Advanced classes Development environment setup

To make sure we all have the same learning experience, and to accelerate learning with oo concept. We will use Kotlin this is a modern statically typed (compiled) programming language.

Getting started

1. Install the Java JDK version 8.

(if you already have version 8 or 9 installed that is fine, 10 isn't).

- To check if you already have a jdk installed if you want run `java -version` and `javac` from a terminal or command window. Also see https://stackoverflow.com/questions/14292698/how-do-i-check-if-the-java-jdk-is-installed-on-mac
- Download and install the jdk http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html
- or openjdk on a nix machine if you like. http://openjdk.java.net/install/

Install with all default options.

2. Install IntelliJ IDEA *Community* Edition

This is a free IDE with a lot of productivity tools built in. This will be used in the Advanced classes.

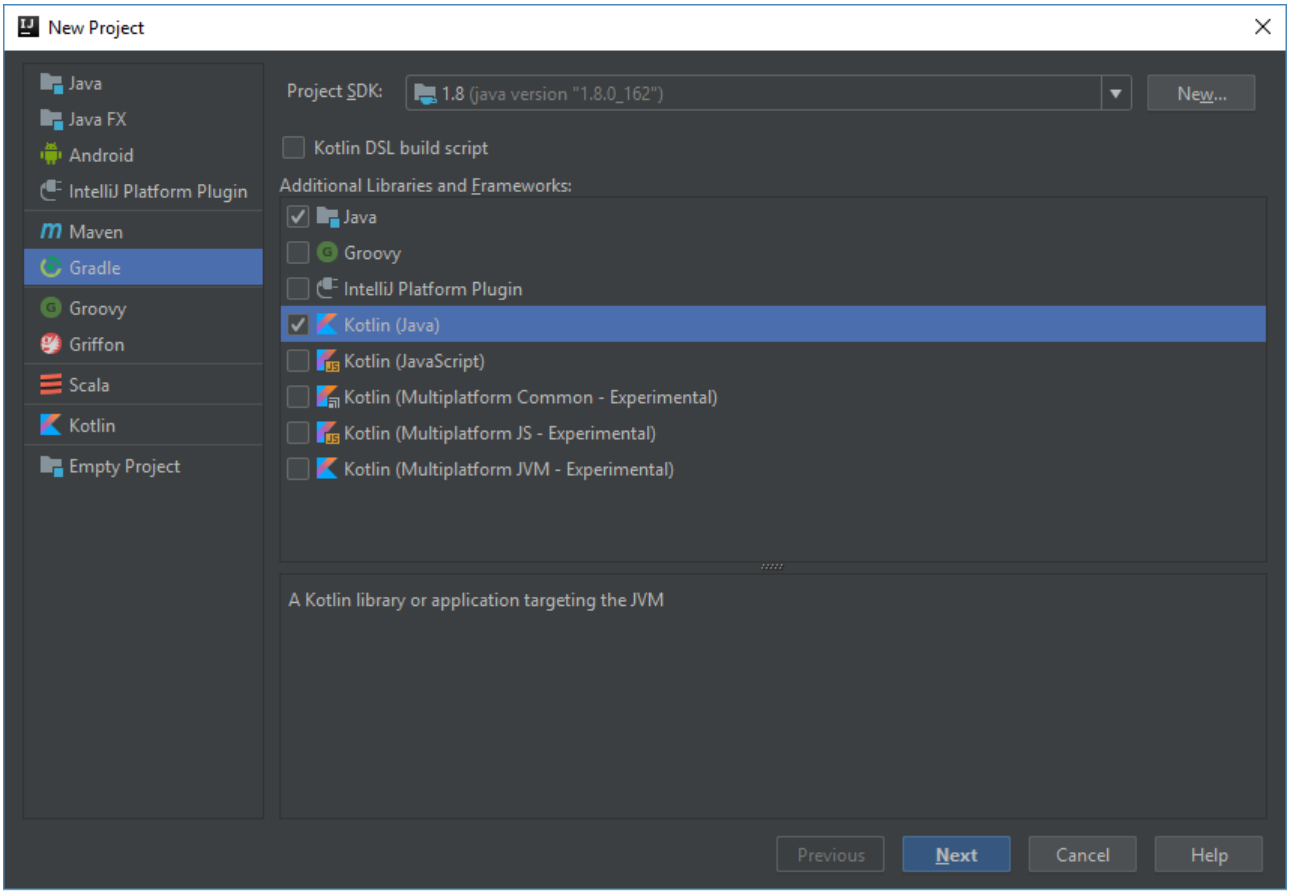
- Download and install the latest version of IntelliJ Community Edition
- During the installation choose the default options.

3. Create a new project and configure the SDK

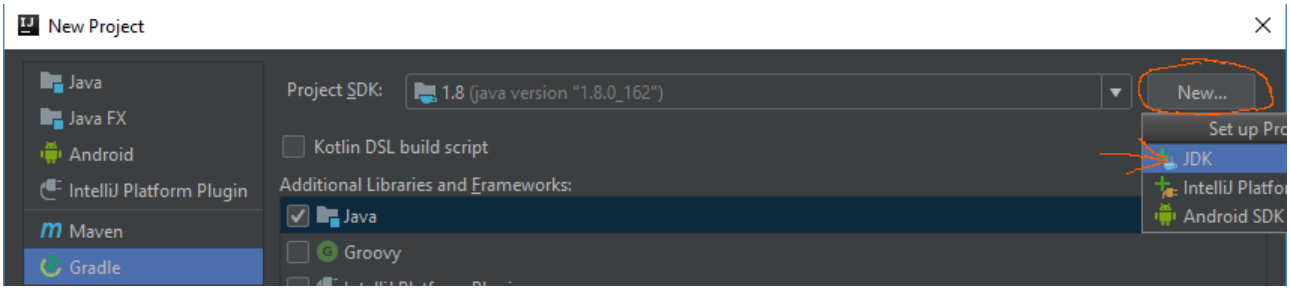
After starting IntelliJ for the first time and you ran through the first time setup wizard. We are going to create a new project.

