CP2406 Programming-II: Practical-1

This document has been prepared by Dr. Dmitry Konovalov for James Cook University. Updated 23 November 2015.

Instructions for on-campus version:

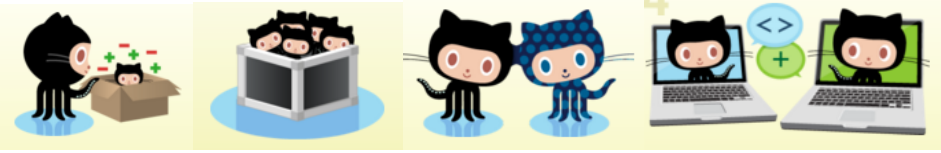
- **WHEN**: Teaching week #2 at JCU; Teaching week #1 at JCUS/JCUB (scheduled after lectures)
- **DURATION**: two hours
- **ATTENDANCE**: compulsory (students must attend). You (student) ***must sign/initial the attendance sheet*** provided by your instructor.
- **MARKING [1 mark]**: Complete the tasks from this practical and show the completed tasks to your instructor. Each completed practical is awarded **ONE participation mark** towards the participation assessment component of this subject.
- **EARLY SUBMISSIONS**: You are encouraged to attempt (and complete) some or all of the following tasks ***before*** attending the practical session.
- **LATE SUBMISSIONS**: You may finish the following tasks in your own time and then show your completed tasks during the following week practical. **The main intent here is to encourage you as much as possible to complete all practicals**. ***If you are late by more than one week***, you will need a valid reason for your instructor to be awarded the marks.

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

IntelliJ IDEA

- In this subject we will be using the **Community** Edition of IntelliJ IDEA from https://www.jetbrains.com/idea/ ;
- 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 you personal or university computer and launch it.
- **NOTE**: Software and websites are regularly updated. So 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=L_jXj0XTwSg
    - 2) Finding your way around IntelliJ 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/idea/help/creating-and-running-your-first-java-application.html
- Show your “hello world” program running to your instructor to be marked off for this task.

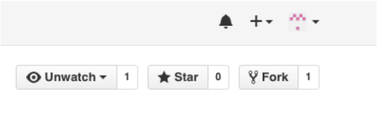
TASK-2: GitHub [15-30 min]



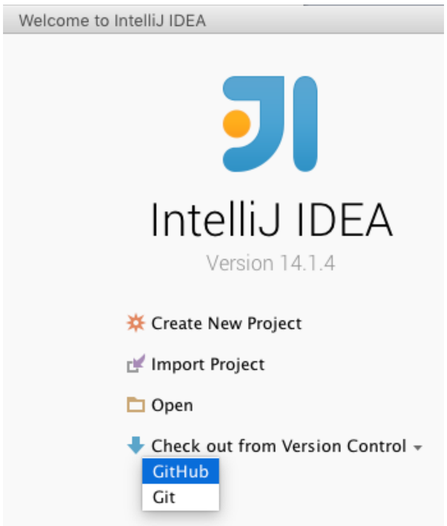
- In this subject we will be using GitHub for all source code management.
- Sign-in (or sign-up) to g https://github.com/
- NOTE: All github repositories are public by default. You need to apply for private repositories as a student (if you have not done so already), see https://education.github.com/guide/private_repos .
- Work your way through the help document on how to connect your IntelliJ to your github account: https://www.jetbrains.com/idea/help/using-github-integration.html
- You should have some knowledge about git-repositories from other subjects (e.g. CP1404-Programming-1). If however, you feel you may need a re-fresher, please work through the github-hello-world tutorial, see https://guides.github.com/activities/hello-world/ .

