



# LAB 1 Basic Junit



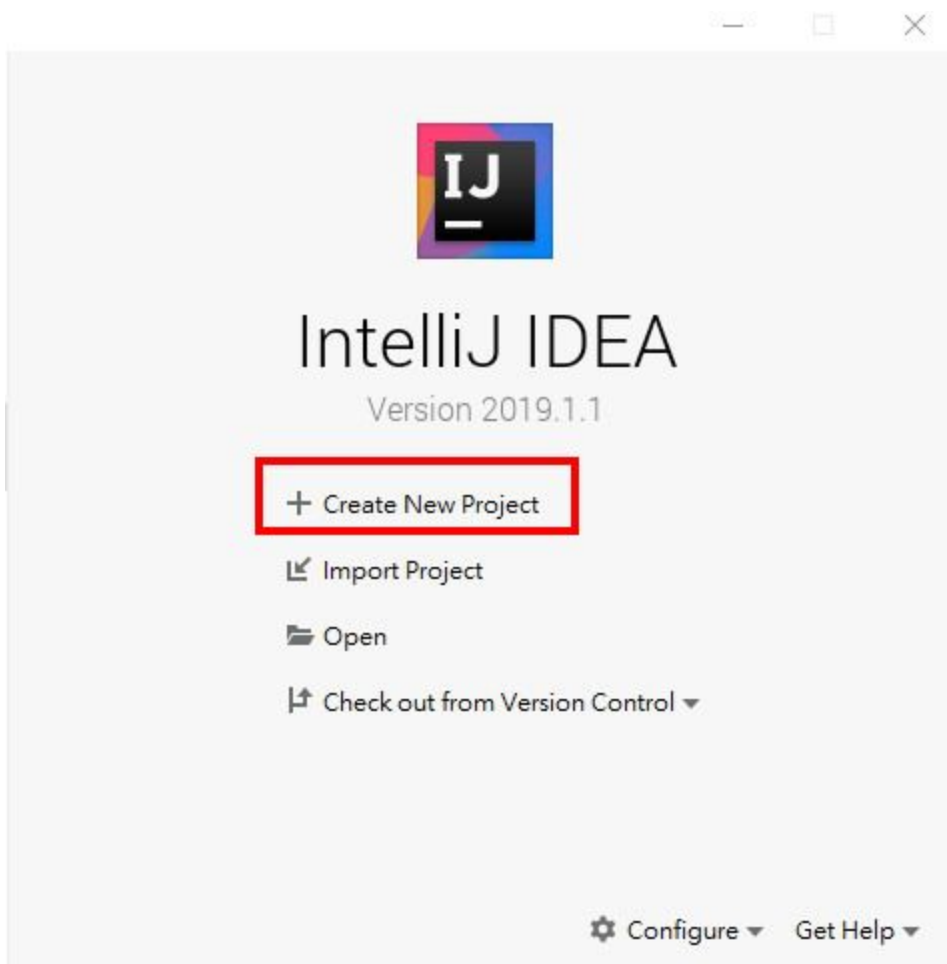
Software Testing 2020