- **Select: Create new project**

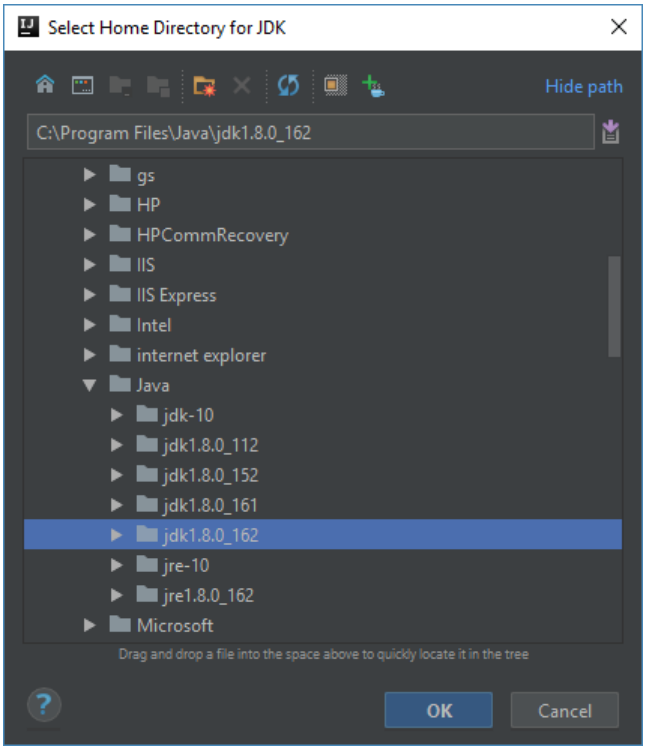
- **Choose Gradle, make sure both Java and Kotlin (Java) are selected.** (Do not press Next yet, in the next step we are going to configure the project SDK.



