Quick Help

1. Creating tests for a class
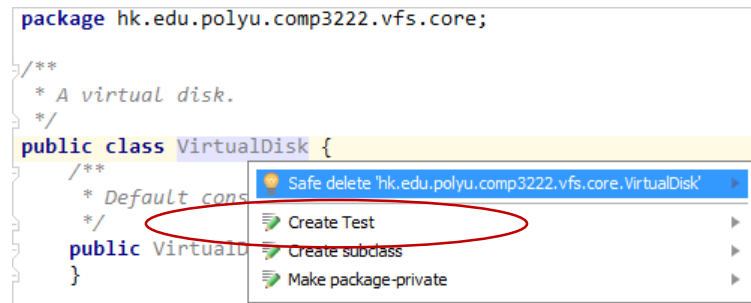
Detailed instructions are available at IntelliJ IDEA Help:
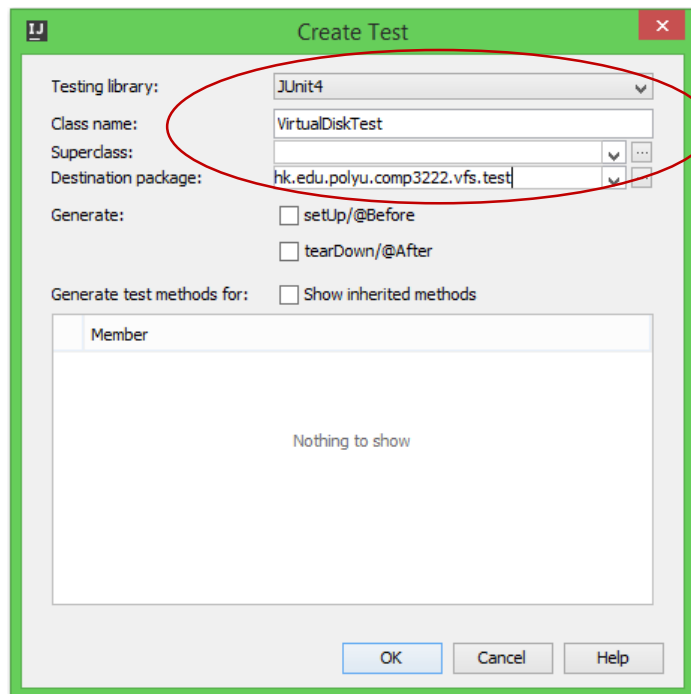
Create a test class

Open the class you want to test in your editor and place your cursor within the line containing the class definition;

Press Alt+Enter to show the list of intention actions, and choose Create Test from the suggestion list

```
package hk.edu.polyu.comp3222.vfs.core;

/**
 * A virtual disk.
 */
public class VirtualDisk {
    /**
     * Default cons
     */
    public VirtualD
    }
```
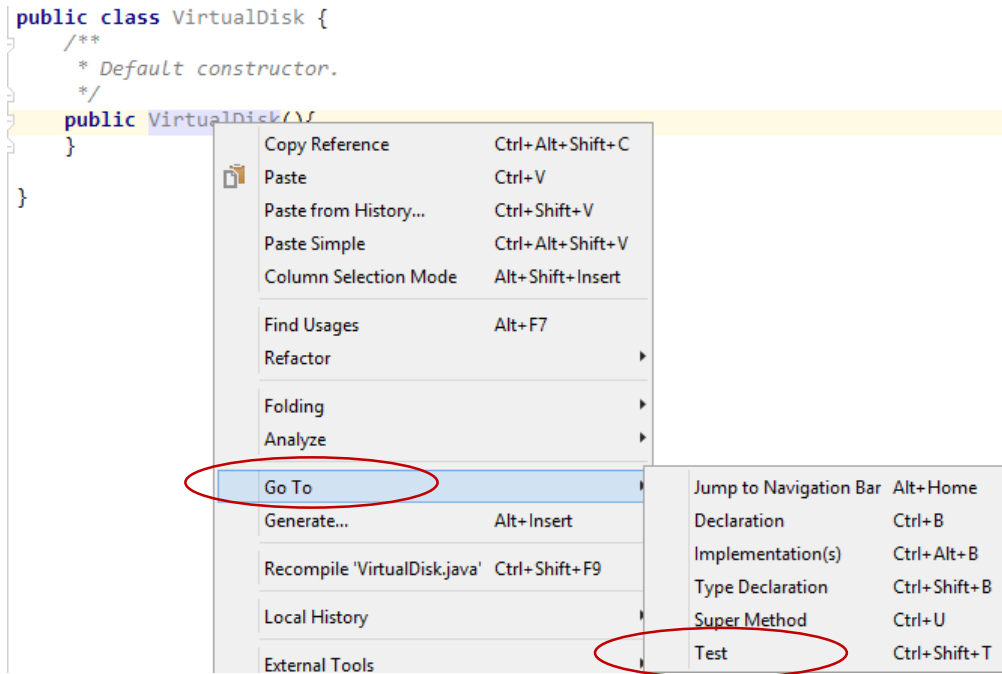
Safe delete 'hk.edu.polyu.comp3222.vfs.core.VirtualDisk'
Create Test
Create subclass
Make package-private