TASK-3: Connecting Github and IntelliJ [15-30 min]

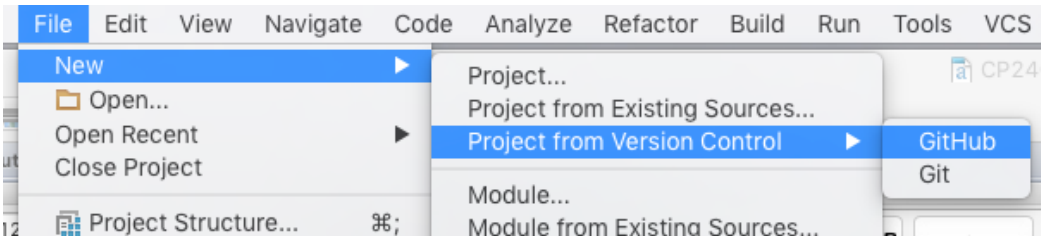
- Sign-in (see task-1 above) to your github account and then click https://github.com/CP2406Programming2/cp2406_farrell8_ch01 repository.
- **Working with GitHub:** You need to make a copy of this repository so you could modify it. This is done by clicking “Fork” (in the upper left corner of the repository page)



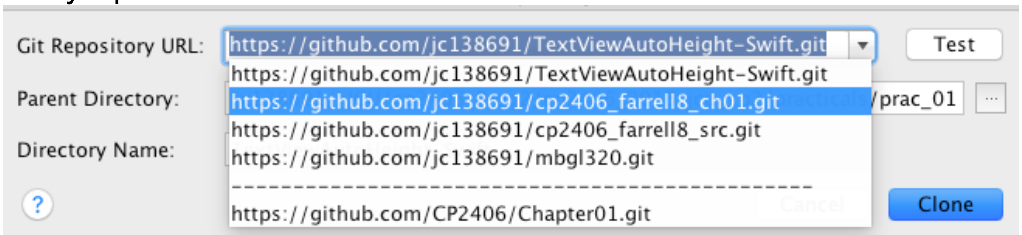
  - For more help with “forking”/copying, see https://guides.github.com/activities/forking/ (or google “how to fork on github”)
  - NOTE: You do no need to make “Pull-requests”. All repos in these subject practicals are for distribution purpose. That is, you do not want your solution source code to be “pulled” into the master copy.
- **Connecting IntelliJ:** Now you should have your own copy (repo) of the source code for this task. ***Please read carefully now!*** It may be confusing, but what you want to end up with is your solution to be ***saved into your github account***.
  - Relevant IntelliJ help page is https://www.jetbrains.com/idea/help/using-github-integration.html
  - Start IntelliJ; Select “Check out from Version Control”, and “GitHub”



-
- If you already have IntelliJ running, go to the Menu and select File:ProjectFromVersionControl:GitHub



- You will be presented with a few steps, where you must enter ***your github login details***.
- Once IntelliJ connected you to your github account, you should be able to select today's prac. Select and click "Clone".



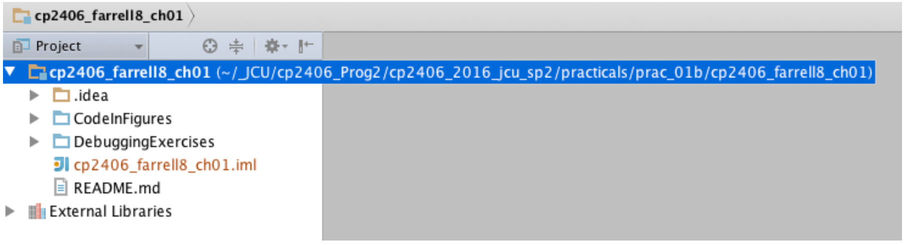
Git Repository URL: https://github.com/jc138691/TextViewAutoHeight-Swift.git

https://github.com/jc138691/TextViewAutoHeight-Swift.git
https://github.com/jc138691/cp2406_farrell8_ch01.git
https://github.com/jc138691/cp2406_farrell8_src.git
https://github.com/jc138691/mbgl320.git
https://github.com/CP2406/Chapter01.git

