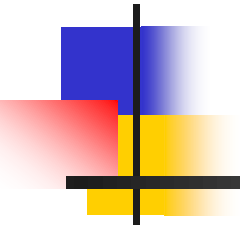


Project 7: Looping

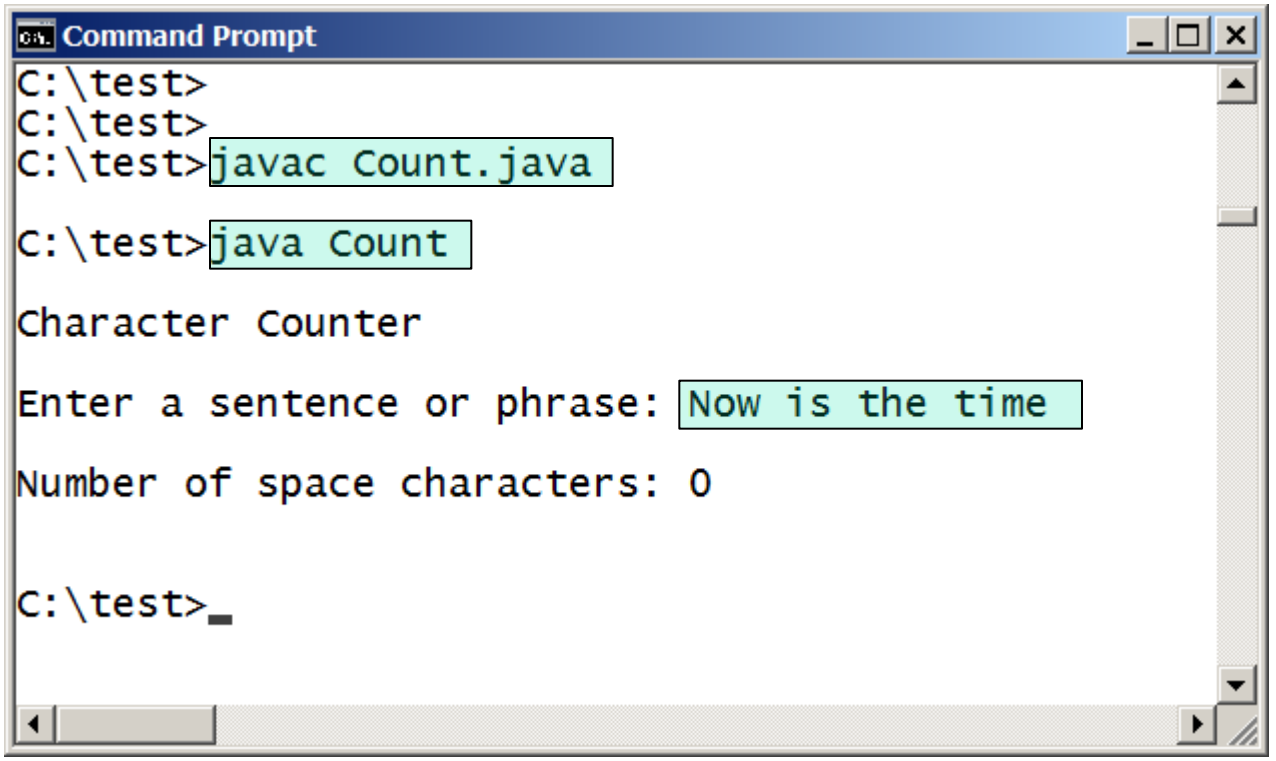
Project 7

- For this project you will produce two Java programs.
- The requirements for each program will be described separately on the following slides.
- Your .java source files for both programs will be submitted together as one assignment.

Program 1: Counting Spaces

- The file *Count.java* contains the skeleton of a program to read in a string (a sentence or phrase) and count the number of blank spaces in the string.
- Create a test directory on your computer and download Count.java from the class web site:
  - http://www.csee.usf.edu/~turnerr/Programming_Concepts/Downloads/Project_7/
- Compile and run the program as downloaded.
  - See next slide.

