CP2406 Programming 3: Practical 1

This document has been prepared by Dr. Jason Holdsworth for James Cook University. Updated 20 August 2019.

Instructions for students and staff:

- DURATION: two hours
- ATTENDANCE: Students ***must sign/initial the attendance sheet*** provided by staff.
- MARKING: Students complete the tasks for this practical for staff to review with feedback during the practical.
- SUBMISSION: Student must upload completed practicals to LearnJCU. Staff record feedback and assign rubric marks for these submissions on LearnJCU.

TASK-1: Java Editor: IntelliJ IDEA [15-30 min]

IntelliJ IDEA

- JetBrains offers free access for JCU students to the **Ultimate** Edition of Intellij IDEA. Signup here: https://www.jetbrains.com/student/ using your JCU email address, if you haven't already done so in a previous programming subject.
- Alternatively, you could use the **Community** Edition. To install it on your personal computer, go to https://www.jetbrains.com/idea/download/ and download (and then install) the **Community** Edition.
- Locate the installed IntelliJ program on your personal or university computer and launch it.
- *NOTE: Software and websites are regularly updated. If any of the links are broken or referred to an older version of IntelliJ, please Google "hello world in IntelliJ tutorial".*
- Watch the first two Getting-Started tutorials from https://www.jetbrains.com/idea/documentation/ :
  - 1) Running-IntelliJ-for-the-first-time tutorial https://www.youtube.com/watch?v=c0efB_CKOYo
  - 2) Finding your way around IntelliJ (slightly old but useful) https://www.youtube.com/watch?v=X49xqVDR8VQ
- Work your way through the help document on how to create and run your first java program: https://www.jetbrains.com/help/idea/creating-running-and-packaging-your-first-java-application.html
- Show your "hello world" program running to your instructor to be marked off for this task.

TASK-4: Chapter-1 Debugging Exercises [15-30 min]

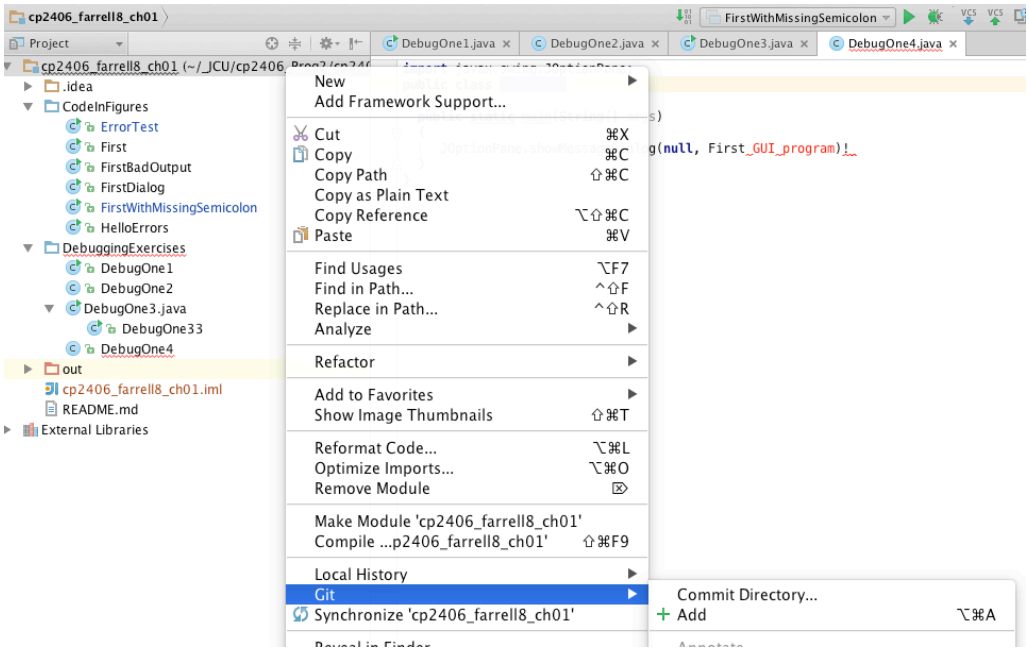
Debugging Exercises

1. Each of the following files in the Chapter01 folder in your downloadable student files has syntax and/or logic errors. In each case, determine the problem and fix the errors. After you correct the errors, save each file using the same filename preceded with *Fix*. For example, DebugOne1.java will become **FixDebugOne1.java**.

