Warm-up – Installing and Creating Projects using different IDEs and setting up an environment for assignments

Students say before this lab: I know how to do it! It will be a waste of time.

Lecturer/Tutors say: You must go, it is important to set up your environment correctly for exams. It will help you to focus on the exam and not spend time.

Students during exams: The project I created is not working. I need more time!

Avoid this situation and attend this lab.

Lab

Task for this week – part 1

- This lab has no assessable items.
- In this lab you will learn how to set up an environment for assignments.

- Task 1
  - Install JDK on your personal computer
- Task 2
  - Set-up an environment for individual assignments

Task 1 - Install Java Development Kit (JDK)

To develop Java programs, you have to install JDK first. JDK is the development package for Java development including compiler, analysis tools, Java Runtime Environment (JRE), etc.

- Download Oracle's JDK (available at https://www.oracle.com/java/technologies/javase-downloads.html). You can use either JDK 8+ or 11 LTS for the course of this semester. The installation is straightforward.

- Windows:
  - Download and launch the executable file, follow the instructions of the wizard and the installation should be completed.

- Mac OS:
  - Download and Launch the macOS Installer (.dmg file) and then there should be a .pkg file inside. Launch the .pkg file and follow the instruction and the installation should be completed.

Task 1 - Install Java Development Kit (JDK)

- Linux:
  - Use your package manager to simplify your installation.
  - For example, in Ubuntu, you can install JDK 11 by:

```
apt install openjdk-11-jdk
```

and JRE by:

```
apt install openjdk-11-jre
```

- You may need to do `apt update` for updating your package list.

Task 1 – Let’s test it!

- Create a simple HelloWorld Java Program!
  - https://www.geeksforgeeks.org/beginning-java-programming-with-hello-world-example/

- If you were able to compile the HelloWorld example described in the geeksforgeeks website, the installation is successful.

- Let’s move to the next task!

Task 2 – Set-up an environment for assignments

- To successfully complete the assignments, you first need to set up a Java environment in your personal computer (Task 1).

- There are two well-known advanced development environment that you may want to use in this course:
  - Eclipse (https://www.eclipse.org)
  - IntelliJ IDEA (https://www.jetbrains.com/idea/)

  - Go to the official site to get an up-to-date instruction for the installation (instructions may very depending on your OS).

Task 2 – Set-up an environment for assignments

- The key to successful development environment of the assignments is to know how to include libraries (e.g., JUnit4) into the Java build path. **Junit4** will be used throughout the entire course (take the time to learn how to import it successfully).

- We will provide any necessary JAR files in the assignment repository (if you need to add a library, we will explicitly mention it).

- Adding JAR files into the build path depends on the application and system.

- The next slides list the basic steps used to set up representative applications: Terminal, Eclipse and IntelliJ.

- For the next examples, use the code “Concatenate.java” and “ConcatenateTest.java” available on the repository for this course (see Wattle).

Task 2 – Set-up an environment for the individual assignment

- Using Terminal (check manual or ask tutors if it doesn’t work on your computer):

- Compile files using the following command:

```
(Linux/MAC) javac -cp .:"path/to/junit/jars/*" ConcatenateTest.java
(WINDOWS) javac -cp .";path/to/junit/jars/*" ConcatenateTest.java
```

- If you running the compiled class file with the following command, it will automatically execute the test cases in `SomeTestClass`.

```
(Linux/MAC) java -cp .:"path/to/junit/jars/*" org.junit.runner.JUnitCore ConcatenateTest
(WINDOWS) java -cp .";path/to/junit/jars/*" org.junit.runner.JUnitCore ConcatenateTest
```

(do not forget to replace with the directory containing the JUnit jar files)

Note that we use javac to compile and after java to run the program/tests!

Task 2 – Set-up an environment for the individual assignment

- Using Eclipse:

- There are multiple ways to do it, but if you were given the source code files, you can follow the steps below:

1) Open Eclipse. Go to File → New → Java Project
2) Uncheck user default location
3) Click Browse... and find a root folder which contains all source files, then Click Open.
4) Click Next

If you can see all files under the correct package (often default package), it's all done.