Initial Version of Count.java



Count.java

- The program currently has the declarations and initializations and prints the results. All it needs is a loop to go through the string character by character and count the space characters

- Since we know how many characters there are (the *length* of the string) we can use a count controlled loop.
  - *for* loops are especially well-suited for this.

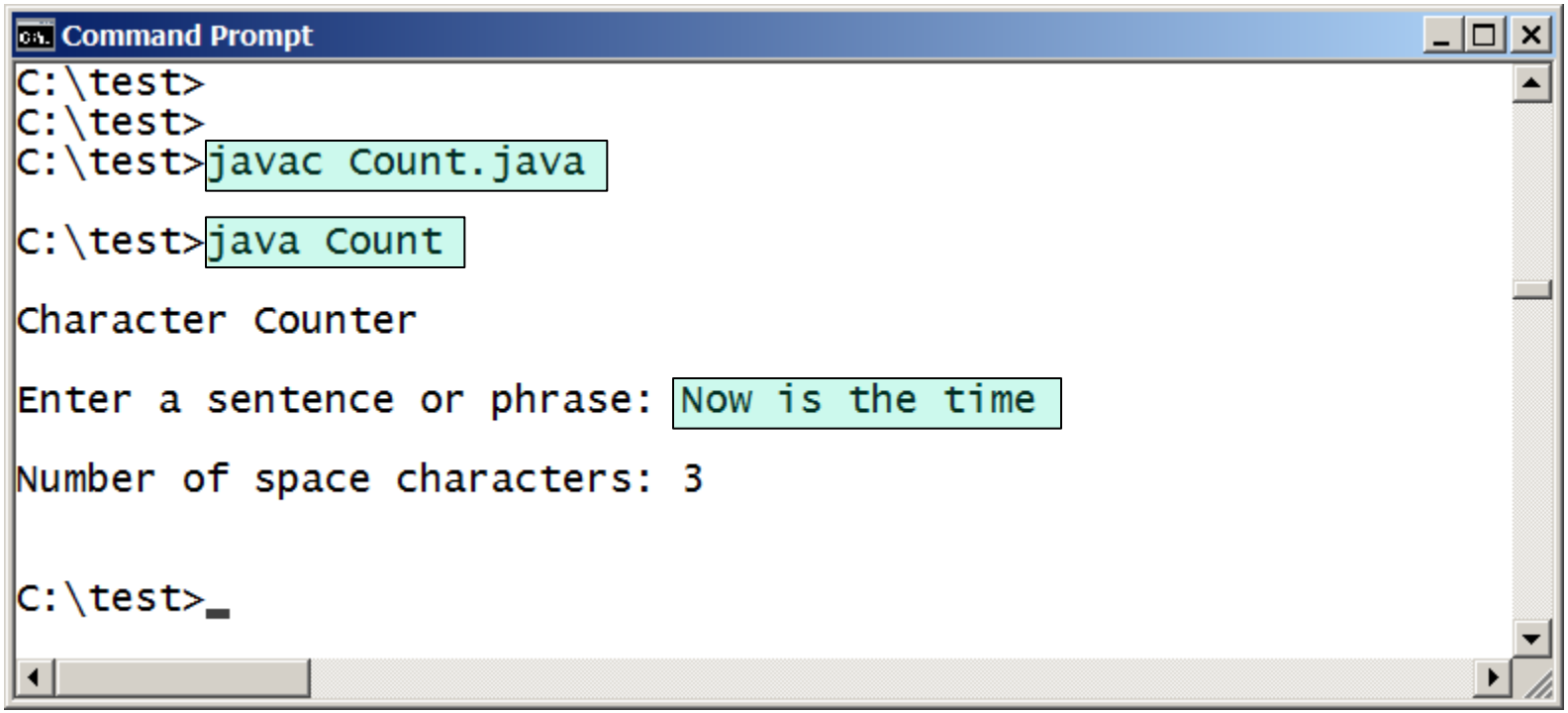
Add the `for` Loop

- Use the String method **`charAt()`** to examine the character at a specified index in a string.
  - Remember that the first character is at index 0.
- The expression

  ```
  phrase.charAt(i);
  ```

  gets the character at position i of phrase.
- Add a for loop to examine each character in turn and count the space characters.
  - Compare to the character literal `' '`
  - That's a single space character inside single quotes.