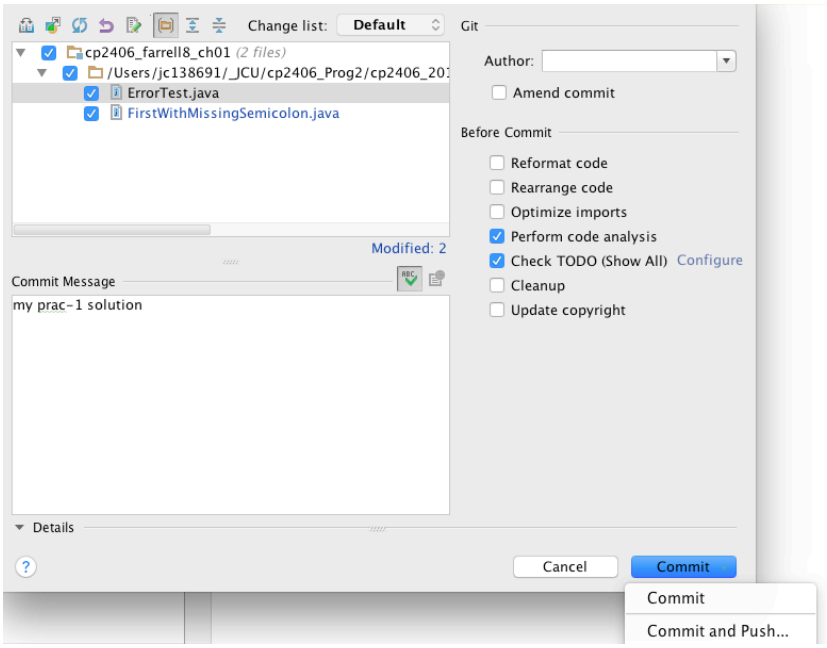
   a. DebugOne1.java
   b. DebugOne2.java
   c. DebugOne3.java
   d. DebugOne4.java

When you change a filename, remember to change every instance of the class name within the file so that it matches the new filename. In Java, the filename and class name must always match.

- The above description of the debugging exercises is from the textbook.
- IntelliJ will display ***compiling*** errors. Work your way through all of them until all compiling errors are fixed. See the following Figure for help:
- Commit and then push your solutions back to the CP2406 repo on GitHub. For help see: https://www.jetbrains.com/help/idea/commit-and-push-changes.html
- If you do not wish to read the help links above, here is an easy way to commit and push the whole project. Right-click on the project name and select Git:CommitDirectory…

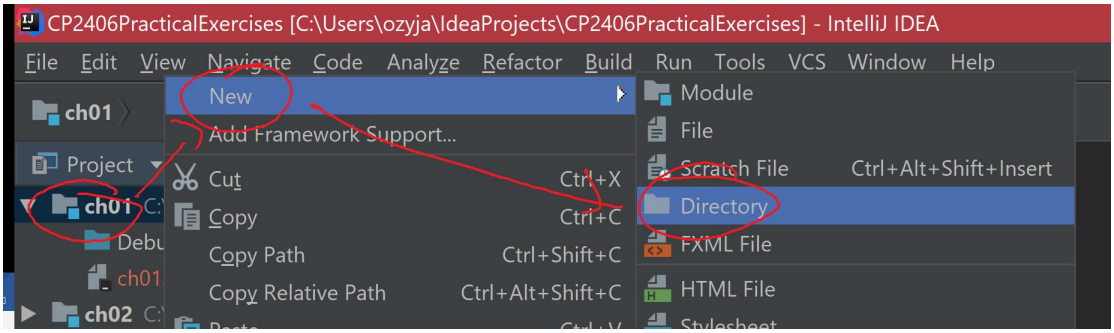


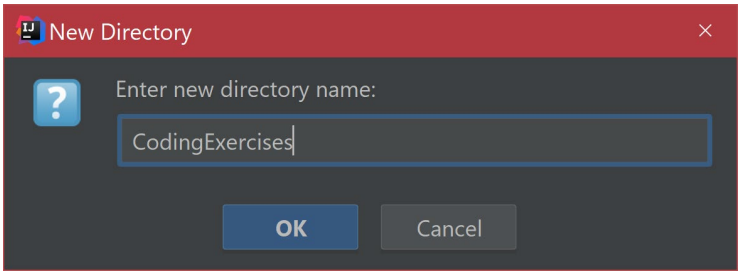
Then Commit and Push in one go (The button “Commit” turns into sub-buttons):



