**CPSC 231: Computer Science II**

**Programming Mastery Project 1:**

# Java for Fun and ~~Profit~~ Art

## Why?

This assignment will ensure you have a good grasp of the fundamentals of the Java programming language - both its limitations and advantages over other languages such as Python. Everything we do in this class will build on these concepts. We will also refactor this assignment in the coming weeks to demonstrate how an OOP approach can make this program more extensible and easier to use.

## The Assignment

If you're not yet comfortable with Java loops, reference types, methods, and 2D arrays - buckle up! This assignment will test your command of the Java programming fundamentals we've learned in class and be our only assignment that does not require an OOP solution.

You will write a program that can be used to draw exciting ascii art in the terminal. You should represent a canvas as a 2-dimensional char array and allow a user to add characters to the canvas using a standard x and y coordinate system.

For this assignment, you will be given a starter code source file with skeletons for the following methods:

**createCanvas**

This method will take in two integers for the width and height of the canvas and return a two-dimensional char array of the specified width and height after initializing each item in the array.

Your canvas should have a decorative border where all corners contain "+", all vertical edges contain '|', and all horizontal edges contain '='. All of the cells in the middle of the canvas should be initialized to contain a space ' '.

For example, calling `createCanvas(10, 5)` would return a char array that looks like this:

```
4 + = = = = = = = = +
3 |                   |
2 |                   |
1 |                   |
0 + = = = = = = = = +
  0 1 2 3 4 5 6 7 8 9
```

*Note: The axis labels above come from the printCanvas method and are included to help you visualize the size of the 2D array. They should not be part of your array.*

**addCharacter**
This method will take in a reference to your canvas, the character you want to paint to the canvas, and the x and y coordinates of where you'd like to place that character. It should modify the canvas array in memory by changing the cell at the specified X and Y coordinates. It should not return anything.

For example, adding the character '(' to the canvas created above at position (4, 2) and the character ':' to the position (5,2) would look like this:

```
4 + = = = = = = = = +
3 |                   |
2 |       ( :         |
1 |                   |
0 + = = = = = = = = +
  0 1 2 3 4 5 6 7 8 9
```

**printCanvas**
This is the method you will use to produce the example outputs above. It will already be implemented in the starter code file, and you should not modify it.

This method loops through the array from top to bottom, left to right, printing out each cell. You will need to adjust the way you set up and index your array to fit this output format.

**main**
Your standard Java main method, which should create a new canvas, add

a few characters to the canvas, and then print the finished artwork. You can be as creative here as you want - feel free to peruse the ASCII art archive for inspiration. You do *not* need to get any user input within this method.

## Getting Started

Not sure where to begin? Download the starter code source file, then play around with your printCanvas method using a char array of hard-coded values. After you understand how that works, move on to the createCanvas method to ensure setting up the borders works correctly. After that, you can move on to your addCharacter method with confidence.

## Submission & Grading

Your deliverable is **one .java file named Drawing.java and a README** including the required information listed in the Chapman University Coding Standards.

Assignments will be submitted through Gradescope. There will be a link to the Gradescope submission associated with each assignment on Canvas, and you will have unlimited attempts to submit your code before the due date. The implementation and functionality of your code will be graded by an auto-grader, and you will be able to see the results immediately after submitting your code. Your assignment will also be manually graded for adhering to style guidelines and including meaningful comments that document your code.

## Citing Outside Sources

You are allowed to use small, isolated lines of code from external sources in your programming assignments as long as they are appropriately cited. Any time you are including code that you did not write yourself, it must be cited. This includes sources from StackOverflow and similar forums, current or previous students, tutors, books, tutorials, ChatGPT, etc. You should wrap the copied code in a comment denoting the start and end of said code, like this:

```
/* BEGIN CODE FROM SOURCE: link/name of source */
  . . . code you did not write . . .
/* END OF CODE FROM SOURCE: link/name of source */
```