Sample Run



Test your program on several phrases to make sure it is correct.

Next Version of Count.java

- Extend your program to count additional characters:
  - a, e, s, and t (upper and lower case)
- Declare and initialize four additional counting variables.
  - e.g., aCount, eCount, etc.
- Replace the current *if* with a switch to test for the 9 cases that we want to count.
  - Upper and lower case a, e, s, and t, and space.

The switch Statement

Your switch statement should like somthing like this:

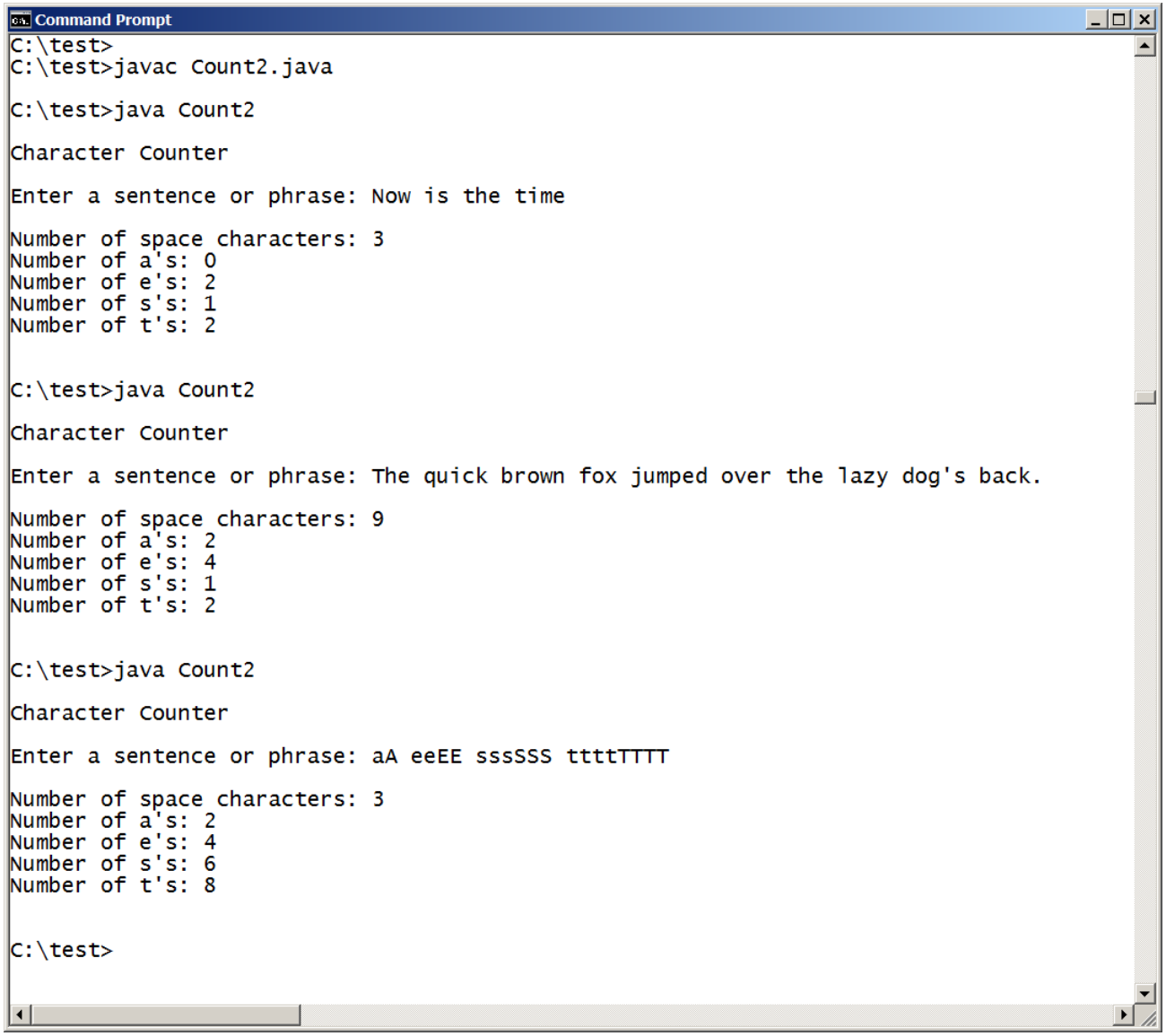
```
switch (ch)
{
    case 'a':
    case 'A':  countA++;
               break;

    case ....

}
```

Add statements to output all of the counts.
Compile and test your program.

