

# CS 249: Assignment 03

---

## Programming Assignments (95%)

### Narrator.java

Create a java file with a public class Narrator. The purpose of this class is to generate narration cards (each of which hold at most 4 lines of text). Each card is surrounded by a boundary character, and all text is centered. Cards have a maximum width of **50 characters**. It will have the following public methods (note that some of these are NOT directly used SilentFilm):

- **public Narrator(String [] lines, char boundaryChar)**
  - o Stores the lines and boundary character
  - o WARNING: when storing the lines, remember to:
    - Reallocate the instance variable
    - Copy the individual array values
  - o **Do NOT just do:** this.lines = lines
    - I would suggest calling setLines() here
- **public char getBoundaryChar()**
  - o Returns the boundary character
- **public String getLines()**
  - o Returns a SINGLE String that concatenates the lines, with a newline "\n" at the end of each line.
- **public void setBoundaryChar(char boundaryChar)**
  - o Stores the boundary character
- **public void setLines(String [] lines)**
  - o Stores the lines, BUT remember to:
    - Reallocate the instance variable
    - Copy the individual array values
  - o **Do NOT just do:** this.lines = lines
- **public String generateBoundaryLine()**
  - o Returns a String with 50 boundary characters followed by a newline
- **public String generateCenteredLine(String text)**
  - o Returns a String with a CENTERED line of text flanked by the boundary character, ending in a newline.
  - o You can assume that the text will never be too large to fit (i.e., never over 48 characters).
  - o To do this:
    - Start with an empty String (or you can use StringBuilder)
    - Append a boundary character

- Compute how many total spaces will be needed:  $50 - (\text{length of text}) - 2$
  - Get half of the total number of spaces (using integer division)
  - Get the second half of spaces by:  $(\text{total spaces}) - (\text{first set of spaces})$
  - Append the first half of spaces
  - Append the text
  - Append the second half of spaces
  - Append the boundary character
  - Append a newline
  - Return the String
- **public String toString()**
  - This **returns a String** with a set of narration cards.
  - **NOTE: This function does NOT print anything out! In other words, DON'T use System.out.println here!!!**
  - Each card will have the following dimensions:
    - 50 characters in width
    - 8 lines in height
  - Start with an empty String (or you can use String builder)
  - For every 4 lines:
    - Append a boundary line
    - Append a centered line with an empty String (basically a blank line flanked by boundary characters)
    - Compute how many lines are left at this point
    - If the number of lines left are less than 4:
      - Line count will be however many lines are left
      - Extra line count will be  $4 - (\text{lines left})$
    - Otherwise:
      - Line count will be 4
      - Extra line count will be zero
    - Append (line count) number of lines from your array of lines
    - Append (extra line count) number of blank lines
    - Append a single blank line regardless
    - Append a boundary line
  - Return the single String containing all of these narration cards

## SilentFilm.java

The purpose of this program is to ask the user for information for their Narrator, and then print out the final set of cards. Create a class SilentFilm, and add these methods (both are public and static):

- **public static Narrator askForNarration(Scanner input)**
  - **WARNING: Scanner input has ALREADY been created! Do NOT create a new one here!**
  - Print "**Enter boundary character:**" using System.out.println().
  - Get the boundary character as the first character of the nextLine() from the Scanner object.
    - NOTE: You may assume the user will NOT enter an empty line.
  - Print "**Enter number of lines:**" using System.out.println().
  - Get the number of lines by:
    - Reading in the next String LINE using nextLine()
    - Using Integer.parseInt() to convert this line to an integer
  - Create a String array with the appropriate number of Strings (*allLines*).
  - Print "**Enter lines:**" using System.out.println().
  - In a loop, read in the correct number of lines from the user using nextLine() on the Scanner object and store each line in the String array *allLines*.
  - Create a new Narrator object using *allLines*, and boundary character.
  - Return the newly-created Narrator object.
- **public static void main(String [] args)**
  - Create a Scanner object to read from System.in.
  - Create a Narrator object using askForNarrationData(), remembering to grab what the method returns and putting it into a variable *n*.
  - Print "**Our story begins...**" using System.out.println().
  - Print out the narration cards using: System.out.println(n)

Example Runs (user input highlighted in blue):

<pre>Enter boundary character: * Enter number of lines: 0 Enter lines: Our story begins...</pre>
<pre>Enter boundary character: . Enter number of lines: 1 Enter lines:  Our story begins... ..... . . . . . . .....</pre>
<pre>Enter boundary character: ! Enter number of lines: 1 Enter lines: ... Our story begins... !! ! ! ! ! ! ! !!</pre>
<pre>Enter boundary character: % Enter number of lines: 4 Enter lines: Somebody Once told me The world was gonna roll me I ain't the sharpest tool in the shed... Our story begins... %% % %                Somebody                % %            Once told me                % %        The world was gonna roll me        % %    I ain't the sharpest tool in the shed...    % % %%</pre>

Enter boundary character:

\$

Enter number of lines:

5

Enter lines:

A long time ago

in a galaxy far, far away...

#### STAR WARS

Our story begins...

\$

\$ \$

\$ A long time ago \$

\$ in a galaxy far, far away... \$

\$ \$

\$ \$

\$ \$

\$

\$

\$ \$

\$ STAR WARS \$

\$ \$

\$ \$

\$ \$

\$ \$

\$

Enter boundary character:

@

Enter number of lines:

8

Enter lines:

Space: the final frontier

These are the voyages of the starship

Enterprise

Its five-year mission:

to explore strange new worlds;

to seek out new life and new civilizations;

to boldly go where no man has gone before!

Our story begins...

@@

@ @

@ Space: the final frontier @

@ @

@ These are the voyages of the starship @

@ Enterprise @

@ @

@@

@@

@ @

@ Its five-year mission: @

@ to explore strange new worlds; @

@ to seek out new life and new civilizations; @

@ to boldly go where no man has gone before! @

@ @

@@

## Testing Screenshot (5%)

Submit a screenshot showing the results of running the test program(s).

## Grading

Your OVERALL assignment grade is weighted as follows:

- 5% - Testing results screenshot
- 95% - Programming assignments

For the **PROGRAMMING** portion of the assignment, in addition to the usual penalties:

<i><b>Issue</b></i>	<i><b>Penalty (in %)</b></i>
Narrator.java missing / not implemented	70
SilentFilm.java missing / not implemented	30
Narrator.java not properly implemented	35
SilentFilm.java not properly implemented	15