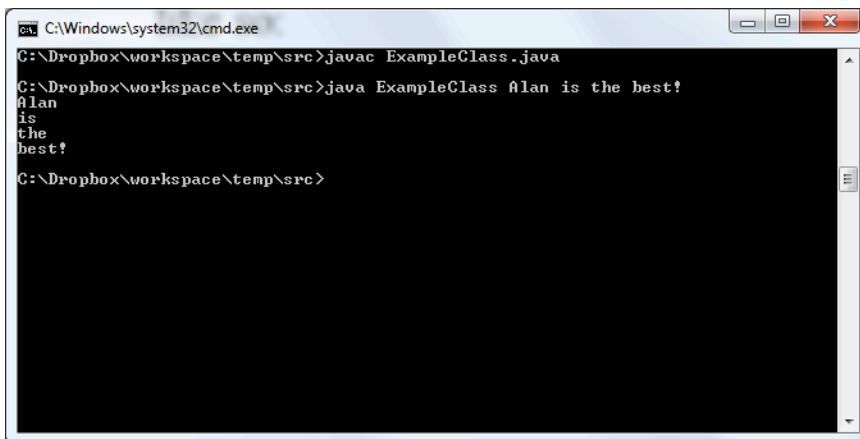COSC 201
Command Line Usage Handout
Alan C. Jamieson
Last Updated: 9/27/2012

The primary purpose of this handout is to walk you through using command line arguments in Java, specifically in Eclipse. All of the examples in this document are from Eclipse running on a Windows machine, but the Mac version is very similar.

I've highlighted a segment of the following code:

```
public class ExampleClass{
        //This main method simply prints out all of the arguments
        //passed through the command line
        public static void main (String [] args){
                for (int i = 0; i < args.length; i++){
                        System.out.println(args[i]);
                }
        }
}
```
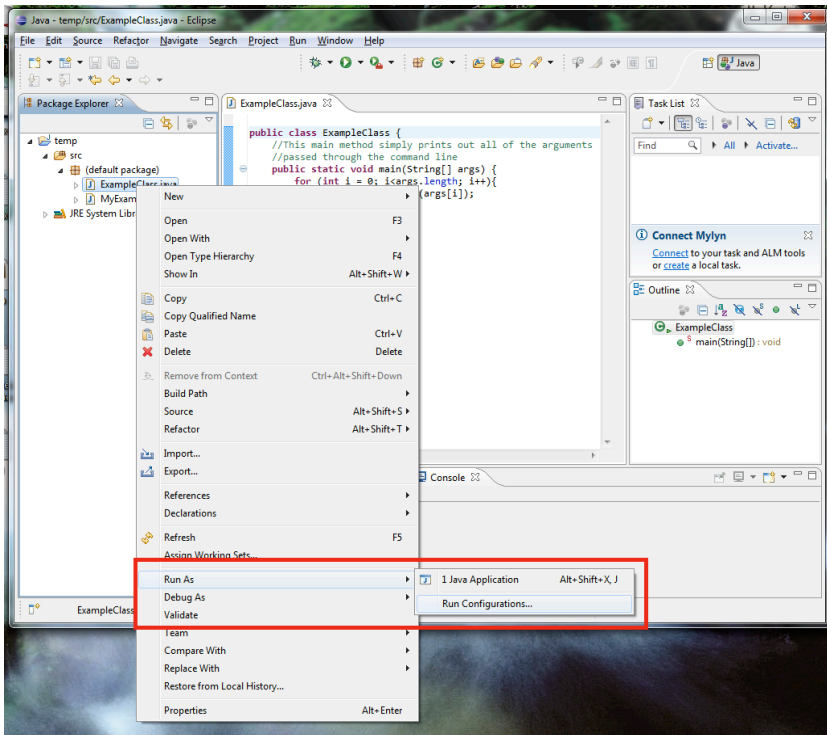
This is a mandatory part of the main method signature, but as of yet, you likely haven't used it before. This parameter represents a String array that holds all information that is passed in to the execution of your Java class at the command line. This harkens back to an earlier time when you typically would do your compilation and run through a terminal, like so:
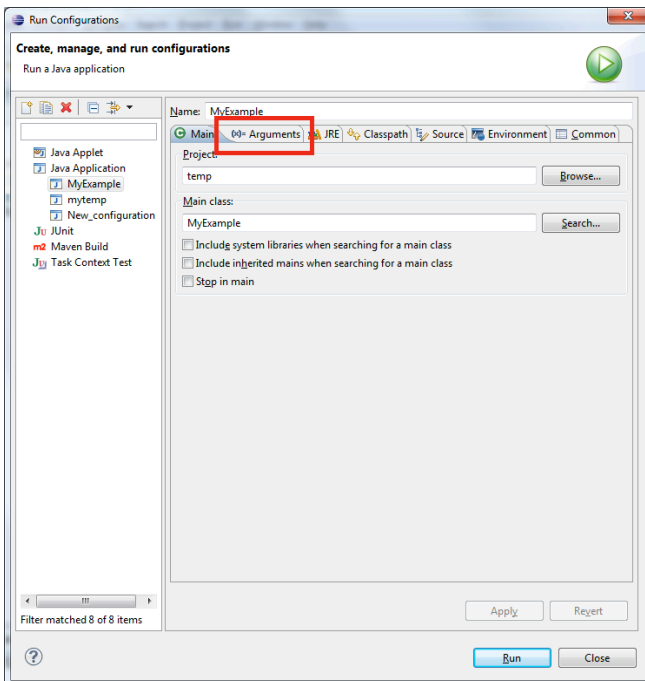