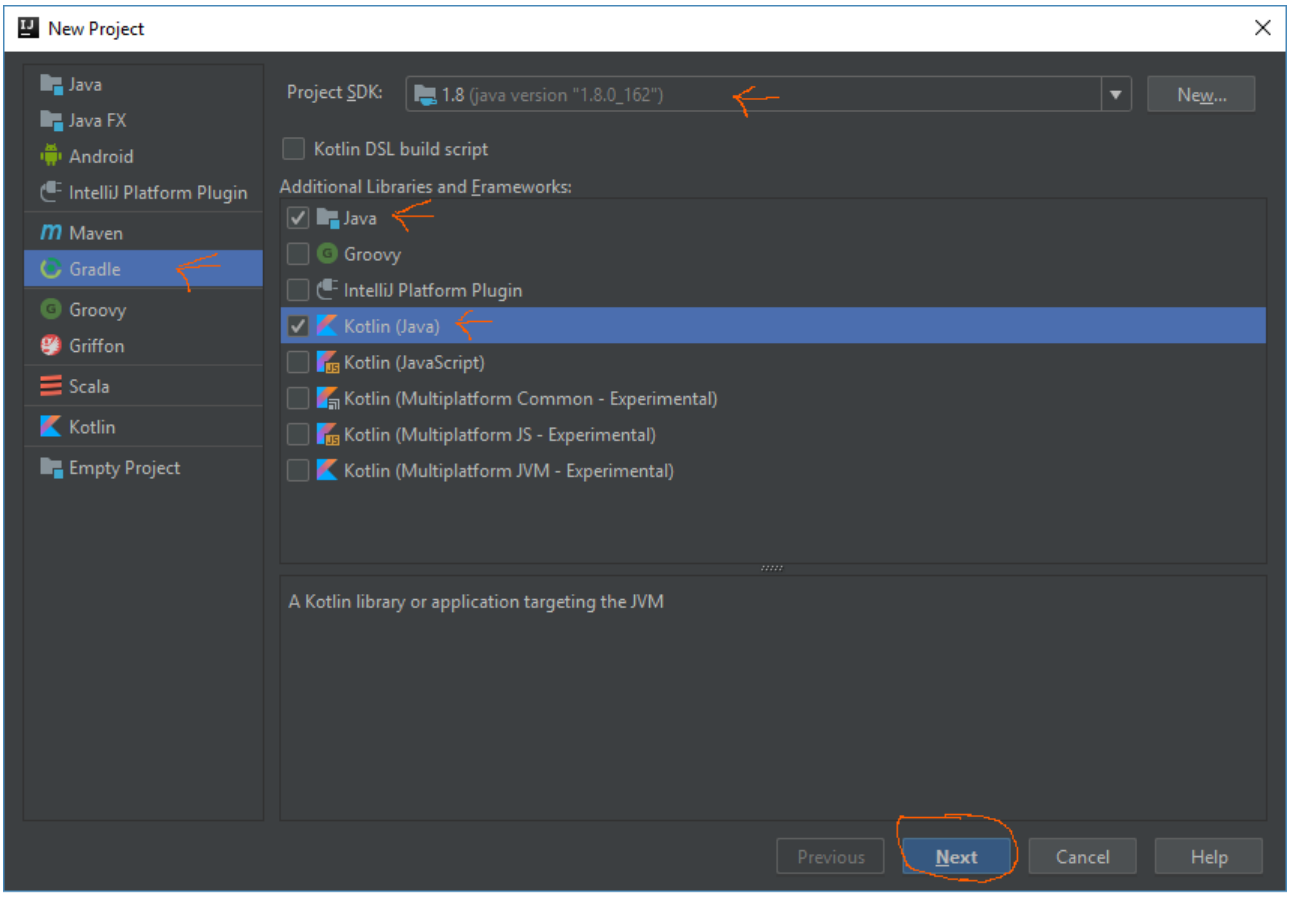
- **If no SDK is selected, press New**



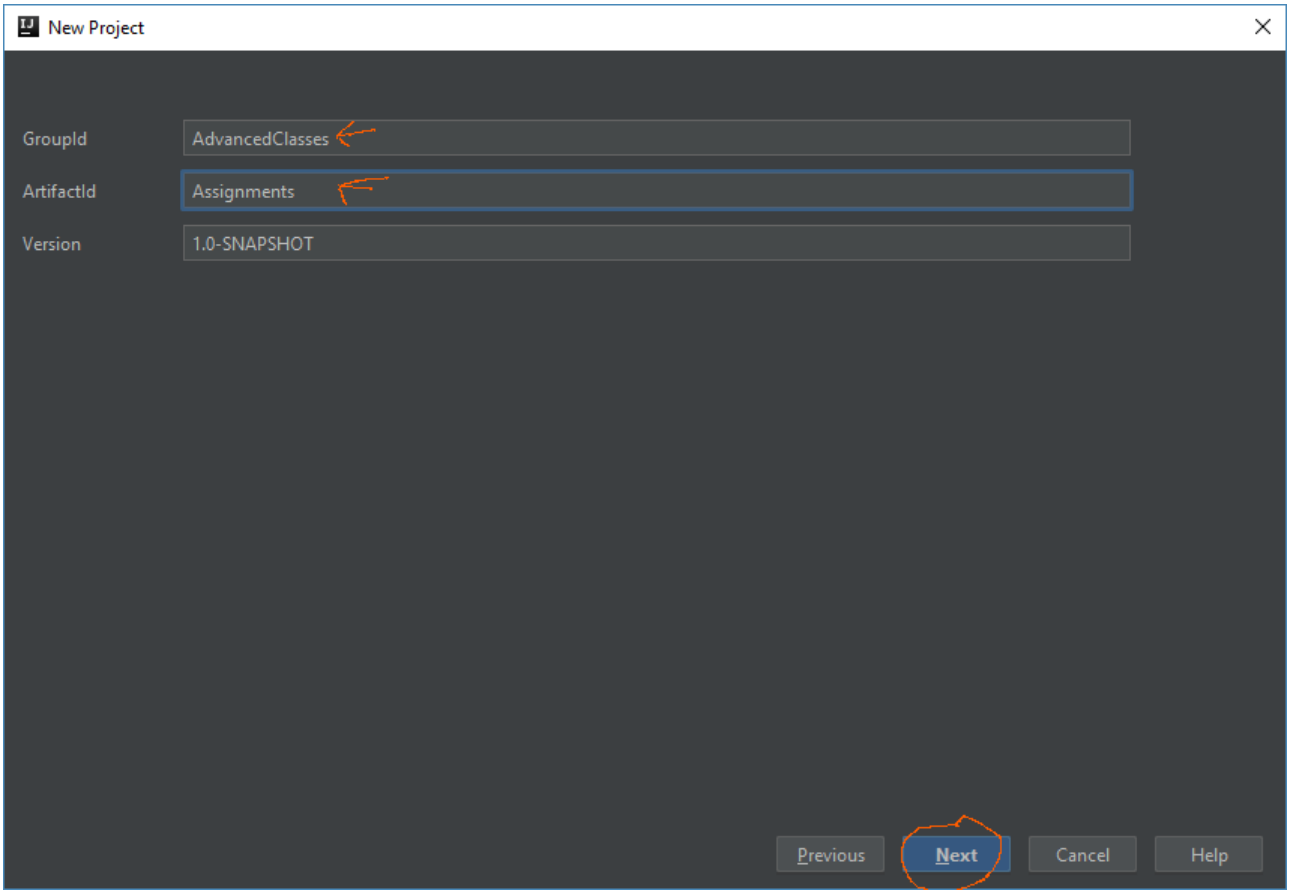
- **Select the path to where you installed the Java SDK. and click OK.**