Parent Directory: /prac_01

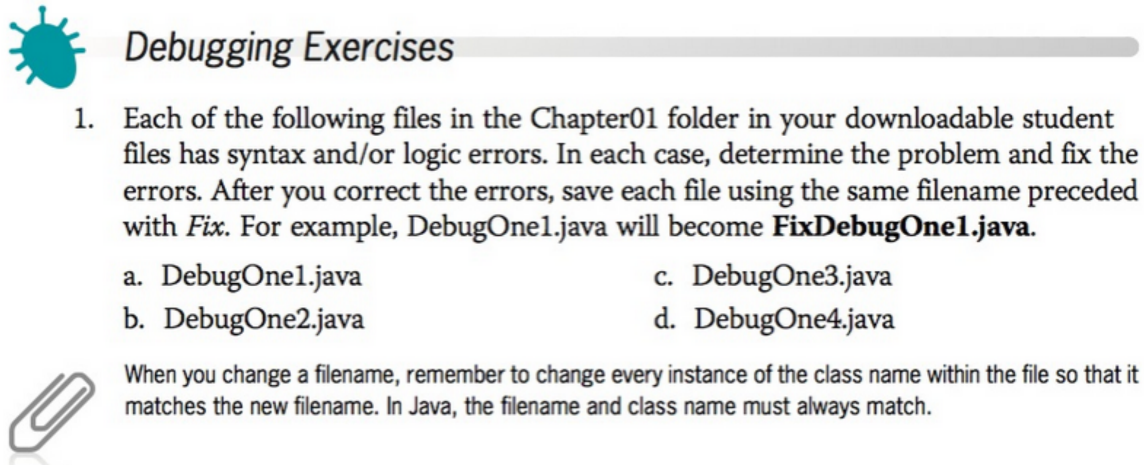
Directory Name:

- If you do not see "cp2408_farrell8_ch01.git", then you did not forked/cloned the master repo from https://github.com/CP2406Programming2/cp2406_farrell8_ch01 . Go back to the beginning of this Task-3 and try again.
- IntelliJ will present you with the new-project-wizard pages. Accept all defaults and you should end up with something like this:



cp2406_farrell8_ch01
Project
cp2406_farrell8_ch01 (~/_JCU/cp2406_Prog2/cp2406_2016_jcu_sp2/practicals/prac_01b/cp2406_farrell8_ch01)
.idea
CodeInFigures
DebuggingExercises
cp2406_farrell8_ch01.iml
README.md
External Libraries

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



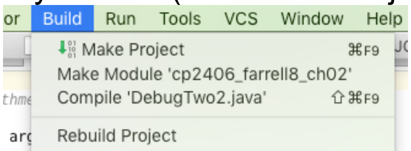
Debugging Exercises

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**.

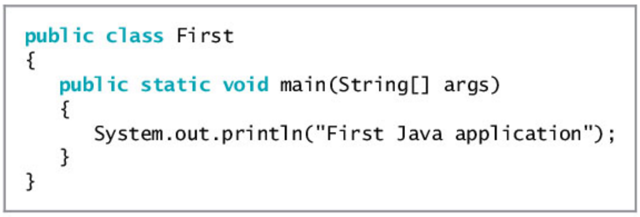
   a. DebugOne1.java
   b. DebugOne2.java
   c. DebugOne3.java
   d. DebugOne4.java

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.
- Try to build (aka "Make Project") your chapter-1 project:

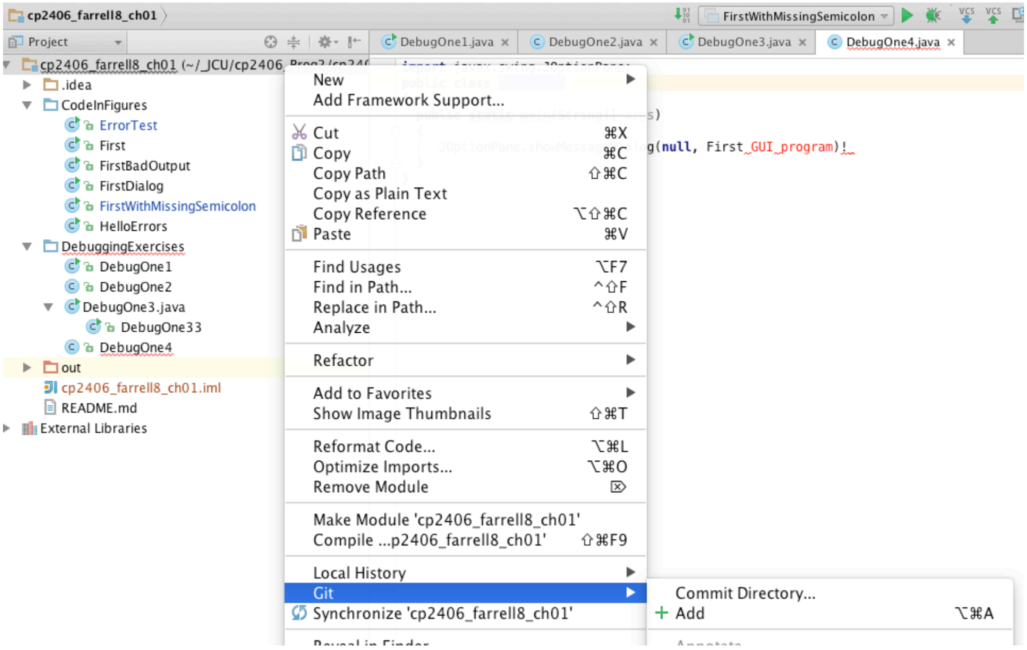

or
Build
Run
Tools
VCS
Window
Help
Make Project ⌘F9
Make Module 'cp2406_farrell8_ch02'
Compile 'DebugTwo2.java' ⇧⌘F9
Rebuild Project

- If needed, see https://www.jetbrains.com/idea/help/creating-and-running-your-first-java-application.html#run_app on how to do this.
- IntelliJ will display ***compiling*** errors. Work your way through all of them until all compiling errors are fixed. See the following Figure for help:

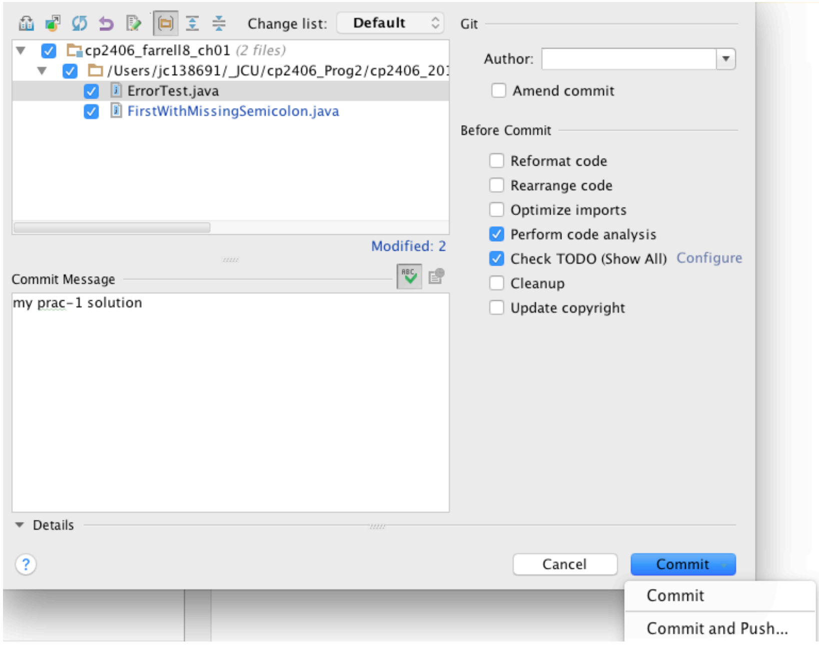


```
public class First
{
   public static void main(String[] args)
   {
      System.out.println("First Java application");
   }
}
```