Yes, you can still compile and run your program this way by using the Command Prompt in Windows, or the Terminal in OS X. Notice what I did in the second line? I added several words to the end of my "java ExampleClass" execution. These are my command line arguments, and they are stored in args, word by word (delimited by space). So, when I execute ExampleClass, I get the output above.
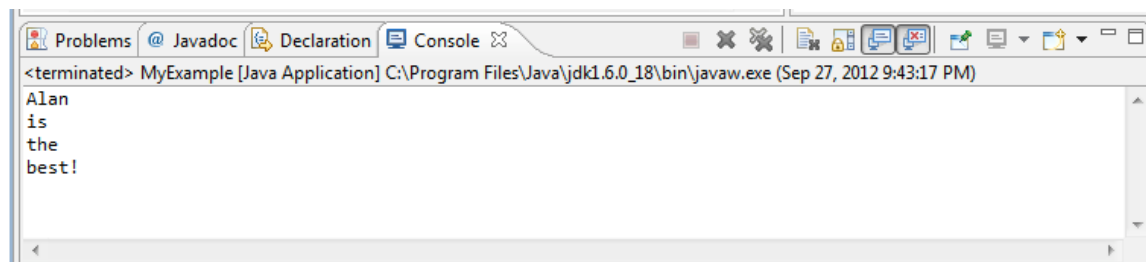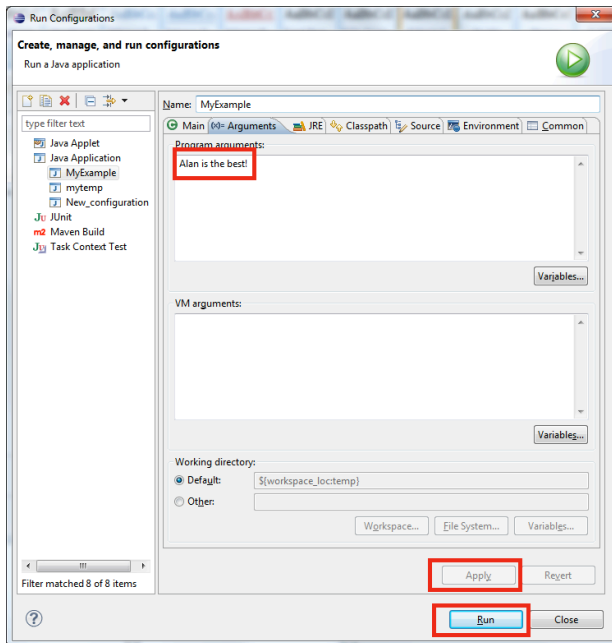
To get this same kind of behavior in Eclipse, we have to do some digging. First, right click on the source file with your main method (ExampleClass for instance) and go to Run As -> Run Configurations:

You'll see the following window pop up. We want to select the Arguments tab:

Then, type in your arguments in the Program arguments box, click Apply, then click Run.

All done! As always, let me know if you have any questions or concerns about this handout.