

Assignment: WordCount

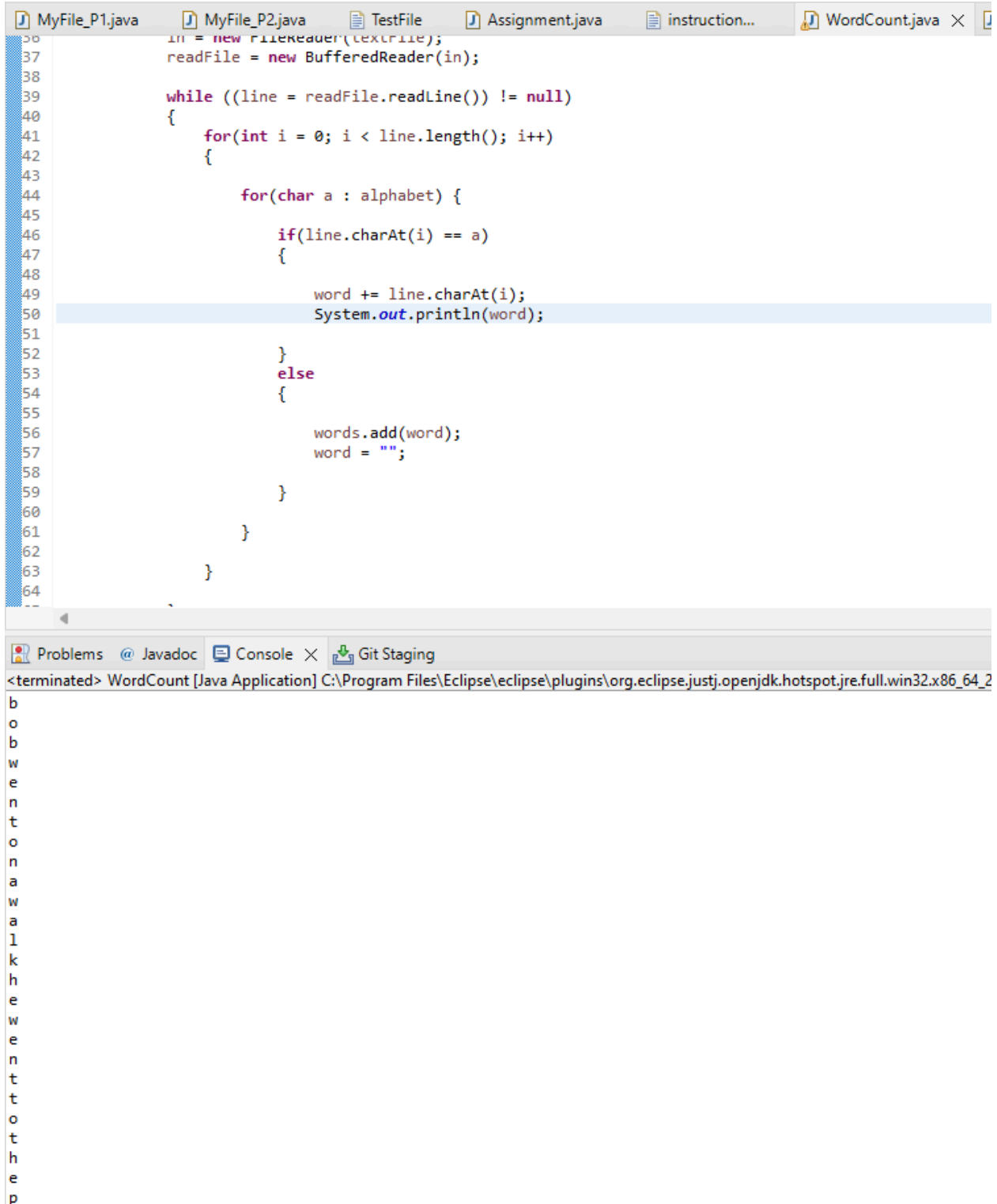
Course: CSE2130 - File Structures and Exception Handling

How has your program changed from planning to coding to now? Please explain?

My initial plan was to use an array of the alphabet to separate the words and count the letters to solve for average word length.

```
String[] alphabet = {"a", "b", "c", "d", "e", "f", "g",  
                    "h", "i", "j", "k", "l", "m", "n",  
                    "o", "p", "q", "r", "s", "t",  
                    "u", "v", "w", "x", "y", "z"};  
  
String word = "";
```

I ditched this plan because I kept getting errors and the array list wasn't working like I wanted it to.



The screenshot shows the Eclipse IDE with several tabs open: MyFile_P1.java, MyFile_P2.java, TestFile, Assignment.java, instruction..., and WordCount.java. The WordCount.java tab is active, displaying the following Java code:

```
36 in = new FileReader(textFile);
37 readFile = new BufferedReader(in);
38
39 while ((line = readFile.readLine()) != null)
40 {
41     for(int i = 0; i < line.length(); i++)
42     {
43
44         for(char a : alphabet) {
45
46             if(line.charAt(i) == a)
47             {
48
49                 word += line.charAt(i);
50                 System.out.println(word);
51
52             }
53             else
54             {
55
56                 words.add(word);
57                 word = "";
58
59             }
60
61         }
62     }
63 }
64
```

Below the code editor, the console window is visible, showing the output of the program:

```
<terminated> WordCount [Java Application] C:\Program Files\Eclipse\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_2
b
o
b
w
e
n
t
o
n
a
w
a
l
k
h
e
w
e
n
t
t
o
t
h
e
p
```

The array would just end up a bunch of letters.

Instead I used an array of non-characters then used the string split method to separate the string into words and used a for loop and counter variables to count the amount of words and record the length of the words.

```
// For loop to iterate through each element in the array
for (String s : words)
{
    System.out.print(s + " ");

    // Calculates number of words and average word per word(element of array)
    numWords += 1;
    wordLength += s.length();
}
}
```

The hardest part for me was correctly splitting the string with the correct amount of words and average length.