CS102

Introduction to
data structures, algorithms,
and object-oriented
programming

April 12th, 2017

Problems with lab 8

- When reading from a file, always check that the input file is not empty.
- The counting error.
- Looking into data files.
- How to use my starter files.

Reading Command-Line Arguments

- Command-line arguments are read through the main method's array of Strings parameter, usually called args (or whatever you want to call this parameter to main).

  IMPORTANT: *You must have a non-empty file in the same directory as the TestReadWriteSplit.class file called "in.txt" when you run this program!!* Also, any file called "out.txt" in the current directory will be overwritten.

  All I/O files should be closed when you are done with them.

Reading Command-Line Arguments

- In the version of the TestReadWriteSplit program on the next slide, suppose args[0] = "input.txt" and args[1] = "output.txt" during execution of the program.
- You can run this file in the Interactions window of DrJava (after it compiles with no syntax errors) by typing:

```
java TestReadWriteSplit input.txt output.txt
```

- Before this will work, you need to create a non-empty file in the same directory as your Java program called "input.txt". If you have any file in the directory that is already called "output.txt", its contents will be overwritten.

```
4. 1. import java.io.*;
 2. public class TestReadWriteSplit {
 3.    public static void main (String[] args) throws Exception{
 4.     BufferedReader fileIn = new BufferedReader
                          (new FileReader(args[0]));
 5.     PrintWriter outFile = new PrintWriter
                         (new FileWriter (args[1]));
 6.     String line;
 7.     String[] token;
 8.     while ((line = fileIn.readLine()) != null) {
 9.         token = line.replaceAll("[^a-zA-Z ]","")
                .toLowerCase().split("\\s+");
10.          String lcString = "";
11.          for(int i = 0; i < token.length; i++) {
12.            lcString += token[i];
13.          }
14.          outFile.println(lcString);
15.     }
16.     fileIn.close();
17.     outFile.close();
18.   }
19. }
```

Swing Component for large text blocks

- JTextArea:

  You can read and display multiple lines of text in a swing component knows as a JTextArea
  .
  Often used along with a JScrollBar so that the text can be larger than the size of the component.

  Does not generate an action event when enter is pressed.

Debugging your code

1. Most professional programmers use what are called debuggers, for finding logical errors in the code. You will learn about this in CMPU 203.
2. Many of the people I work with and went to school with, use the insertion of print statements to find out where an error is occurring rather than using a debugger.
3. If your program goes into an infinite loop, the first place to check is inside a while loop or, less often, a for loop. Just put a print statement inside the loop and if the loop is infinite you'll see a large amount of text printed. You will probably have to use the Reset button (in DrJava) or do a CTRL-C if running at the command line.

7

Debugging your code (cont.)

4. Using print statements helps you determine what your input consists of. If you're not sure a file is being opened, print out the name after it's read.
5. One drawback to using print statements to find errors is that you need to remove them all when your program is ready.
6. Some programmers set a boolean flag debug that is set to true or false at the top of the area you want to test. If debug is true, the println will occur. Otherwise, all println will be bypassed. You need to use an if-else to check for whether the debug flag is true before printing.

8

Before Submitting your Progams

1. Make sure your name is at the top of the file for every file in program. In NetBeans, you should type your name after the @author clause.
2. Remove commented-out code.
3. Remove instructions except where they are a useful way to understand the code
4. Provide a header for every method that explains what the method does.
5. Be sure your code is indented properly.
6. If you forget to do one or more of the above, resubmit the file. You can resubmit any number of times.
7. Always think about the next reader of your code...will they be able to understand what the code does?

9

Class Exercise

- Create a GUI that has a single square JButton in the exact center of the window. When clicked by user, the background color of this JPanel turns red.

  After class, see CenterButton.java

- Go over lab 8 – writing the program of assignment 5.

10

Adding KeyListeners

- public void keyPressed(KeyEvent evt);
- public void keyReleased(KeyEvent evt);
- public void keyTyped(KeyEvent evt);

Using KeyListeners is much the same as using Mouse or ActionListeners, although KeyEvents are generated by pressing a key on the keyboard and responding to the system callback on keyPressed, keyReleased, or keyTyped. Key events don't need be associated with particular components when the GUI is set up.

11

Two-Dimensional Arrays

- Declared as:

```
int[][] multTable = new int[rows][cols];
```

- Like all arrays, the size is fixed and must be set when the array is instantiated:

```
multTable = new int[9][9];
for (int i = 1; i < int.length; i++) {
    for (int j = 1; j < int[i].length; j++) {
            multTable[i][j] = i * j;
     }
 }
```

12

Two-Dimensional Arrays

- Not to be confused with parallel arrays.

  parallel arrays are arrays of the same size that are processed together.

Show picture of two-dimensional array.

"Ragged arrays" occur when each row of the array may have a unequal number of columns

```
triangularArray = new double[4][];
for (int i=0; i < 4; i++) {
    triangularArray = new double[i + 1];
}
```

13