Rochester Institute of Technology
National Technical Institute for the Deaf
Information and Computing Studies Department

Name: Edwad Riley

NACA.161 Programming Fundamentals II
Practice Exercise #30 – Splitting Strings

Overview

This exercise is designed to let you practice separating data from a string.

Part 1: Words and Spaces

1) Create a class called **Splitter** that contains a main method.

2) Create a loop that will keep getting words from the user until they enter an "X". Make sure you give a prompt that explains what to do. Compile and run the code to make sure this is working

3) Before the loop, open a file named **words.txt** and choose a class that will allow you to write a String at a time. Do not write a line at a time.

   What IO class did you choose? File

4) While inside the loop, write the word the user entered to the file followed by one space and write it to the file

   What does this statement look like?
   S.append(line); then S.append("_ ");

Compile the code before continuing.

5) After the loop, close the file.

   Compile and run the code to make sure you are creating the file correctly. You should have only one line with each word separated by a space from the last one.

6) Now open the file for reading, use a class that will allow you to read one line at a time.

   What class did you use? File Reader

   What was the argument to the constructor?
   FileReader fr = null;

   Compile your code.

7) Now read the line inside the file

What method did you use? line= pr.nextLine ( );

Compile the line.

8) Print the line that you read to the screen. Compile and run the code to make sure you read the file correctly before moving on.

9) Now split the line you read into pieces where the space character is used as the delimiter.

What did this code look like to split the string?
String[] tokens = line.split(" ");

10) Using a standard for-loop, display each word on a different line on the screen

What was your header of the for-loop?
for(int i = 0; i < tokens.length; i++) {

Compile and run to make sure your program works correctly.

11) Add another loop using the enhanced for-loop to display each word on a different line on the screen.

What was your for-loop?
for(String s2: tokens)

Compile and run to make sure it works.

Part 2: Numbers and Commas

12) Adding on to the program you have written, initialize a string with the value 10,20,30,40,50

13) Now split the line you read into pieces where the comma is used as the delimiter.

What did this code look like to split the string?
string [] nums = numString.split(",");

14) Using the enhanced for-loop, display each number on a different line on the screen
Compile and run the program to make it sure it works.

15) Now add each number together, and print the sum.
Compile and run the program to make sure it works.

Part 3: Numbers and Multiple Commas

16)Copy the code that you wrote for steps 12 to 15. Modify the initial value of the string to be 10,,20,,,30,40,,,,50.

Now split the string so that you get the same results as in step 15.

What did this code look like to split the string?

String[] NUMS2 = numString2.split(",");
____

Signoff

When you complete all of the steps successfully and answer all of the questions, contact your instructor to check if your application(s) executes correctly and to review your code. We will initial the line below.

________Successful execution of code

If you do not finish the program during the class period, contact your instructor to check to review your code and initial below.

________Code not completed during lab time

You may then submit your work at the start of next class. You may not use the work period of the next class to complete this assignment. If you do not have a