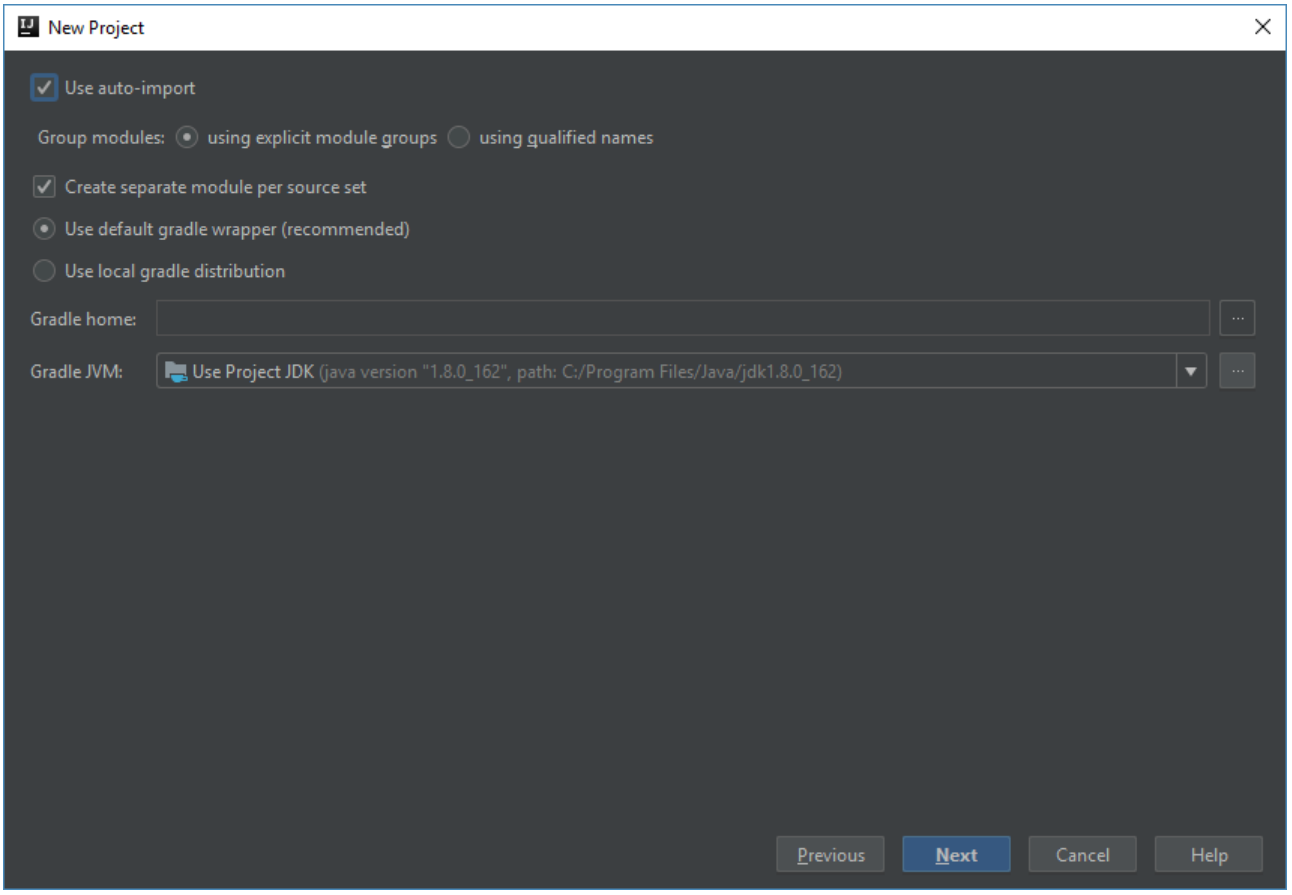
- **Now that we have our project configured. Press Next**



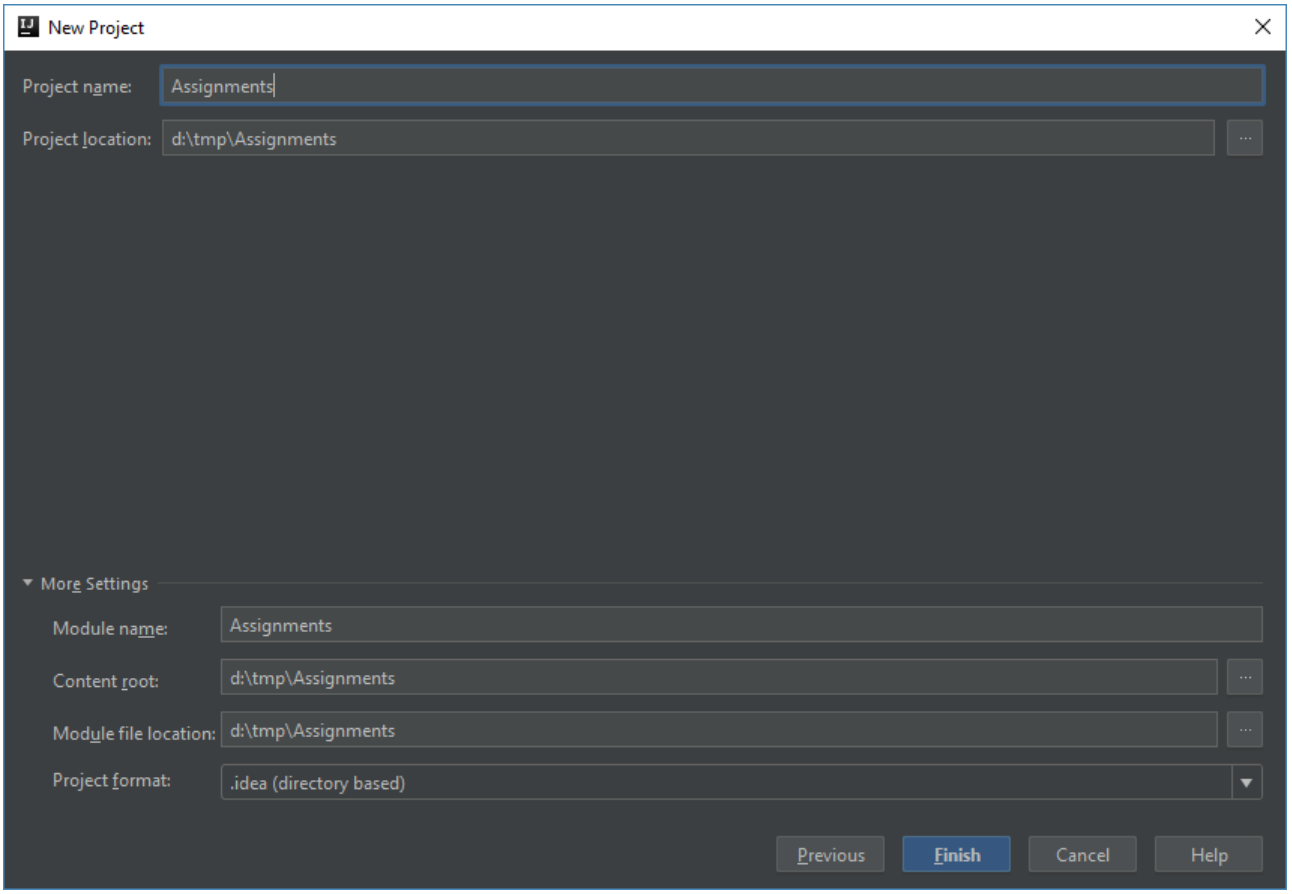
- **Type "Advanced Classes" as the GroupId and "Assignments" as the ArtifactId. And press Next**