- **Figure 1-4** The `First` class
- Commit (see https://www.jetbrains.com/idea/help/committing-changes-to-a-local-git-repository.html) and then push (see https://www.jetbrains.com/idea/help/pushing-changes-to-the-upstream-git-push.html) your solution back to your github account.
- 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):


Change list: Default
Git
cp2406_farrell8_ch01 (2 files)
/Users/jc138691/_JCU/cp2406_Prog2/cp2406_201
ErrorTest.java
FirstWithMissingSemicolon.java
Modified: 2
Details
Cancel
Commit
Commit
Commit and Push...

Author:

☐ Amend commit

Commit Message

my prac-1 solution

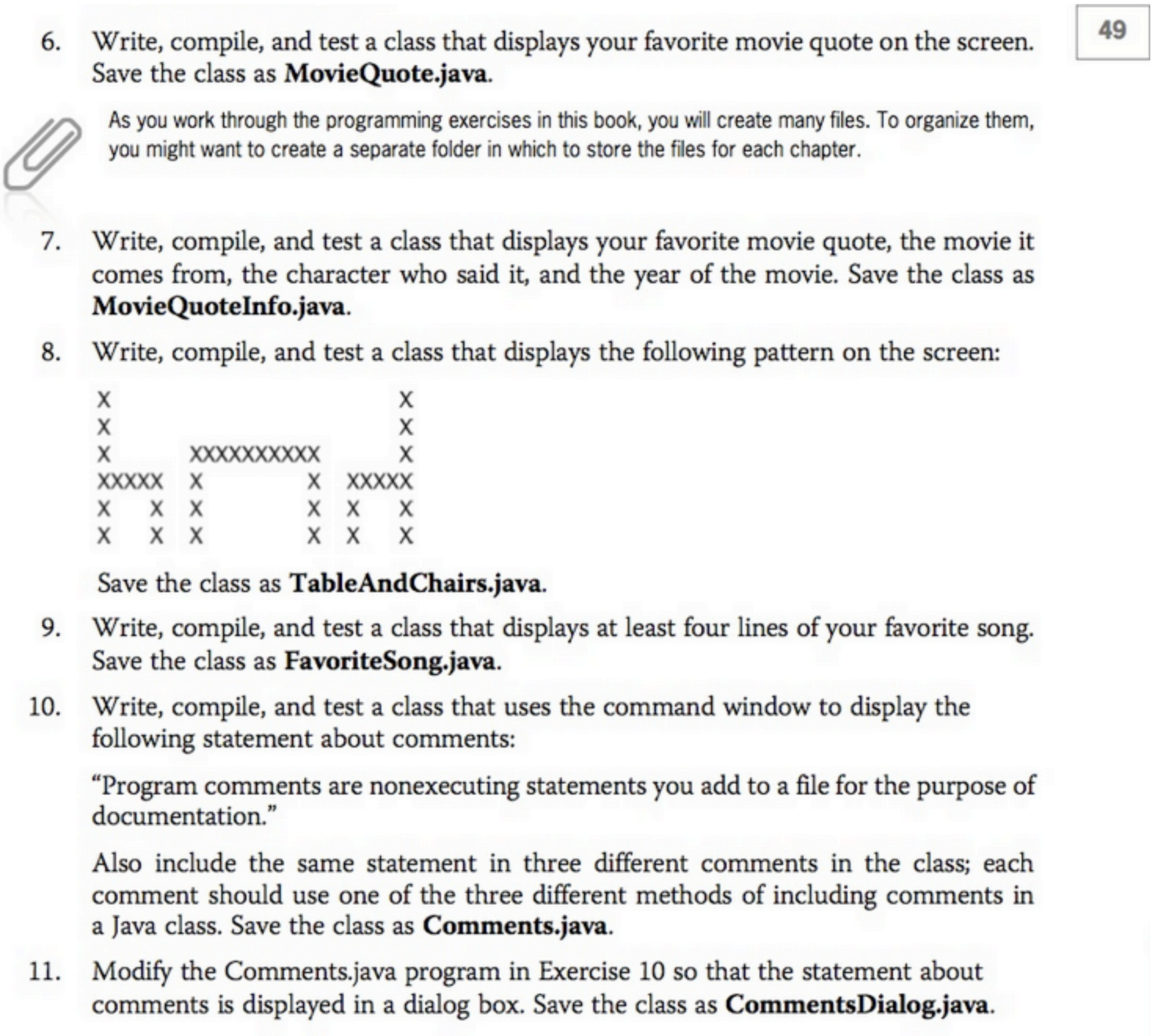
Before Commit

☐ Reformat code
☐ Rearrange code
☐ Optimize imports
☑ Perform code analysis
☑ Check TODO (Show All) Configure
☐ Cleanup
☐ Update copyright

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

- Complete any ***two*** exercises from the following list, ***or as directed by your instructor***.