**IntelliJ**

# Install

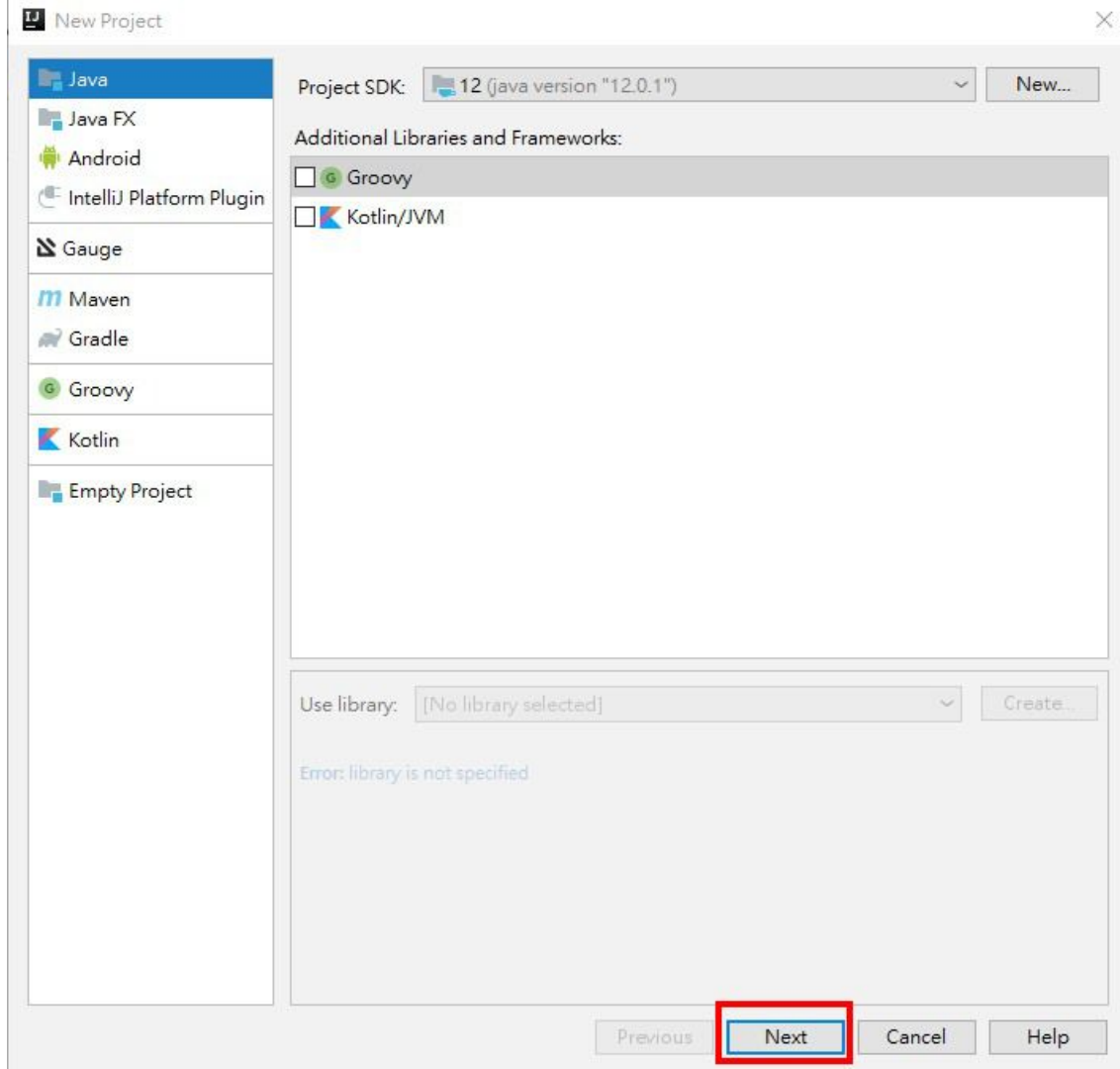
<https://www.jetbrains.com/idea/>