Select JUnit4 as the testing library, name your test class, leave Superclass blank, set the Destination package to be hk.edu.polyu.comp3222.vfs.test, check/uncheck the rest based on your need, and click on OK to create the test.

Create Test

Testing library: JUnit4
Class name: VirtualDiskTest
Superclass:
Destination package: hk.edu.polyu.comp3222.vfs.test

Generate: ☐ setUp/@Before
                        ☐ tearDown/@After

Generate test methods for: ☐ Show inherited methods

Member

Nothing to show

OK   Cancel   Help

Go to the test class created earlier
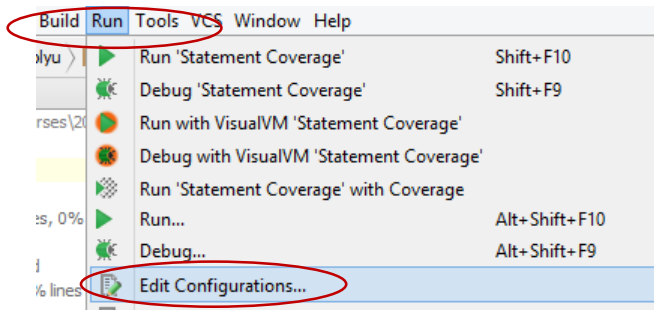
Right click on the class name in the editor to call up the context menu, then choose Go To -> Test to open the existing test class.

```java
public class VirtualDisk {
    /**
     * Default constructor.
     */
    public VirtualDisk(){
    }

}
```

	Copy Reference	Ctrl+Alt+Shift+C
📋	Paste	Ctrl+V
	Paste from History...	Ctrl+Shift+V
	Paste Simple	Ctrl+Alt+Shift+V
	Column Selection Mode	Alt+Shift+Insert
	Find Usages	Alt+F7
	Refactor	▶
	Folding	▶
	Analyze	▶
	Go To	▶
	Generate...	Alt+Insert
	Recompile 'VirtualDisk.java'	Ctrl+Shift+F9
	Local History	▶
	External Tools	

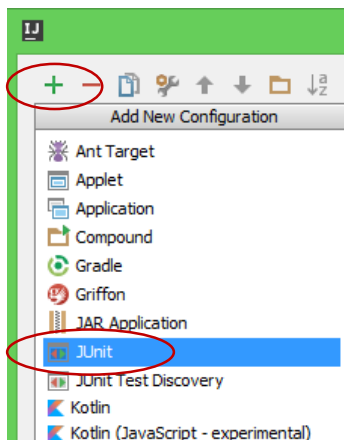| Jump to Navigation Bar	Alt+Home
Declaration	Ctrl+B
Implementation(s)	Ctrl+Alt+B
Type Declaration	Ctrl+Shift+B
Super Method	Ctrl+U
Test	Ctrl+Shift+T