- **Select "Use auto-import" and make sure the "Use default gradle wrapper" and the Gradle JVM has "Use Project JDK" is selected. Press Next.**

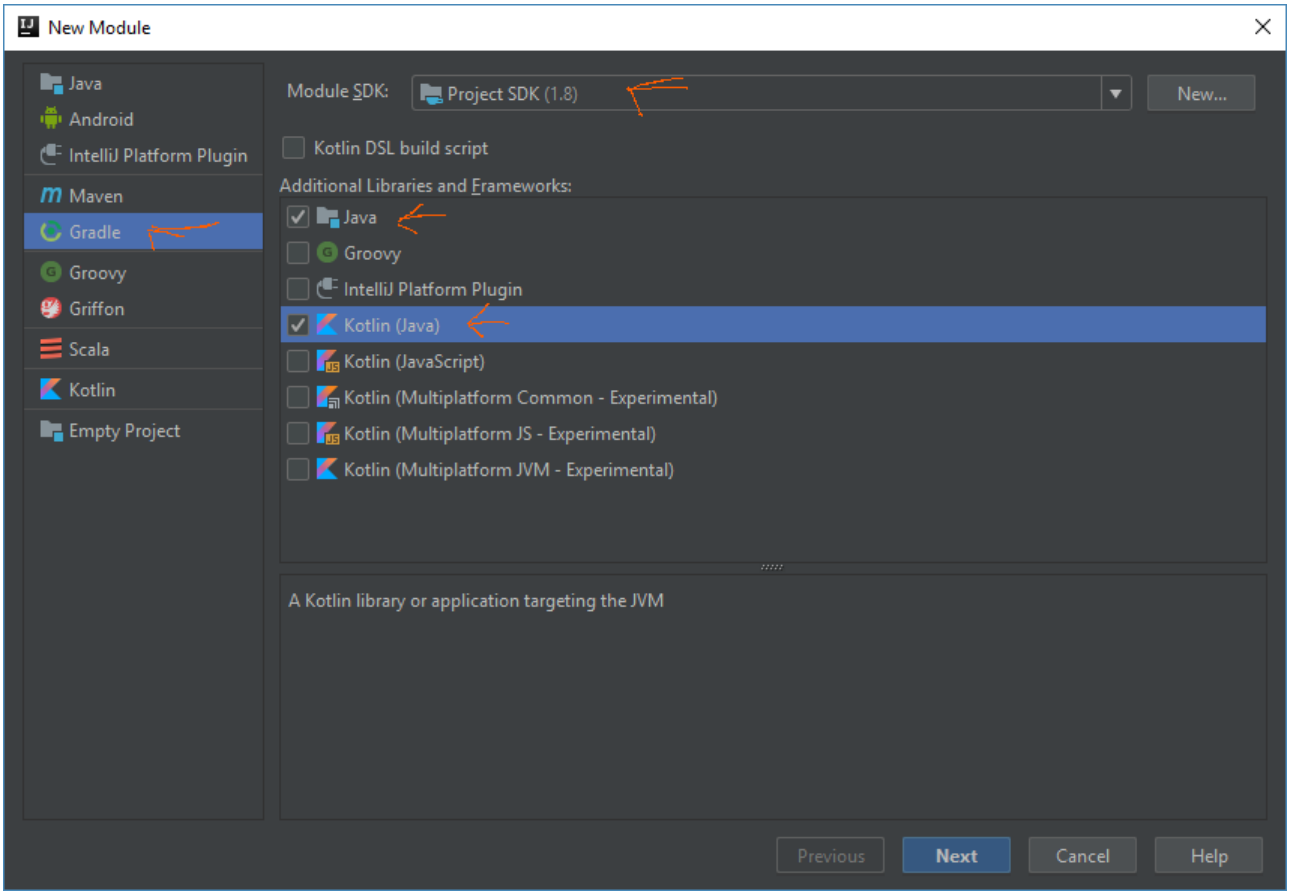


- **Select the path on your machine where you want to store the project. (You can leave the defaults) and click Finish**

