit tests using @Test annotation?   
   **A** **ns**: @Test(expected={exception class}. For example: @Test(expected=IllegalArgumentException.class)
6. Write a sample unit testing method for testing exception named as IndexOutOfBoundsException when working with ArrayList?
7. @Test(expected=IndexOutOfBoundsException.class)
8. public void outOfBounds() {
9. new ArrayList<Object>().get(1);

}

1. Write a sample unit testing method for testing timeout?
2. @Test(timeout=100)
3. public void infinity() {
4. while(true);

}

1. What is unit testing method naming convention that you follow?   
   **Ans**: Unit tests name should act like the documentation giving a clear idea on what functionality is tested. There are various techniques that could be used to name unit tests. Some of them are following: Given…When…Then.. or Should…Then… etc.
2. What are called as test smells in relation with unit testing?   
   **Ans**: Following could be termed as test smells: Multiple assertions within one unit test, long-running unit tests etc
3. What are top 2-3 advantages of writing unit tests?   
   **Ans**: Design testability, Code testability, Code maintainability as good unit tests enforces OO principles such as Single Responsitbility etc which enables people avoid code smells such as long classes, long methods, large conditionals etc.
4. What are top 2-3 characteristics of Hard-to-test code?   
   **Ans**: Long methods, Long conditionals, mixing concerns, bulky constructors
5. What is mocking & stubbing? What are key differences between them? Did you use any mocking framework?   
   **Ans**: Read details on What is difference between stubbing and mocking on this [page](http://stackoverflow.com/questions/346372/whats-the-difference-between-faking-mocking-and-stubbing).. There are various different Java mocking framework such as Mockito, EasyMock etc.
6. What are top 2-3 advantages of mocking?   
   **Ans**: 1. Verify interactions 2. Test the block of code in isolation
7. How is code cyclomatic complexity related to unit tests?   
   **Ans**: As code cyclomatic complexity is determined based on number of decision points within the code and hence execution paths, higher cyclomatic complexity makes it difficult to attain achieve test/code coverage.
8. What is code coverage? How is it measured? Name some JUnit code coverage tools? **Ans**: Cobertura, EclEmma
9. Name some code smells which makes it hard to achieve high code coverage?   
   **Ans**: Long method, Large conditionals
10. What is test-driven development (TDD)?   
    **Ans**: Read the details on [Wikipedia page for TDD](http://en.wikipedia.org/wiki/Test-driven_development).