CS 280
Programming Language Concepts

About Assignment 1

About The Specification

- Part of “think first, then code” is making sure that you understand everything about the problem that you’re asked to solve
  - Is everything clear?
  - Is anything missing?
  - Are there things that are implied in the specification that I should verify?

What is the starting line length?

- Good question! It’s not in the specification.
  - In software development, you should assume nothing. That means don’t assume a number, and don’t assume that you should ask the user for a number (without checking with your customer first)
- Resolution: Default starting line length is 60

"Extra spaces are added in multiple places"

- The specification is unclear!
  - What does it actually mean to add spaces in multiple places? What places? How many spaces can I add in each place?
- Resolution:
  - Distribute the extra spaces that you have to add evenly between all the words.
    - Example: 4 words means there are 3 spaces between the words…
      - If you must add 3 spaces, add 1 between each word
      - If you must add 6 spaces, add 2 between each word
  - If you must add spaces between some words and not other words, randomize where you add them
    - This applies if you have 3 spaces to add in 5 places (an extra space in 3 out of 5 places).
    - It also applies if you have 3 spaces to add in 2 places (an extra space in one place and two extra spaces in the other place).

Can I end a paragraph if there isn't one to end?

- The specification says that a valid ".ll N" line causes a new paragraph to begin
  - … which of course implies that the previous paragraph must end, and a line be skipped, for the new paragraph to begin
- What do I do if I see a ".ll N" line, I have already ended the previous paragraph, and have not processed any input for the next paragraph? Do I skip another line?
- Resolution: If you have already performed the "end of previous paragraph" action, don't repeat it
- NOTE this also implies that having a ".ll N" line at the very beginning of the input does not skip a line

Can a word extend across more than two lines?

- What if a word is so long – or a line is so short – that breaking up a word into two pieces won't fit on two lines?
- Resolution: yes, you can break up a word as many times as you might need to

Given that last slide,
what if the line length is 1?

- That's a mess, isn't it?
- A word of length greater than 2 would need to be split up into pieces of size 2 (letter dash, letter dash) and so could never be output!
- Resolution: We will decide on a minimum line length of 10. Any attempt to set the line length to a value less than 10 should be ignored

Breaking up words

- What if the word that I am breaking up into pieces has a dash in it already? Should I try to break up the word at the location of the dash that is already there?
- Resolution: that's more complicated than I want this assignment to be ☺ so you don't have to handle that case

All these things seem to be implied

- Specifying "a line of the form .ll N" seems to imply that that is the entire line: the very first character has to be a dot, and there should be nothing else on the line
- Extra items on the .ll line (for example, a line that is ".ll N on we go") is an incorrect line that should be ignored
- Ignoring a poorly formed .ll line, or ignoring a line with a . at the beginning that is *not* a correctly formed .ll line, is done silently: no error messages are specified so none should be printed out
- Ignoring a poorly formed .ll line implies that you do not end the current paragraph
- Reaching end of file on input implies that you would end the current paragraph, then finish the program
- Resolution: yes, all of these statements are correct interpretations of the specification

The sample has 41 characters per line!

- Sometimes, the specification has an error ☺

You Got The Lyric Wrong!

- Mary's dress doesn't sway. It waves.
- Resolution: yeah, my mistake. I have been singing it wrong in the shower for a few decades now…

Steps to a solution

- What does the program have to do?
  - collect up words from input
  - keep track of the desired length of the output lines
  - decide when an output line is ready
  - decide how many extra spaces are needed
  - decide where spaces go
  - decide when to add a hyphen
  - process lines that begin with a dot
  - process .ll directives