



CMSC 131

Object-Oriented Programming I

Marquee Project Notes

**Dept of Computer Science
University of Maryland College Park**

This material is based on material provided by Ben Bederson, Bonnie Dorr,
Fawzi Emad, David Mount, Jan Plane

Task

- Implement the class MarqueeDataManager that implements the DataManager interface
- The DataManager implements a single method (step). This represents the array you need to return for each animation step
- Feel free to add as many methods you understand you need
- There are different approaches to implement the project; as long as you pass the public and release tests, you are OK 😊
- **Remember you may NOT use ArrayList in this project**

Implementation

- Create a MarqueeDataManager class that extends the DataManager interface. The MarqueeDataManager class must import:

```
import java.awt.Color;  
import cmscMarqueeLib.*;
```

Methods

- There are a set of auxiliary methods that will help you in this project
- **append** method
 - `Cell[][] appendArrays(Cell[][] first, Cell[][] second)`
 - It will return an array with all the rows of the first array appended to the second. You can assume both arrays have the same number of rows
- **generateEmptyCellsArray** method
 - This method returns a `Cell[][]` with cells that have the empty Color (`Color.WHITE` for this project)

Methods

- **Convert 2-Dim array of integers to 2-Dim array of Cells method**
 - In order to convert a message to a 2-Dim array of cells you first need to convert each character to a 2-Dim array of integers using the method **Alphabet.toIntArray** we provided. You should define a method that takes the 2-Dim array of integers returned by **Alphabet.toIntArray** and then returns a 2-Dim array of Cells where zeros are mapped to cells with a white color and ones are mapped to Cells with a red color
- **Convert original message to Cells array method**
 - Using the methods described earlier you can now generate a 2-Dim array of Cells for the provided message
 - For each character of the message generate a 2-Dim array of cells, and append all the arrays generated
 - Remember that you need to add empty columns of Cells between each Cell array that represents a character

Shifting Over Array

- The MarqueeDataManager constructor must create a 2-Dim array of Cell objects (let's calling it the shiftingOverArray) from which we can return a subarray each time the **step** method is called. **Remember you don't call the step method; the code we provided will call the step method**
- There are several approaches to define the shiftingOverArray array. One approach is by breaking down the shiftingOverArray in three parts:
 - **First part** → Array of empty (white color) cells that has as width the marquee's width
 - **Second part** → 2-Dim array of Cell objects that has the message (with empty columns between letters) to display
 - **Third part** → Array of empty (white color) cells that has as width the marquee's width
- The first window will start at index 0 of the shiftingOverArray and end at index marqueeWidth - 1

About step

- Each time step is called a subarray from the shiftingOverArray should be returned. Defining a getSubArray method can help you
- Notice that when the step method is called it returns the current window and sets variables that define the next window that will be defined

Other

- Configuration values (e.g., CHARACTER_WIDTH, CHARACTER_HEIGHT) can be found in the cmscMarqueeLib package, ConfigValues class
- Notice that if a message has spaces there is nothing special you have to do. Treat the space as a normal character (Alphabet.toIntArray will return the correct array and you will add column space as if it were a letter).