Project 9

CS 1323, Fall 2015

# Learning Objectives

1. Create a main program that calls user defined methods. (15 points; 10 points for each method)
2. Transfer existing code into methods. (30 points; 10 points for each method)
3. Transfer information to a method using arguments. (10 points)
4. Receive information from a method using a return value. (10 points)
5. Declare a local variable in a method other than the main method. (10 points)
6. Change an oversize array implementation to a perfect size array implementation. (15 points; 5 points for each method)

10 points will be awarded for the documentation of your program. That means using good names for variables, proper and consistent indentation of code, comments and meaningful use of whitespace.

Section 10: When your program is completed and running, have the teaching assistants check it to get credit for the lab. If code is submitted after the laboratory, this project is due by 11:59 p.m. on October 26.

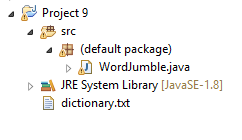
Sections 1 and 995: This project is due by 11:59 p.m. on October 26.

# Description

Software maintenance is the process of taking existing software and adding new features, restructuring, or improving it. This week we will do software maintenance by taking a program that was written in a single method and separating it into multiple methods. Once that has been done, we’ll improve the structure of the program by changing the data structure (moving from an oversized array to a perfect sized array). Software maintenance is a very common job for computer scientists who are early in their careers.

This code is available on Janux, under Project 9. It is distributed as a zip file. You’ll need to import that file (File -> Import -> General -> Archive File). The dictionary.txt file should be imported into the home directory of your Project 9 project. The WordJumble.java file should be imported into the src directory of your project. If you do not put these files in the proper directories the programs will not run.

This is what the package explorer should look like when the files are successfully imported.



Then, you will need to play the game to see how it works. This game was inspired by jumbles, where you’re given the letters in a word in the wrong order and have to guess the word. The dictionary file included contains some really tough words! You may need to cheat (for example, by printing out the word being guessed) to practice winning—I sure did!

The next thing to do is to read the code carefully. Reading someone else’s code is different from writing your own. Think about which part of the game is implemented by each segment of code and how you could break this program apart into smaller and easier to understand pieces.

I’ve created a worksheet that describes some methods that make sense for this program, and helps you figure out how to separate parameters, local variables, and return values. I recommend that you work through one method in the worksheet first (and check answers with your lab partner if you’re in Section 10). Then go to the code and program that one method. I filled in the first one to help you get off to a good start.

Once you have the program running with this new and improved design, it is time to change the program to using a perfect sized array instead of an oversized array. This may change the signature of the methods, and will change the implementation of one method substantially.

# StringBuilders

When we are doing things like shuffling the letters in a word, the String class is a poor choice. The String class is designed for words that aren’t going to be changed. Adding and deleting characters in a word is a change. There is another class that is used to handle words that can be changed. It’s called StringBuilder. It works very much like a String, except it doesn’t have any special syntax (like being able to append using + or the special constructor).

You do not need to know anything about the StringBuilder class to be able to accomplish this project. The code that is given uses it, and it will be hidden inside a method. You should not need to modify any code that uses a StringBuilder.