Sample Runs


Command Prompt
C:\test>
C:\test>javac Count2.java
C:\test>java Count2
Character Counter
Enter a sentence or phrase: Now is the time
Number of space characters: 3
Number of a's: 0
Number of e's: 2
Number of s's: 1
Number of t's: 2
C:\test>java Count2
Character Counter
Enter a sentence or phrase: The quick brown fox jumped over the lazy dog's back.
Number of space characters: 9
Number of a's: 2
Number of e's: 4
Number of s's: 1
Number of t's: 2
C:\test>java Count2
Character Counter
Enter a sentence or phrase: aA eeEE sssSSS ttttTTTT
Number of space characters: 3
Number of a's: 2
Number of e's: 4
Number of s's: 6
Number of t's: 8
C:\test>

Make it more user-friendly

It would be nice to have the program let the user keep entering phrases rather than having to restart it every time. To do this we need another loop surrounding the current code.

That is, the current loop will be nested inside a new loop.

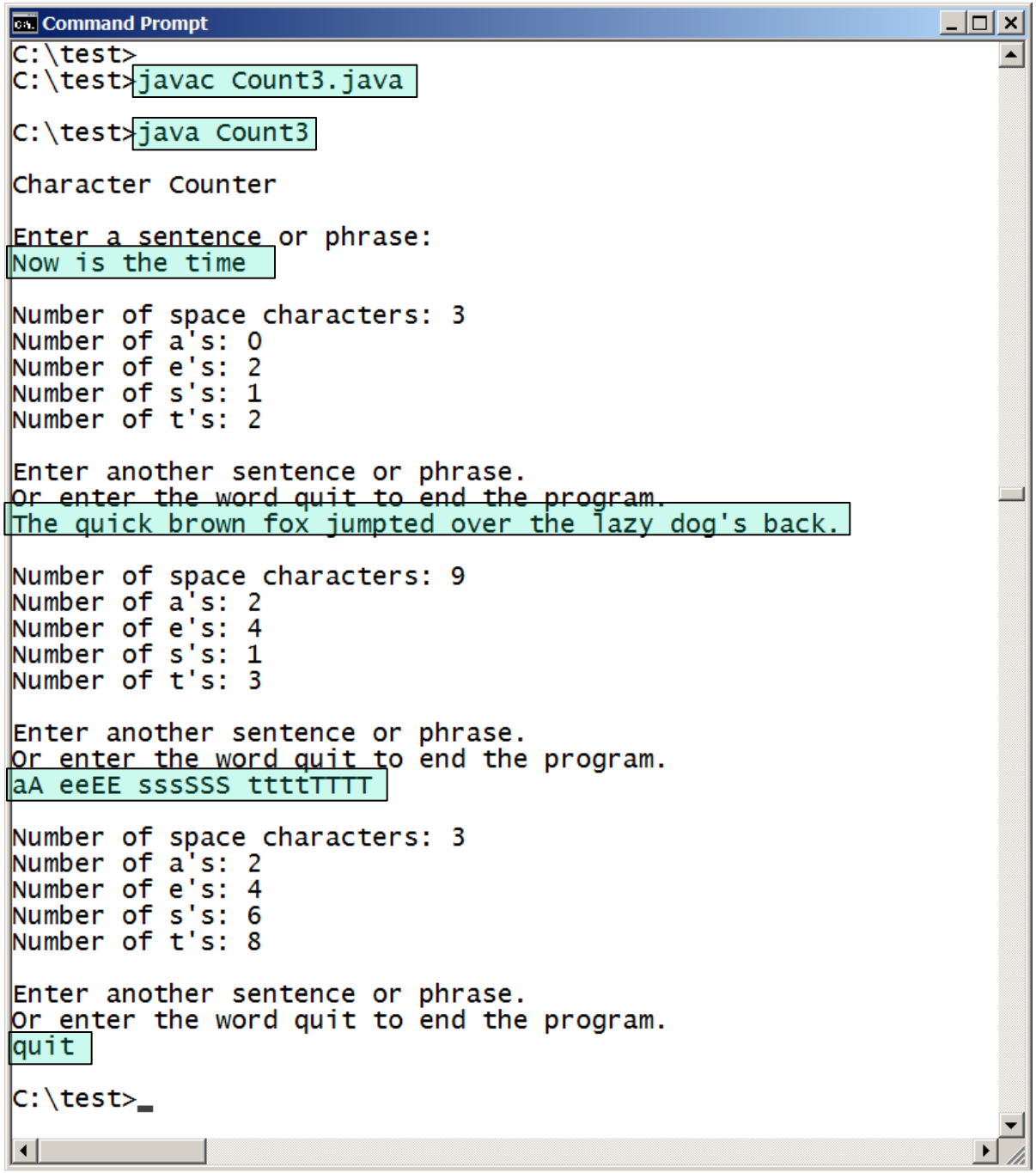
Add an outer loop that will continue to execute as long as the user does NOT enter the phrase *quit*.