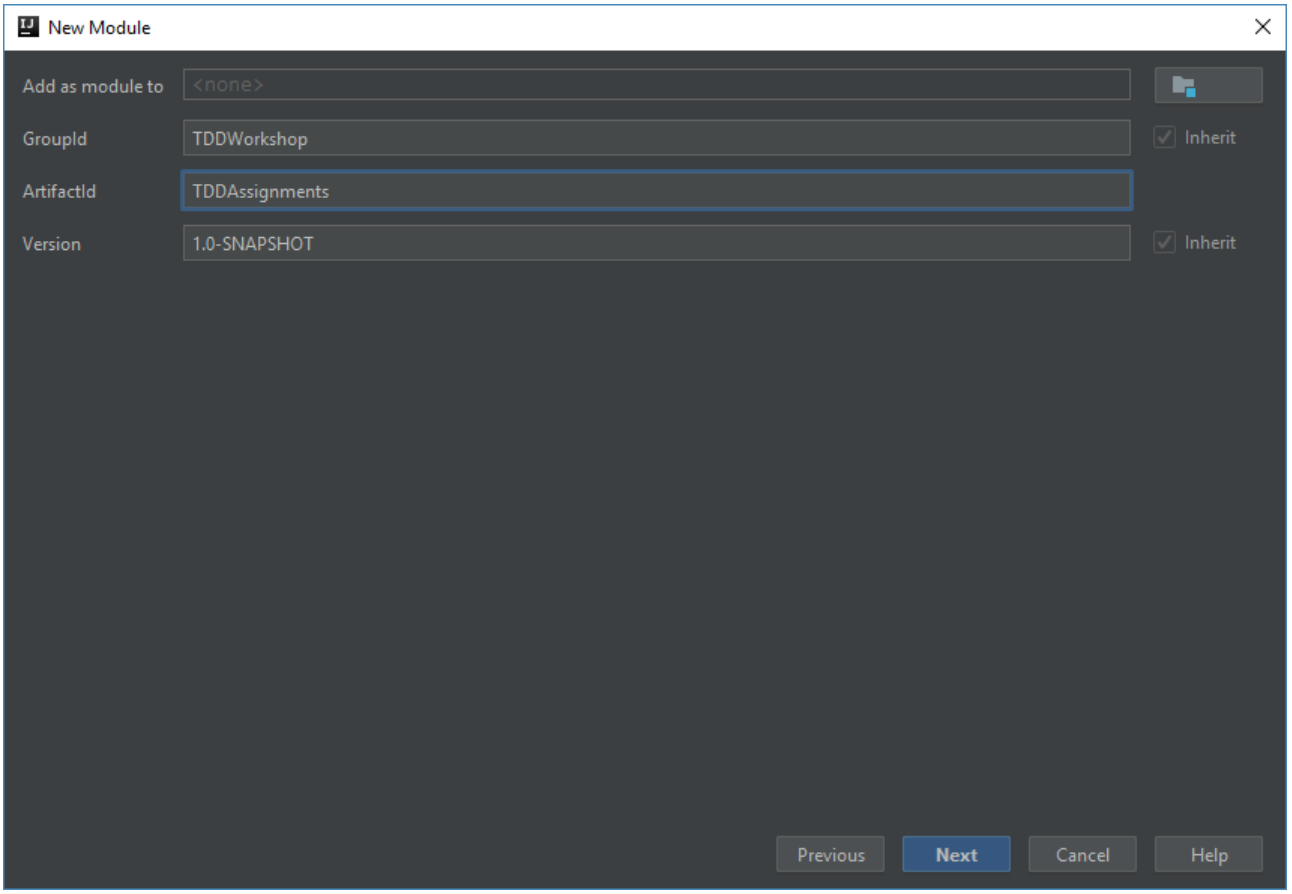


Your basic project is setup. Now we are going to add a new module.

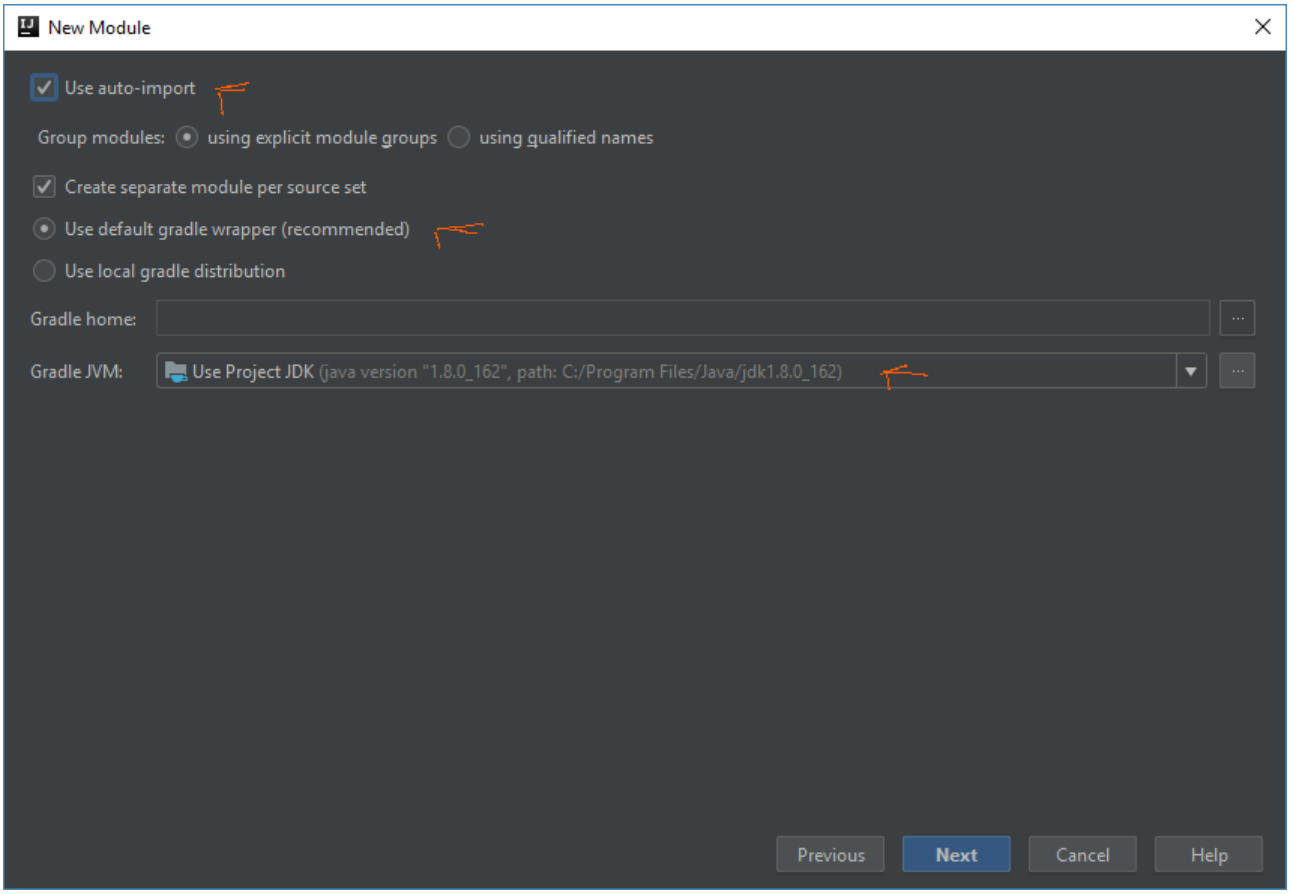
- **Select File->New->Module**
- **Make sure only the Java and Kotlin(Java) are checked. And that the Module SDK has the 'Project SDK' selected. Press Next**