Run all the tests with coverage

[If your project is based on the VFS-template *from the project material bundle, you can simply use the configuration* StatementCoverage *to execute all the tests in package* hk.edu.polyu.comp3222.vfs.test *and show the statement coverage information. Follow the instructions below if you want to add the configuration by yourself.]*
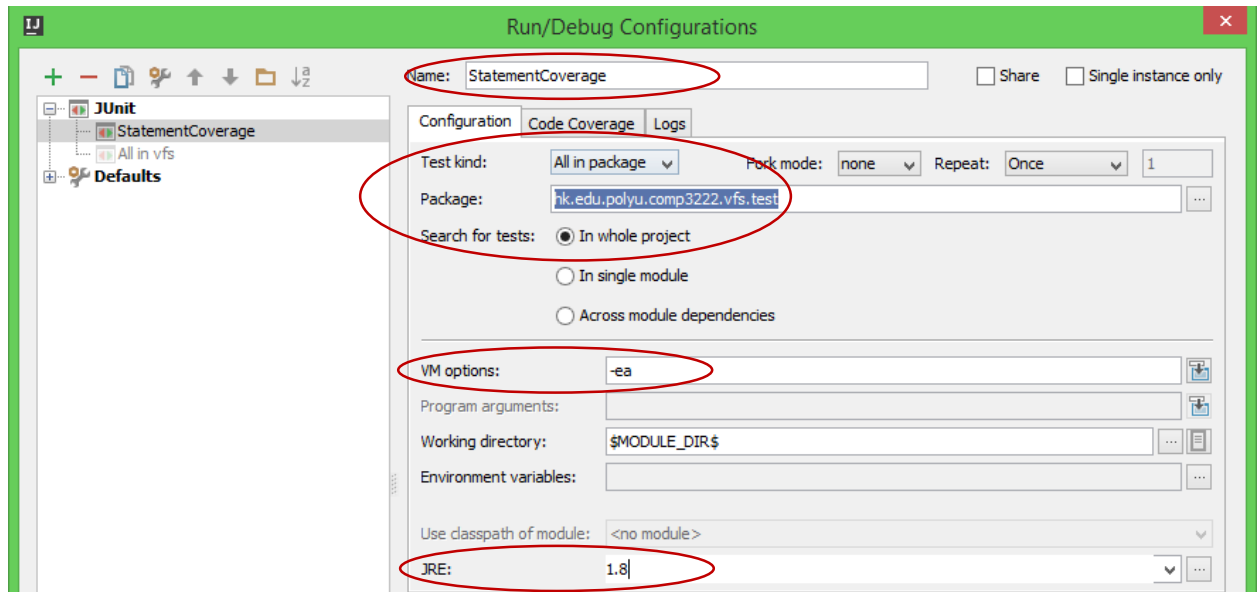
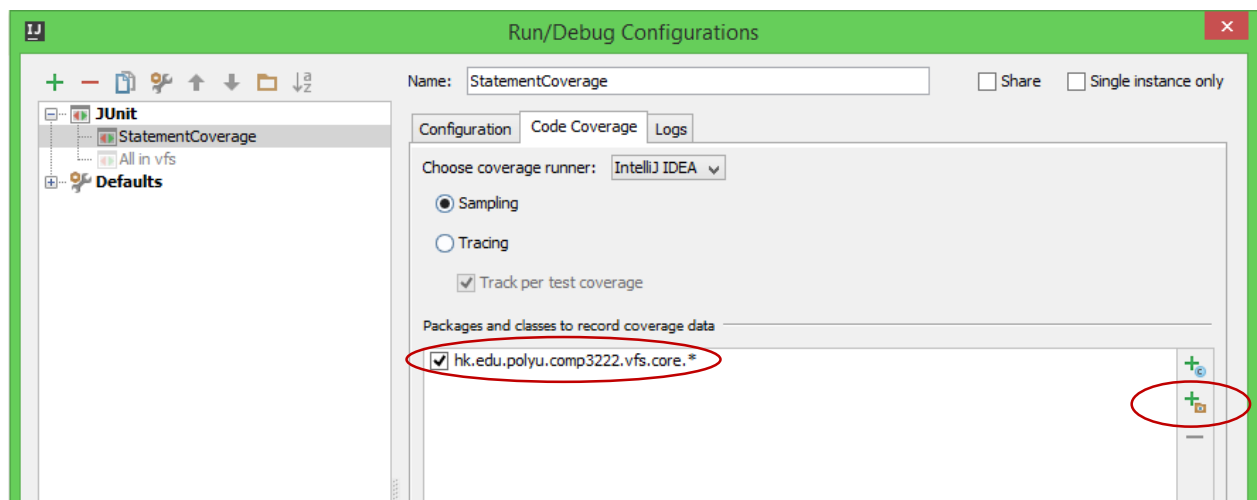Open the Edit Configurations dialog by clicking on Run -> Edit Configurations... .

