LAB #6 – Command Line Arguments: String[] args

You will learn to take command line arguments from the user using Eclipse and uploading your .java program to access.engr.oregonstate.edu to compare the differences. Below is an example of how you supply command line arguments to a program in Eclipse.

1. Begin by writing a small program that takes the strings “hello” and “world” as command line arguments and prints it to the console:

```java
public class ArgsTest {
    public static void main(String[] args) {
        System.out.println("Your message is: " + args[0] + args[1]);
    }
}
```

2. First, run this program normally, and what do you notice? Now, select Run, Run Configurations... (or Pull down the menu on the green run button and select Run Configurations...). You should get a window that looks like the following:
3. Make sure the right program is selected on the left side of the Run Configurations window, and click on the Arguments tab at the top. Enter the two strings, hello and world, and press the Run button.

Notice that even though you enter a space between the Strings in the arguments, this space does not print. This is because each string is its own token to the String[] args.

4. Now, upload this same program to access.engr.oregonstate.edu. Compile the program using `javac ArgsTest.java`, and run the program using:

```
java ArgsTest hello world
```

5. You will change the ArgsTest.java program to compute the integration of $2x^2$ using the trapezoid method. For some of you, this gives you a chance to get caught up on assignment #3, and for others, this will be a simple copy and paste from your assignment.

6. You will supply the program with command line arguments for the a, b, and N values, in that order.

7. **Error Handling**: Your program MUST print an error message to the user when no arguments are supplied or when the user doesn’t supply enough arguments. **Hint**: Use `args.length`. 
**Extras:**

- Extra error handling: Print error message if the user doesn't supply a number for any one of the values, if N is not an integer value, and when the a and b values are switched.
- Change the program so that it can read the a, b, and N value in any order. Hint: you will need to use options with the values to determine the tokens you are reading, i.e. –a 0 –b 1 –N 10.
- Print an error message if the user forgets any of these options/values.