- **Learning tip:** Try complete more exercises to learn and practice programming skills. ***Programming is a contact sport!*** You do not become a better programmer by watching others.
- First try to complete the exercises without looking in the textbook (google for Java classes if needed).
- If you need more help, look through the chaper-1 textbook, source code and/or debugging code, https://github.com/CP2406Programming2/cp2406_farrell8_ch01 (or your own fork).

- If you are still struggling, pick at the solutions, https://github.com/CP2406Programming2/cp2406_farrell8_prac_solutions/tree/master/Chapter01/ProgrammingExercises .



49

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**.

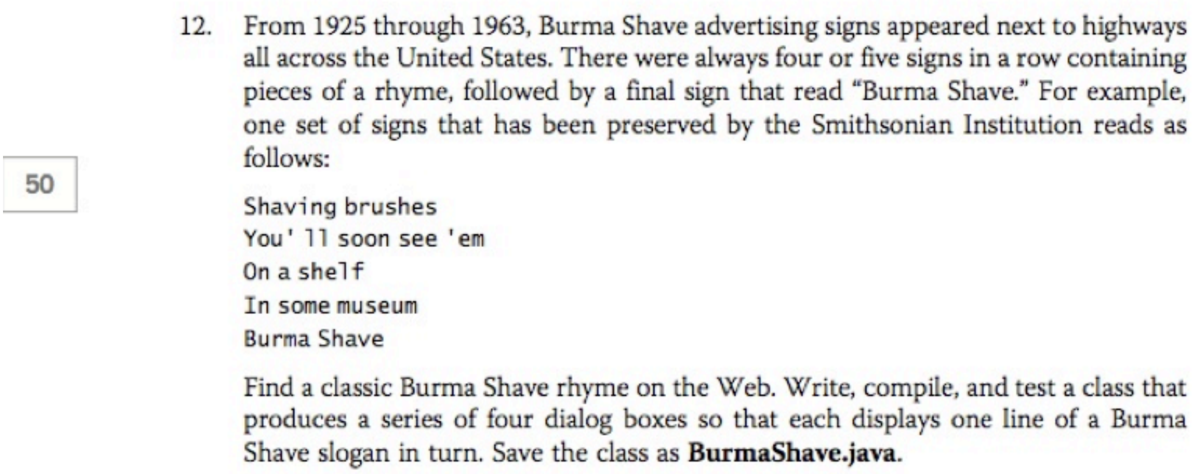
9. Write, compile, and test a class that displays at least four lines of your favorite song. Save the class as **FavoriteSong.java**.

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**.



50

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**.

=== END OF THIS PRACTICAL ☺ ===