Build	Run	Tools	VCS	Window	Help
	▶ Run 'Statement Coverage'				Shift+F10
	🐞 Debug 'Statement Coverage'				Shift+F9
	● Run with VisualVM 'Statement Coverage'				
	● Debug with VisualVM 'Statement Coverage'				
	▒ Run 'Statement Coverage' with Coverage				
	▶ Run...				Alt+Shift+F10
	🐞 Debug...				Alt+Shift+F9
	▣ Edit Configurations...				

Click on the **+** symbol on the top-left corner of the dialog to call up the Add New Configuration menu. Choose JUnit from the list.

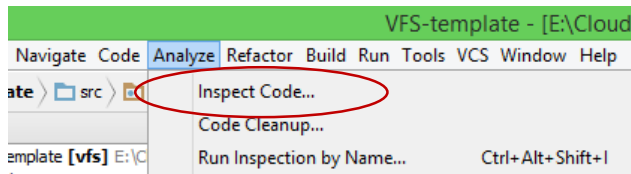| Add New Configuration |
|---|
| ※ Ant Target |
| ▣ Applet |
| ▤ Application |
| ▯ Compound |
| ● Gradle |
| ● Griffon |
| ▮ JAR Application |
| ▦ JUnit |
| ▥ JUnit Test Discovery |
| K Kotlin |
| K Kotlin (JavaScript - experimental) |

Use the settings as shown below for the Configuration tab,

and those shown below for the Code Coverage tab.

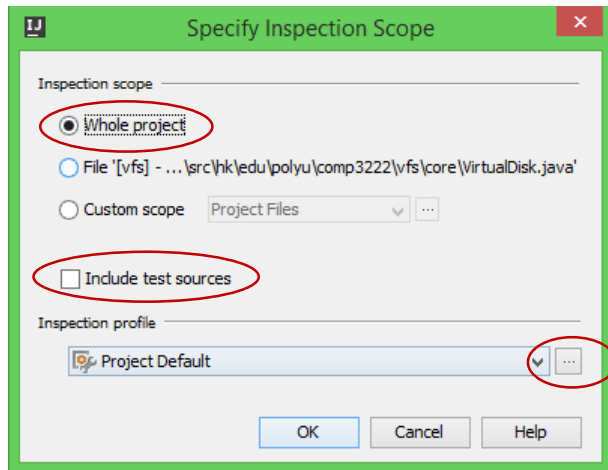
Click on OK to run the configuration or Apply to save it.
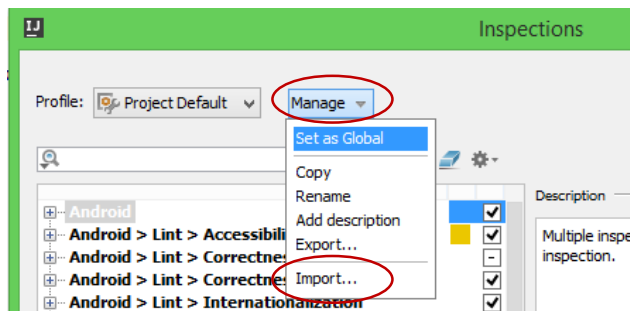
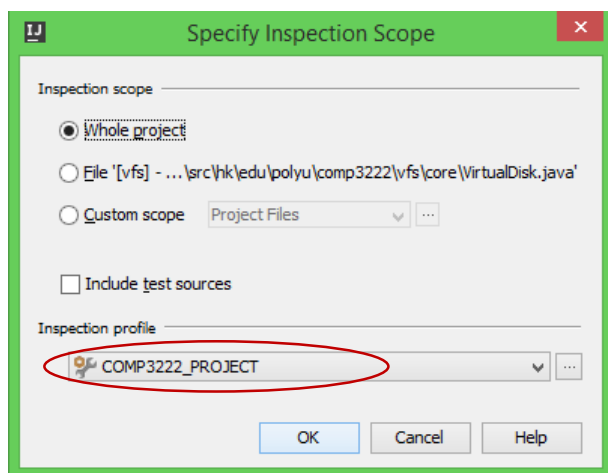## 2. Running the inspection tool

Go to Analyze -> Inspect Code...

In the dialog, select Whole project and uncheck Include test sources. Then click on the ellipses at the bottom-right corner of the dialog.

Click on Manage, then Import...

Import the COMP3222_PROJECT.xml file from the project material bundle, and click on OK to go back to the previous dialog. Select COMP3222_PROJECT as the active Inspection profile.

Click on OK to run the inspection. Inspection results will be shown in the Inspection Results tool panel.