Be sure to handle at least one phrase.

After the first iteration, tell the user to enter a phrase or *quit* to quit.

Make it more user-friendly

- Note that all of the initializations for the counts should be inside the new outer loop.
  - We want the counts to start over for each new phrase entered by the user.
- Be sure to go through the program and properly indent after adding code.
  - With nested loops the inner loop should be indented.


Command Prompt
C:\test>
C:\test>javac Count3.java
C:\test>java Count3
Character Counter
Enter a sentence or phrase:
Now is the time
Number of space characters: 3
Number of a's: 0
Number of e's: 2
Number of s's: 1
Number of t's: 2
Enter another sentence or phrase.
Or enter the word quit to end the program.
The quick brown fox jumpted over the lazy dog's back.
Number of space characters: 9
Number of a's: 2
Number of e's: 4
Number of s's: 1
Number of t's: 3
Enter another sentence or phrase.
Or enter the word quit to end the program.
aA eeEE sssSSS ttttTTTT
Number of space characters: 3
Number of a's: 2
Number of e's: 4
Number of s's: 6
Number of t's: 8
Enter another sentence or phrase.
Or enter the word quit to end the program.
quit
C:\test>

Program 2: Correcting a defect

- In the Downloads area of the class web site there is a defective version of the extremes program from Project 6.
- http://www.cse.usf.edu/~turnerr/Programming_Concepts/Downloads/Project_7/

- This program correctly computes the min, max, and average of numbers input by the user. But it does not automatically stop after 10 inputs. It continues asking for inputs until the user enters a 0.

Program 2: Correcting a defect

- Download defective_extremes.java
- Rename the file as corrected_extremes.java
- Change the class name to corrected_extremes.
- Correct the defect by adding code to stop the program after 10 inputs (if not stopped previously by a 0 input.)
- Use a logical AND in the condition for the do loop to exit the loop if either the user has entered a 0 or 10 numbers have been input and processed.

Program 2: Correcting a defect

- Compile and test the corrected program using data from the sample runs in the assignment for Project 6:

http://www.csee.usf.edu/~turnerr/Programming_Concepts/055_Project_6_Extremes.pdf

Submission

- Put both of your Java source files into a folder and zip it.
- Submit your zipped Java source files via Canvas Assignments.

- Project is due by 11:59 PM
  - Sunday, February 21 Sections 001 and 002
  - Monday, February 22 Sections 003 and 004

Ground Rules

- It is OK to *discuss* the project with other students BUT
  - Do not share your code with other students.
  - Before or after submitting the project.
- Do not copy any other student's code.
  - Or even look at it.
- Do not let anyone copy or examine your code.

Ground Rules

Except for code posted on the class web site

- Do not copy code from the Internet
  - or any other source (other than the textbook.)
- Do not ask for help on an Internet forum.
  - If you need help, ask your instructor or a TA.
  - Come to lab and help sessions.
- Write your own code.