# Guide

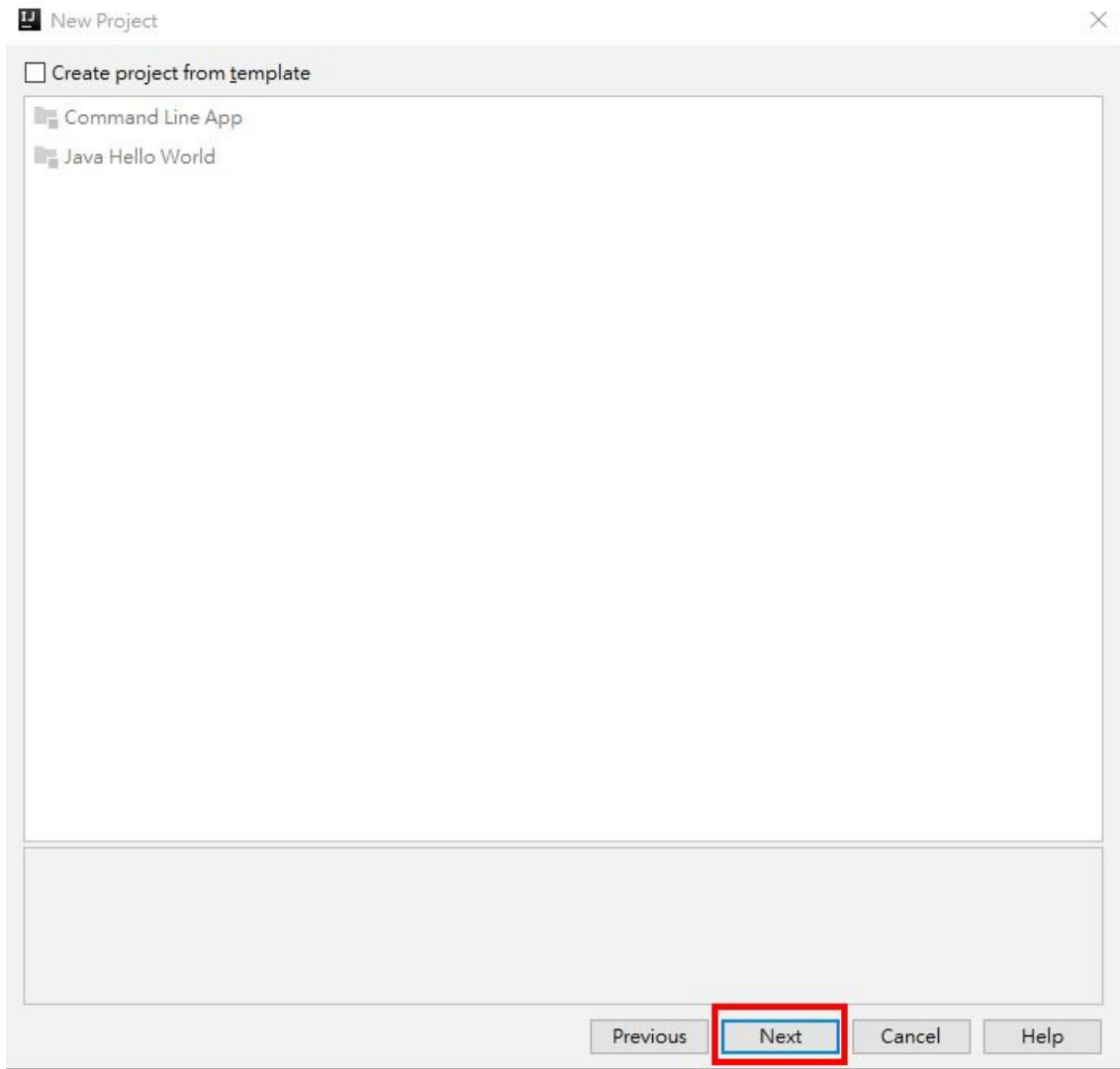


# Guide(cont.)

[OBJ]



