Using Command Line Arguments
What are command line arguments?

When you run programs from the command line (pictured below) you enter the name of the program followed by a list of arguments that the program can access (or ignore).

When you run a Java program, you would first type the name of the interpreter, followed by the name of the class to run, followed by a list of arguments. Something like this:

```
java EchoWords hello world
```
"Command Line" Arguments in Eclipse

So when you run your program from Eclipse, where do the "command line" arguments go?

You can specify them by
- going to the Run menu,
- selecting Run Configuration
- going to the Arguments tab.

The command line arguments can be entered in the Program arguments dialog.
Using Command Line Arguments

The operating system reads the command line arguments and passes them to the Java Virtual Machine. Then JVM passes them to your class inside the `String [] args` array that is the parameter to `main()`.

The length of the array `args.length` tells your program how many arguments there are. You should always check that the user provided as many arguments as your program expects before trying to access them.

The array locations are filled with individual arguments:

```
    args[0], ..., args[args.length - 1]
```

All the arguments are passed as `String` objects. Your program needs to convert them to other types, if need be.
Example

```java
public class EchoWords {

    public static void main(String[] args) {
        System.out.printf("There are %d command line arguments.\n",
                           args.length);
        for (int i = 0; i < args.length; i++) {
            System.out.println(args[i]);
        }
        for (int i = args.length - 1; i >= 0; i--) {
            System.out.println(args[i]);
        }
    }
```