5) Click Finish

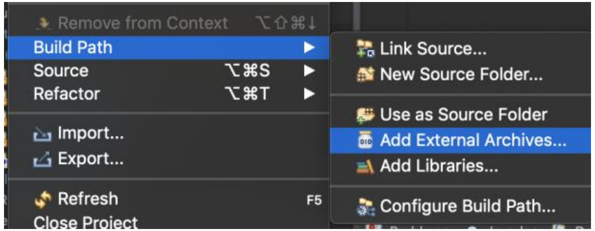
*Note that for all tasks/exams we will assume default package (unless explicitly specified).

Task 2 – Set-up an environment for the individual assignment

- If your project is already created on Eclipse, it is time to add the required jars for the individual assignment:

1) Right Click the project and go to:

Build Path → Add External Archives… → Select the jars using dialog box



2) Once you successfully import these libraries, you will not see any error messages from ConcatenateTest.java file.

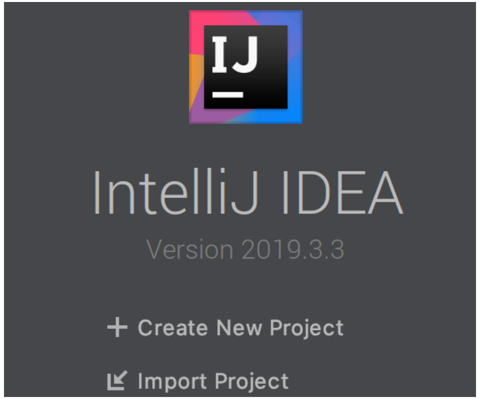
3) Run the java application to check whether the library has been linked properly

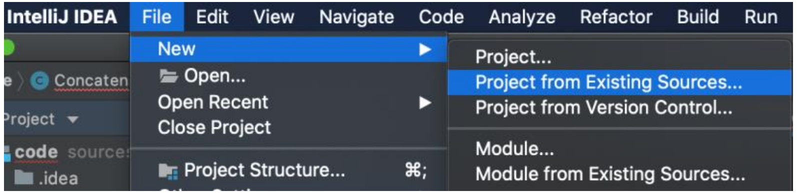
Task 2 – Set-up an environment for the individual assignment

- Using IntelliJ IDEA:
- If you were given the source code files, follow the steps below:

  1a) Open IntelliJ. From Welcome Panel:
  Import Project

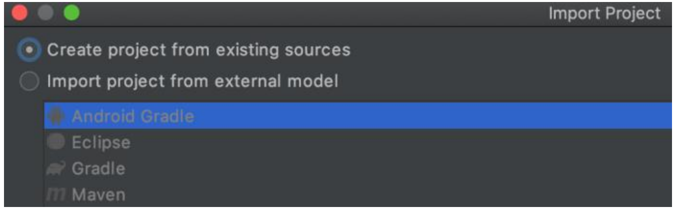
  1b) Open IntelliJ. From Project Panel:
  Go to File → New → Project from Existing Sources...





Task 2 – Set-up an environment for the individual assignment

2) Find a root folder which contains all source files and Click Open
3) Select Create project from existing sources and then Click Next



Import Project

● Create project from existing sources

○ Import project from external model

4) Configure project name and Click Next.
5) IntelliJ will find source files automatically. Click Next.
6) Review libraries automatically found. Click Next.
7) Review Modules. Click Next.
8) Set SDK version. Click Next.
9) Finish

*Note that for all tasks/exams we will assume default package (unless explicitly specified).

Task 2 – Set-up an environment for the individual assignment

- If your project is already created on IntelliJ, it is time to add the required jars for the individual assignment:

1) File → Project Structure → Modules → select Dependencies → Click (+) button → select JARs or directories → add jars from the dialog box.

2) Once you successfully import these libraries, you will not see any error messages from ConcatenateTest.java file.

3) Run the java application to check whether the library has been linked properly

Attention!

- Do not add any package name in your code of assignments, test, mid-term and exam!
- All the given code in this course does not contain any package name
- You should not add any package name in the code when you submit your answers
- Otherwise, your code will be incompatible with our marking program
- You will be receive severe penalty for any violation
  - There will be 50% mark reduction
- Sometimes, your IDE (IntelliJ) may add a package name without your awareness
- Always check your code carefully before submission

Part 1 is finished!

Did anything go wrong? Let your tutor know and s/he will help you to set up the environment!

Note that how to set up a working environment properly including library (jar files) configuration is extremely important **since** you have to do it by yourself during the online test and final exam!