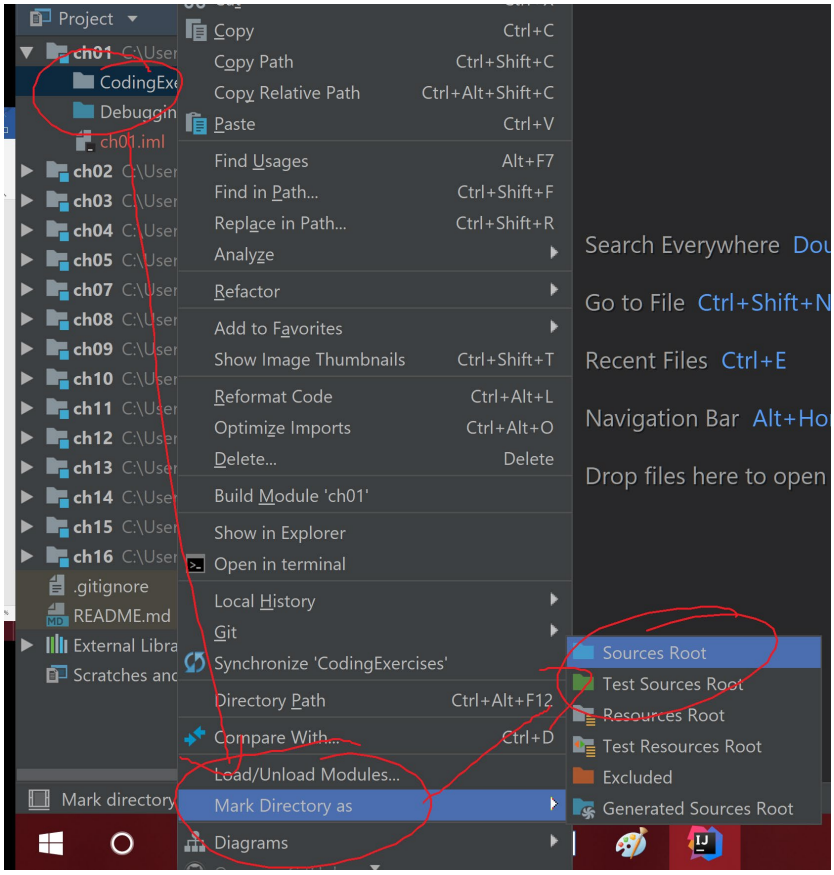
TASK-5: Chapter-1 Programming Exercises [10-20 min]

- Inside “ch01” module, create a new directory called “CodingExercises”:

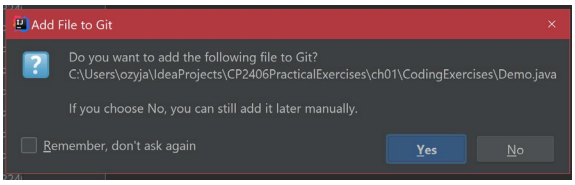




- Then set the new directory to be used for source code:



- Inside this new directory, complete any ***two*** exercises of the following exercises below, ***or as directed by staff***. IntelliJ will ask if you want to add any new source code files – of course you should! The project is under version control.



- **Learning tip:** Try to complete more exercises to learn and practice programming skills. ***Programming is a contact sport!*** You do not become a better programmer by watching others or YouTube.

6. Write, compile, and test a class that displays your favorite movie quote on the screen. Save the class as **MovieQuote.java**.

As you work through the programming exercises in this book, you will create many files. To organize them, you might want to create a separate folder in which to store the files for each chapter.

7. Write, compile, and test a class that displays your favorite movie quote, the movie it comes from, the character who said it, and the year of the movie. Save the class as **MovieQuoteInfo.java**.

8. Write, compile, and test a class that displays the following pattern on the screen:

```
X                       X
X                       X
X       XXXXXXXXXX      X
XXXXX   X         X   XXXXX
X   X   X         X   X   X
X   X   X         X   X   X
```

Save the class as **TableAndChairs.java**.

9. Write, compile, and test a class that displays at least four lines of your favorite song. Save the class as **FavoriteSong.java**.

10. Write, compile, and test a class that uses the command window to display the following statement about comments:

"Program comments are nonexecuting statements you add to a file for the purpose of documentation."

Also include the same statement in three different comments in the class; each comment should use one of the three different methods of including comments in a Java class. Save the class as **Comments.java**.

11. Modify the Comments.java program in Exercise 10 so that the statement about comments is displayed in a dialog box. Save the class as **CommentsDialog.java**.

12. From 1925 through 1963, Burma Shave advertising signs appeared next to highways all across the United States. There were always four or five signs in a row containing pieces of a rhyme, followed by a final sign that read "Burma Shave." For example, one set of signs that has been preserved by the Smithsonian Institution reads as follows:


```
Shaving brushes
You'll soon see 'em
On a shelf
In some museum
Burma Shave
```

Find a classic Burma Shave rhyme on the Web. Write, compile, and test a class that produces a series of four dialog boxes so that each displays one line of a Burma Shave slogan in turn. Save the class as **BurmaShave.java**.