# Guide(cont.)



# Guide(cont.)

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

Lab2020 > src

1: Project

Project ▾

▼ Lab2020 C:\Users\User\Desktop\交大\軟體測試\Lab2020

> .idea

▼ src

• Demo

Lab2020.iml

> External Libraries

Scratches and Consoles

右鍵可以新增檔案

• Demo.java ×

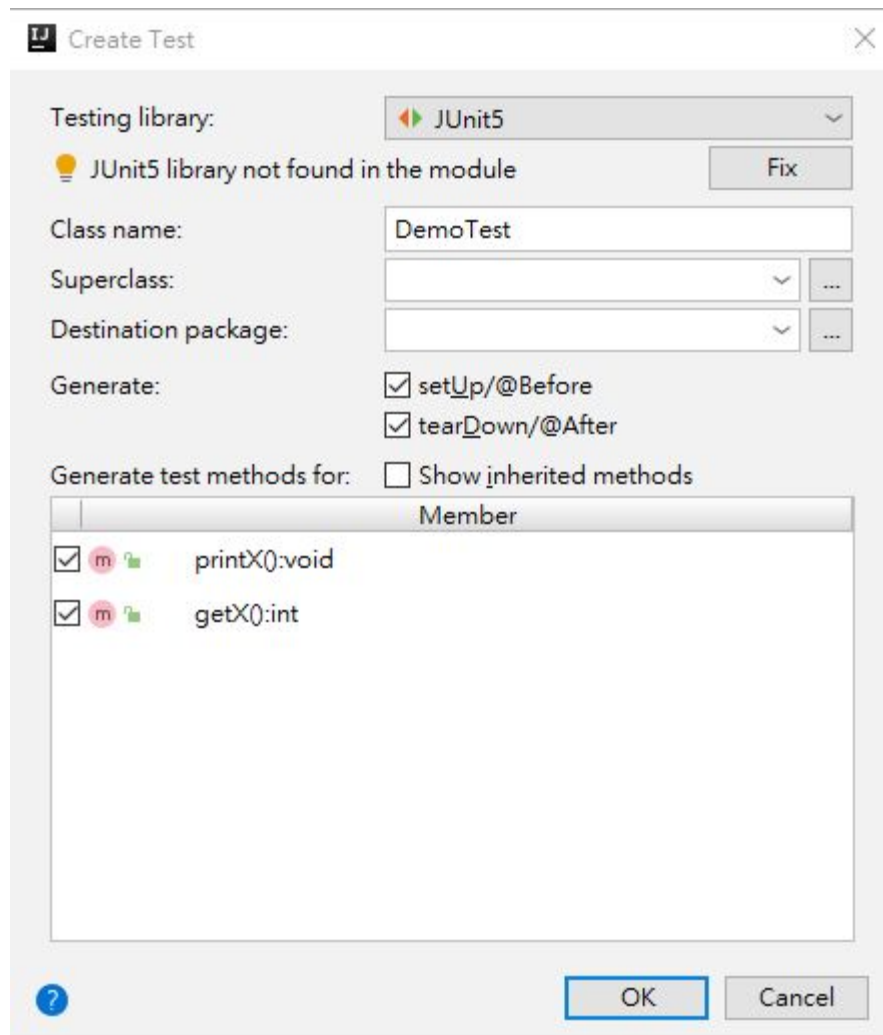
反白後按 alt + enter 點選 create test

```
1 public class Demo {  
2     int x;  
3     @  
4     Demo(){}  
5     Demo(int x){System.out.println(x);}  
6     public void printX(){System.out.println(x);}  
7     public int getX(){return x;}  
8 }
```