- **Choose a GroupId: "TDDWorkshop" and the ArtifactId: "TDDAssignments". Click Next**



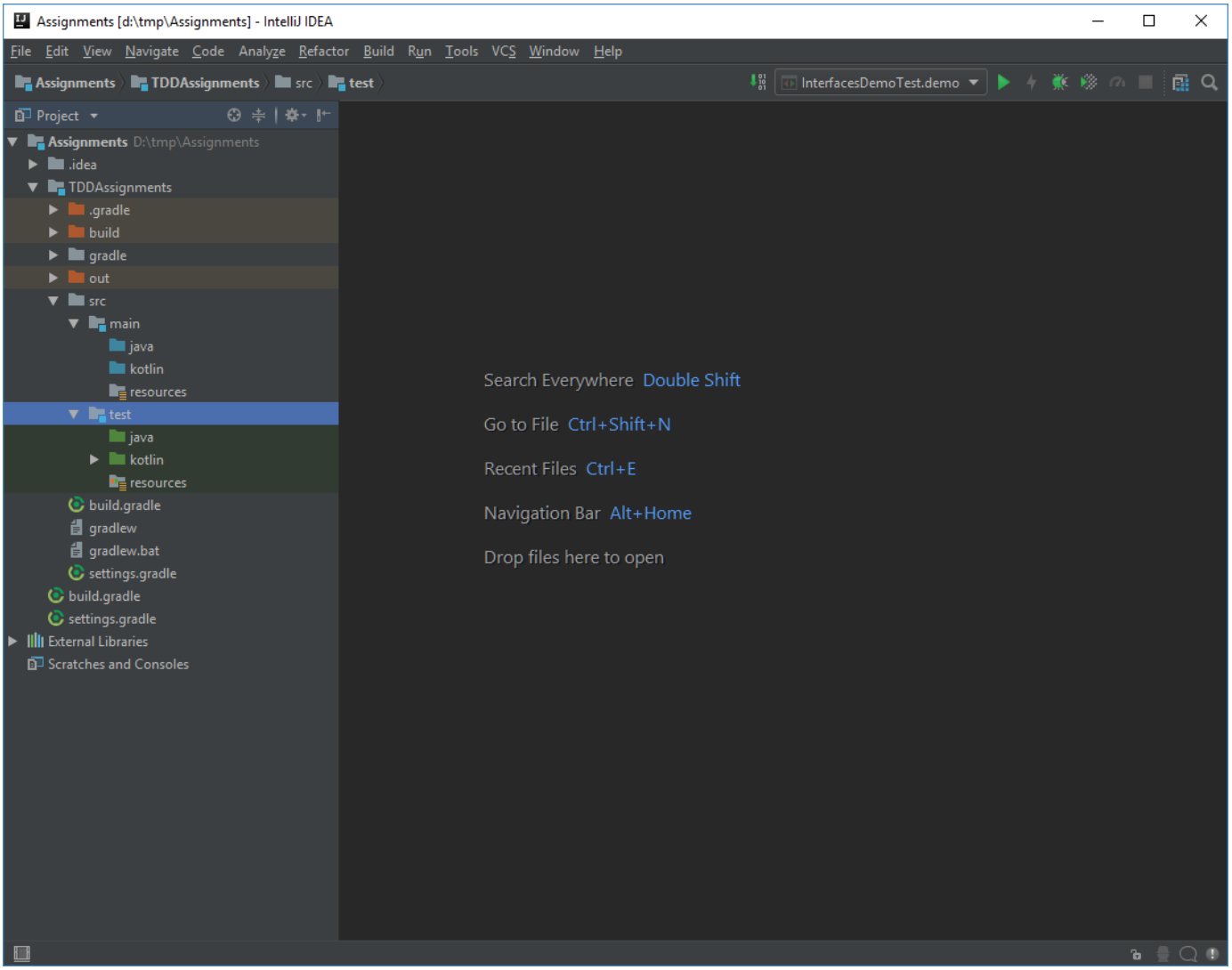
-Select "Use auto-import" and make sure the "Use default gradle wrapper" and the Gradle JVM has "Use Project JDK" is selected. Press Next.



