* Junit is a popular testing framework that simplifies the unit testing in java applications during development phase itself.
* @BeforeClass runs once before any test methods are executed.
* @Before runs before each test method.
* @AssertEquals, @AssertTrue, @AssertSame
* Test Suite is a collection of test classes which are meant to run together.
* @Test(timeout = 1000) -> this tells that to test method should complete execution below 1 second, else it is marked as failed.
* Structure of Junits -> Arrange, Act, Assert
* Mocking a static method is possible with the help of Mockito.mockStatic(MyClass.class) method.
* Testing of private methods is not recommended and not possible directly, instead, call the test the method which uses that private method.
* junit concurrency
  + it is possible to run multiple test cases parallelly, to reduce the time.
  + By enabling the config property, junits does the thing for us.
    - junit.jupiter.execution.parallel.enabled = true
* Mockito is a popular java library used in unit testing to create mock objects to simulate the behaviour of complex and large objects.
* @Mock used to create mock objects.
* @InjectMocks -> automatically injects these mocks to the class which we are testing.
* doNothing() method is used to mock the methods of void return type.
  + doNothing().when(mockObject).voidMethod();
* when().thenReturn() is used to mock a method call
  + when(listMock.add(anyString())).thenReturn(false);
* verify()
* when().thenThrow() is used to mock an exception