SYSC 4806 – Lab 1: Using the Maven Build Tool

We're going to use the IntelliJ IDE this term, like we did in SYSC3110. But this term we're going to use the Ultimate Edition, which is still free for students. So your first order of business is getting Ultimate Edition (we won't need Ultimate's features for this lab, so if you already have the Community edition you can still complete this lab, but make sure you get Ultimate Edition for future labs).

Our goal in this lab is to create a simple AddressBook application, but this time using Maven. There's this 5 minute intro to Maven on the Maven web site you may want to check out first.

1. Find and Open IntelliJ. Click "create new project"
2. In the window that pops up, specifically make sure:
   - your project location is where you can re-use your work in future labs;
   - your build system is Maven;
   - and the JDK is at least 14.

   In the Advanced Settings at the bottom of that window, you can set the GroupId and the ArtifactId if you wish (those are parameters of Maven that were also mentioned in the 5-minute intro to Maven linked above, and which you can always change later in the pom.xml). Click "Create".
3. On the left pane, you can now see the folders and files that were created for you: among others a src/main folder, and a corresponding src/test folder. You should also see the generated pom.xml file. Select that file to see its content and verify that it does contain the GroupId and ArtifactId you provided.
4. Now let's see what we can already do with our Maven project. In the View menu, select "Tool Windows -> Maven ". In the new pane, after

clicking on the small arrow next to your project name, you can see all the phases of the lifecycle, ready to be executed. Try executing the "package" phase, by double-clicking on it. Maven should go through all phases up to "package", and so it will compile (well, there's nothing to compile so far), run the tests (nothing here either), and create a jar file for you!

5. Fresh from this success, now create your AddressBook program, like you did back in SYSC3110: AddressBook is a class that has-a list of BuddyInfo objects, which each contain the name and phone number of your buddies:
    - Right-click on the package (named after the groupId you provided earlier) inside the java folder in the Project pane, and select "new->class", to create BuddyInfo and AddressBook. Double-check that those classes are located inside the package and have the “package” declaration at the top of their source code (creating these classes in non-default packages will save you headaches down the road…). Populate those classes with the variables and methods they require.
    - IntelliJ provides a lot of contextual help. To access it, select the AddressBook class name in the source, and hit alt-enter. It will offer you many options, among which the possibility to create a corresponding test class. Do it! Be careful to select JUnit 3 or 4 as the target. If IntelliJ doesn’t recognize the JUnit imports or keywords, it could be that your Maven POM is missing a dependency on JUnit. Clicking “fix” should do the job. The pom.xml file should now have the dependency on JUnit (check it!) If not, go back to the Maven link included at the beginning of this lab to manually fix the issue by adding the dependency.
    - Similarly, selecting the class name and clicking alt-insert will provide plenty of automatic code generation options. Feel free to use them but always inspect the result to see if it conforms to what you wanted.

- Complete your basic AddressBook application, with a main() method that simply creates an AddressBook object, populates it with a couple of BuddyInfos and prints the content. Provide unit tests for both AddressBook and BuddyInfo. Using Maven, package your deliverable into a JAR.
- Show your work to the TA and put the entire directory in a zip file and submit on Brightspace.