- **Select the path on your machine where you want to store the module. (You can leave the defaults) and click Finish**

IntelliJ will automatically build the initial project and module setup and tries to retrieve packages. It may be that your firewall will popup and ask permission for Java to connect to internet. Please allow this.

Now we have the following structure.



Next we need to add some dependencies (packages) to the module.

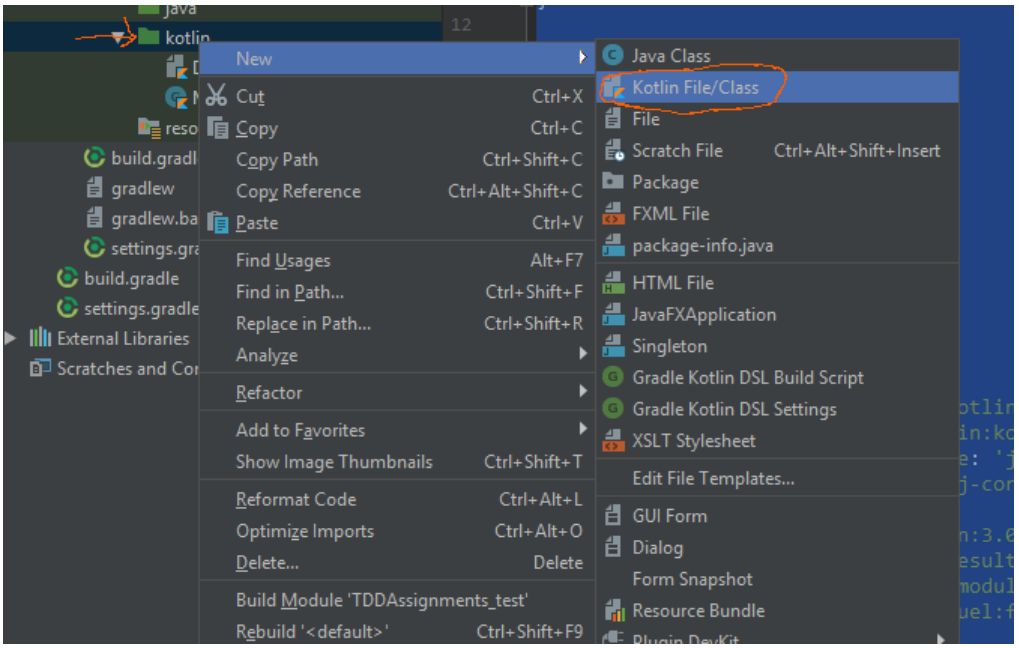
Open the file TDDAssignments/src/build.gradle by double clicking on the file

Make sure it looks like this. and save the file.

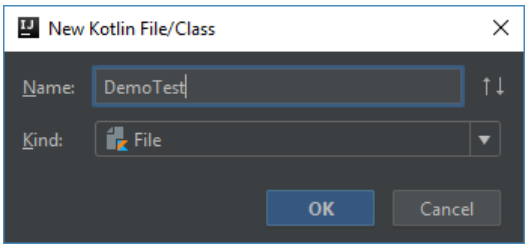
```
buildscript {
    ext.kotlin_version = '1.2.31'

    repositories {
        mavenCentral()

    }
    dependencies {
        classpath "org.jetbrains.kotlin:kotlin-gradle-plugin:$kotlin_version"
    }
}

group 'TDDWorkshop'
version '1.0-SNAPSHOT'

apply plugin: 'java'
apply plugin: 'kotlin'

sourceCompatibility = 1.8

repositories {
    mavenCentral()
    jcenter()
}
```

```
dependencies {
    compile "org.jetbrains.kotlin:kotlin-stdlib-jdk8:$kotlin_version"
    testCompile "org.jetbrains.kotlin:kotlin-test-junit:$kotlin_version"
    testCompile group: 'junit', name: 'junit', version: '4.12'
    testCompile 'org.assertj:assertj-core:3.9.0'

    implementation 'com.beust:klaxon:3.0.1' //native kotlin json ser / des lib
    compile 'com.github.kittinunf.result:result:1.3.0'
    compile 'com.fasterxml.jackson.module:jackson-module-kotlin:2.9.+' //for Jackson support (json)
    compile 'com.github.kittinunf.fuel:fuel:1.12.1' //for JVM

}

compileKotlin {
    kotlinOptions.jvmTarget = "1.8"
}
compileTestKotlin {
    kotlinOptions.jvmTarget = "1.8"
}
```

Right-click on the folder TDDAssignments/src/test/kotlin. Select New->"Kotlin File/Class"



Name it "DemoTest" and click ok



Make sure the file contains the following: import org.assertj.core.api.Assertions import kotlin.test.Test

```
class DemoTest {

    @Test
    fun assertionSimpleDemo() {

        val myNumber = 4
        Assertions.assertThat(myNumber).isGreaterThan(2).isLessThan(5)

        val myStringList = listOf("A", "B", "C")
        Assertions.assertThat(myStringList).hasSize(3)
        Assertions.assertThat(myStringList).contains("A", "B")
```

```
    }
}
```

Run the test! (ctrl+shift+f10) or click the run test icon