**Start Test**



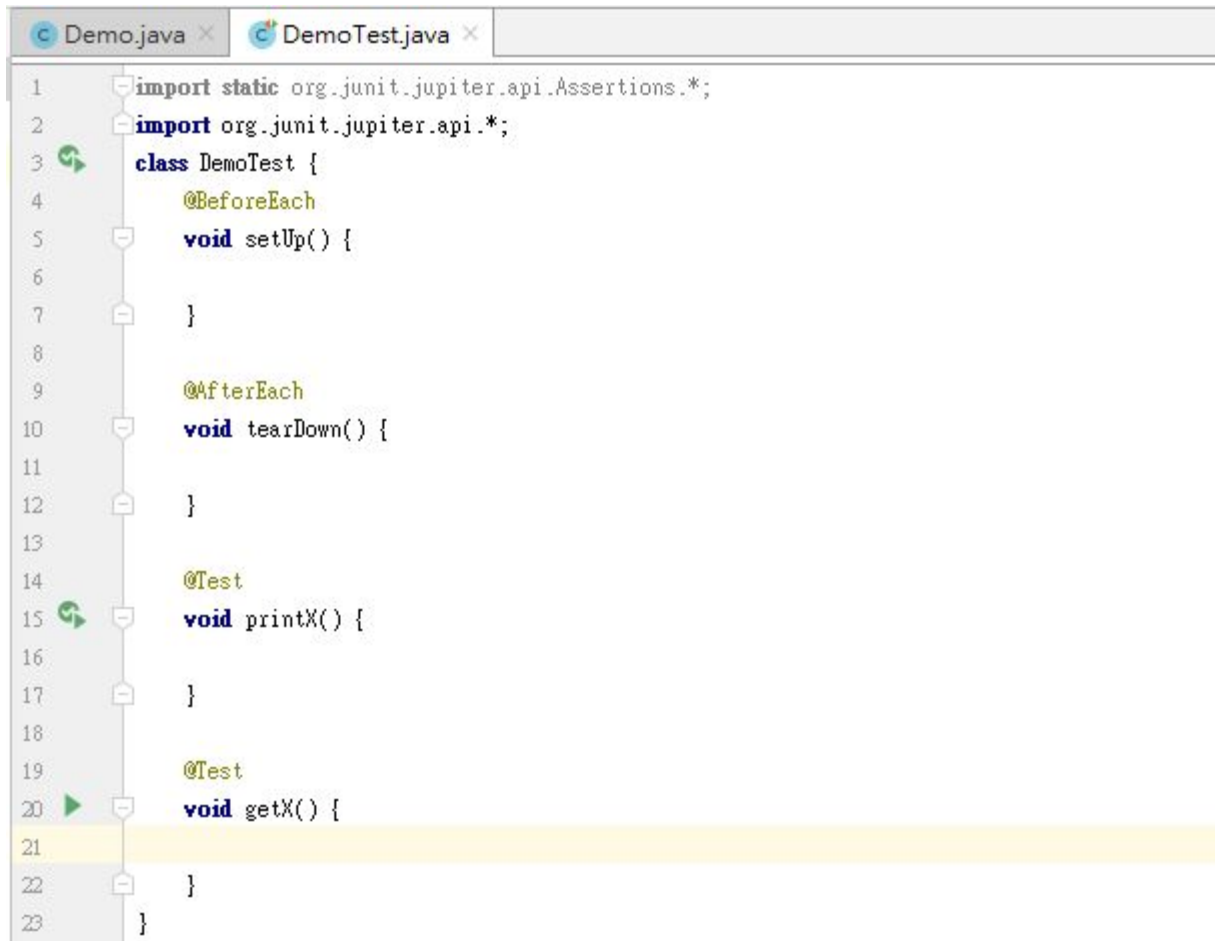
# Guide(cont.)



## Guide(cont.)

If you do not install Junit Libraries, there will pop up a warning.

Just click the warning part with your mouse and press **ALT + ENTER** to install Junit 5.



```
1  import static org.junit.jupiter.api.Assertions.*;
2  import org.junit.jupiter.api.*;
3  class DemoTest {
4      @BeforeEach
5      void setUp() {
6
7      }
8
9      @AfterEach
10     void tearDown() {
11
12     }
13
14     @Test
15     void printX() {
16
17     }
18
19     @Test
20     void getX() {
21
22     }
23 }
```

# Guide(cont.)

Run: DemoTest.printX x

✓ Tests passed: 1 of 1 test – 16 ms

✓ Test Results	16 ms	"C:\Program Files\Java\jdk-12.0.1\bin\java.exe" ...
✓ DemoTest	16 ms	
✓ printX()	16 ms	Process finished with exit code 0

# Coverage

The screenshot displays the IntelliJ IDEA interface with the 'Run' menu open. The 'Run DemoTest with Coverage' option is highlighted. The 'Coverage' window shows the following data:

Element	Class, %	Method, %	Line, %
com			
images			
java			
javax			
jdk			
META-INF			
netbeans			
org			
sun			
toolbarButtonGraphics			
Demo	0% (0/1)	0% (0/4)	0% (0/4)
DemoTest	100% (1/1)	100% (4/4)	100% (5/5)

The 'Test Results' window shows the test passed with the following output:

```
Run: DemoTest
Test Results
  DemoTest
    print() 27ms
    get() 3ms
    Total: 30ms
  
```

The 'Terminal' window shows the following output:

```
IntelliJ IDEA coverage runner ----
sampling ...
include patterns:
exclude patterns:Class transformation time: 0.1807767s for 714 classes or 2.5318865546218495-4s per class
Process finished with exit code 0
```

# Coverage(cont.)

▼ Lab2020 C:\Users\User\Desktop\交大\軟體測試\Lab2020

- > .idea
- ▼ out
  - > production
- ▼ src 50% classes, 55% lines covered
  - Demo 0% methods, 0% lines covered
  - DemoTest 100% methods, 100% lines covered
- Lab2020.iml
- > External Libraries
- Scratches and Consoles

```
1 public class Demo {  
2     int x;  
3     Demo(){}  
4     Demo(int x){System.out.println(x);}  
5     public void printX(){System.out.println(x);}  
6     public int getX(){return x;}  
7 }
```

# **Junit 5 Basic**

# Test Classes and Methods

```
class StandardTests {  
  
    @BeforeAll  
    static void initAll() {  
    }  
  
    @BeforeEach  
    void init() {  
    }  
  
    @Test  
    void succeedingTest() {  
    }  
  
    @Test  
    void failingTest() {  
        fail("a failing test");  
    }  
  
    @Test  
    @Disabled("for demonstration purposes")  
    void skippedTest() {  
        // not executed  
    }  
  
    @Test  
    void abortedTest() {  
        assumeTrue("abc".contains("Z"));  
        fail("test should have been aborted");  
    }  
  
    @AfterEach  
    void tearDown() {  
    }  
  
    @AfterAll  
    static void tearDownAll() {  
    }  
}
```

# Assertions

```
class AssertionsDemo {  
  
    private final Calculator calculator = new Calculator();  
  
    private final Person person = new Person("Jane", "Doe");  
  
    @Test  
    void standardAssertions() {  
        assertEquals(2, calculator.add(1, 1));  
        assertEquals(4, calculator.multiply(2, 2),  
            "The optional failure message is now the last parameter");  
        assertTrue('a' < 'b', () -> "Assertion messages can be lazily evaluated -- "  
            + "to avoid constructing complex messages unnecessarily.");  
    }  
  
    @Test  
    void groupedAssertions() {  
        // In a grouped assertion all assertions are executed, and all  
        // failures will be reported together.  
        assertAll("person",  
            () -> assertEquals("Jane", person.getFirstName()),  
            () -> assertEquals("Doe", person.getLastName())  
        );  
    }  
}
```

<https://junit.org/junit5/docs/current/api/org.junit.jupiter.api/org/junit/jupiter/api/Assertions.html>



# Assertions(Cont.)

```
@Test
void exceptionTesting() {
    Exception exception = assertThrows(ArithmeticException.class, () ->
        calculator.divide(1, 0));
    assertEquals("/ by zero", exception.getMessage());
}

@Test
void timeoutNotExceeded() {
    // The following assertion succeeds.
    assertTimeout(ofMinutes(2), () -> {
        // Perform task that takes less than 2 minutes.
    });
}
```

# Lab 1

# Lab 1

1. Go to [Course Github](#) to download Lab 1 test target “Vehicle.java”
2. Write test of the target for 100% Line cover.
3. You can use any assert function you like and any test logic.
4. In every @test function need to have at least one assertion.
5. Upload your testfile “VehicleTest.java” and the screen shot of test coverage “YOUR\_STUDENT\_NUMBER.jpg” to E3. Don’t need to compress the files.
6. The deadline of every lab is a week after release. Hand in late will get discount in score.
7. Don’t Plagiarism.