Faculty of Science

Course:	CSCI 2020U: Software Systems Development and Integration
Lab:	#2
Topic:	Build tool 1

Overview

In this lab, you'll build a project directory, along with some Java code, create a Gradle build. You'll also add this directory to the local and remote Git repositories, to reinforce proper version control practices. The environment can be replicated as a base for your project and assignments.

Part 1: A Basic Java Project

Boot into your Linux partition (or open Linux in a VirtualBox VM). You will create a new directory in your home directory called `csci2020u/lab02`. We'll add this folder to the Git repositories. Use the following steps to complete this lab:

1. Change into the `csci2020u` directory
2. Make a new directory, called `lab02`
3. Add the `lab02` directory to Git
4. In the `lab02` directory, create the directory structure used by Gradle (the files will be initially empty, you can create them with the `touch` command):

```
lab02
 → build.gradle
 → src
     → main
         → java
             → csci2020u
                 → lab02
                     → SimpleTest.java
```

5. Modify the `build.gradle` file to allow compilation and running of a Java program
6. Create a `.gitignore` file to ignore files that are normally not committed to a repository (e.g. .class files)
7. View all of the available Gradle tasks added as a result of the change
8. Add some code to the `SimpleTest.java` file; a simple hello world will suffice, but be sure that the package name matches the directory structure
9. Compile the code
10. Run the code by hand by changing to the `build` directory and using the `java` command
11. Commit changes locally, and push all changes to the remote repository

***Note**: Do not commit any code to the repository without a detailed message describing the change.*

Part 2: A Project with Dependencies

12. Find the Apache Commons CSV library in the Maven repository at `http://search.maven.org/` by searching for `commons-csv`

13. Use the details on the search results to include Apache Commons CSV as a dependency
14. Create a new source file, CSVDemo.java, using the following code (determine the package on your own):

```
import java.io.*;
import org.apache.commons.csv.*;

public class CSVDemo {
   public static void main(String[] args) {
         try {
               Reader in = new FileReader("data.csv");
               Iterable<CSVRecord> records =
CSVFormat.DEFAULT.withHeader("First Name", "Last Name", "SID",
"Grade").parse(in);
               for (CSVRecord record: records) {
                     String lastName = record.get("Last Name");
                     String firstName = record.get("First Name");
                     String grade = record.get("Grade");
                     System.out.println(firstName + " " + lastName + " (" +
grade + ")");
               }
         } catch (IOException e) {
               e.printStackTrace();
         }
   }
}
```

15. Compile the program, using the `classes` task
16. Commit changes locally, and push all changes to the remote repository

***Note**: To run this second program, we'll need some additional skills. We'll continue this in the next lab.*

How to Submit

Show your local and remote repositories to the TA to prove that you have finished this lab. The TA can provide oral feedback if you do not receive full marks for any lab assignment, but it is most appropriate to ask the TA for this feedback in a timely fashion (i.e. ask now, not at the end of the term).