```
-----------------------------------
2) default void onTestSuccess(ITestResult result)

   -> This method get called when test case is succeed

   -> If we want to handle some activity after test case success,
      those activity we can define inside this method.

3) default void onTestFailure(ITestResult result)

   -> This method get called when test case fails

   -> If we want to perform some activity when test case fails,
      those activity we can define inside this method
```

```
4) default void onTestSkipped(ITestResult result)

   -> This method get called when test case is skipped

   -> If we want to perform some activity when test case is skipped,
      those activity we can define inside this method
```

```
Example
	@Test(timeOut = 2000)
	 public void m2Test() throws InterruptedException
	 {
		System.out.println(" i am inside m2Test");
		// chrombrowser testing
		Thread.sleep(3000);
	 }
	 @Test(dependsOnMethods ="m2Test" )
	 public void m3Test()
	 {
		System.out.println(" i am inside m3Test");
		//firefox browser testing
	 }

	 @Override
	 public void onTestSkipped(ITestResult result) {
		System.out.println("inside onTestSkipped --");
	 }
--------------------------------------
```

```
--------------------------------------
5)default void onTestFailedButWithinSuccessPercentage(ITestResult result)

-> This method get called when some of assert conditions are passed and
   some of assert condition failed and final result of test case is fail
   but successPercentage attribute value is satisfied.
```

```
public class Test101
{

        int cnt=0;
        @Test(successPercentage = 50,invocationCount =5)
        public void m1Test()
        {
                cnt++;
                if(cnt<3)
                {
                        assertTrue(false);//2
                }
                assertTrue(true);//3
        }
}
```

```
Listener class method

        @Override
        public void onTestFailedButWithinSuccessPercentage(ITestResult result) {
                System.out.print("inside onTestFailedButWithinSuccessPercentage" );
        }

--------------------------
```

```
6) default void onTestFailedWithTimeout(ITestResult result)

-> This method get called when test case get failed due to time out

-> To specify time out value we need to "timeOut" attribute in @Test annotation

        @Test(timeOut = 2000)
        public void m1Test() throws InterruptedException
        {
                Thread.sleep(5000);
        }
-----------------------------------
Listener class method

        @Override
        public void onTestFailedWithTimeout(ITestResult result) {
                System.out.println("inside onTestFailedWithTimeout ");
        }
```

```
7) default void onStart(ITestContext context)

Before start execution of all test methods if we want to perform some
activity, those activity we can define inside this method

-> onStart() method get called before onTestStart()

----------------------------------------------------
```

```
---------------------------------------------------
8) default void onFinish(ITestContext context)

On completion of all test execution if we want to peform some activity,
those activity we can define inside this method
============================================================
```

```
==================================================================
2) IExecutionListener

  This listner is used to monitor when a TestNG run starts and ends

  It is present inside org.testng package

  It is child interface of ITestNGListener

  It has two methods and those methods are as below

  1) default void onExecutionStart()

     Invoked before the TestNG run starts.

  2) default void onExecutionFinish()

     Invoked once all the suites have been run.
=====================================
=====================================
3) ISuiteListener

  -> It is present in org.testng package

  -> It is child interface of ITestNGListener

  -> This listener is for test suites

  -> It has two methods

   1) default void onStart(ISuite suite)

     This method is invoked before the SuiteRunner starts

   2) default void onFinish(ISuite suite)

     This method is invoked after the SuiteRunner has run all the tests in the suite
================================================
```