# Introduction

A challenge was posed to create a 2D array and fill it with asterisks, so as to create side-by-side images of a triangle and a wine glass. The array was to be filled dynamically, and not hard-coded. This program will solve this problem, and fulfill all of the following functional requirements:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** |  | **Functional Requirement** |  | **Value** | **Stakeholder** |
| FID001 | **I want to** | create a 2D array | **so that** | I can use the 2D array to store data | student |
| FID002 | **I want to** | dynamically fill the array | **so that** | the program fulfills the requirements of the assignment | student |
| FID003 | **I want to** | fill the array with asterisks and spaces | **so that** | A 2D image will be created | student |
| FID004 | **I want to** | print the 2D array to the console | **so that** | the instructor can see that the program works properly | instructor |

The introduction will flow into the analysis and provide a suitable(e.g., professional) transition for the reader.

# Analysis

Since we knew that only single characters would make up each element of the array, the most efficient way to fulfill FID001 was to use *char[][]*to create the array, and set the size to 8 by 16. Since FID002 required that the array be filled dynamically, conditionals were used, specifically *for* and *if-else* loops. The rows of the array were iterated through, and logic was used to fill each column with asterisks at the appropriate indexes, as per FID003. Finally, *System.out.print()* was used in tandem with *System.out.println()* to print the image to the console, thus satisfying FID004.

# Conclusion

I’m confident that you will find that the program sufficiently solves the problem set forth. Upon running MainEntry.java, a crude image of a triangle and a wine glass made up of asterisks will be printed to the console.

I researched and tinkered for a good 12 hours total, over 4 days, trying to figure out how to complete this assignment, and wasn’t even close. I’m glad that we ended up doing it in class, although it’s unfortunate that it was necessary. I hope we all get up to speed quickly. I, for one, am determined to improve. The repository can be found at:

<https://github.com/cellson7170/ce_wk2_2dLinearDataSet>