

# CS 249: Assignment 04

---

## UML Diagram (10%)

You will submit a diagram for the GameBoard class.

You do NOT need to diagram the Box, DraftingTable, TextBox, or TestGameBoard classes.

## Programming Assignments (85%)

### GameBoard.java

Create a java file with a public class GameBoard. The purpose of this class is to store a 2D array of characters, as well as a default/fill character. It is up to you whether to store the row and column count explicitly. This class will have the following public methods:

- **public GameBoard(int rowCnt, int colCnt, char fillChar)**
  - o Allocate space for your 2D character array instance variable with *rowCnt* rows and *colCnt* columns.
  - o Loop through this array and fill it with the *fillChar*.
  - o Be sure to save the *fillChar* to your instance data.
  - o Also, if you decided to store the row and column counts explicitly, be sure to save that information here as well.
- **public void clear()**
  - o Loop through your 2D character array and fill it with your fill character.
  - o The idea here is that someone may have drawn on this GameBoard, and you are setting it back to a blank canvas (e.g., the fill character).
- **public int getRowCnt()**
  - o Returns number of rows.
- **public int getColCnt()**
  - o Returns number of columns.
- **public boolean isValidPosition(int row, int col)**
  - o Returns true if the given row and column is within the bounds of the GameBoard.
  - o PLEASE remember that you need to check that these values are GREATER THAN OR EQUAL TO ZERO **AND** STRICTLY LESS THAN whatever size is relevant (i.e., rowCnt for row and colCnt for col).
- **public char getPos(int row, int col)**
  - o If the provided row and column is a valid position, return the character at that position.
  - o Otherwise, return a space character ' '.

- **public boolean setPos(int row, int col, char c)**
  - If the provided row and column is a valid position:
    - Set the character value at that position to c.
    - Return true.
  - Otherwise:
    - Return false.
- **public String toString()**
  - Returns a String with: rowCnt + " x " + colCnt + " GameBoard (default: " + fillChar + ")"
  - *Example:* if the rowCnt = 5 and the colCnt = 4 and the fillChar = '\$'
    - "5 x 4 GameBoard (default: \$)"
- **public String getBoardString()**
  - Returns a String with the data from the 2D array.
  - PLEASE be sure to put a newline "\n" at the end of each row!
  - *Example:* if the rowCnt = 5 and the colCnt = 4 and the fillChar = '\$' and someone set the character at (2,3) to be '@':
    - \$\$\$\$  
 \$\$\$\$  
 \$\$\$@  
 \$\$\$\$  
 \$\$\$\$

## Box.java

Create a java file with a public class Box. The purpose of this class is to store an axis-aligned box (with a starting row and column and an ending row and column). This box may be drawn on a GameBoard with a specified drawing character. The box may also be filled or unfilled; if unfilled, ONLY the perimeter is drawn. As it happens, this class is **immutable**. This class will have the following public methods:

- **public Box(int sr, int sc, int er, int ec, boolean filled, char drawChar)**
  - Store the starting row, starting column, ending row, ending column, whether the box is filled or not, and what character to draw on the box.
  - You may assume the user has provided  $sr \leq er$  and  $sc \leq ec$ .
- **public String toString()**
  - *Example:* let's say we have a Box that starts at (1,2) and ends at (3,4) (row,col), with a drawing character of '#':
    - If filled:
      - Return "Filled Box from (1,2) to (3,4) with char '#'"
    - Otherwise:
      - Return "Box from (1,2) to (3,4) with char '#'"

- **public void draw(GameBoard s)**
  - Draw the box onto the provided GameBoard s. Remember you have the setPos() method from GameBoard.
  - NOTE: If a FILLED Box goes from (1,2) to (3,4), rows 1 to 3 INCLUSIVE and columns 2 to 4 INCLUSIVE should be filled in with the drawing character.
  - ANOTHER NOTE: If the Box is NOT filled, the only positions that should be set to the drawing character are those that are:
    - In range [start row, end row] AND [start col, end col]
    - AND
      - Have row == start row
      - OR
      - Have row == end row
      - OR
      - Have col == start col
      - OR
      - Have col == end col

### DraftingTable.java

I have provided a runnable program for this assignment. Unlike previous assignments, there are two major differences with this program:

1. This SHOULD run using the application → run task WITHOUT the tests passing.
2. There is no test file for this program.

Your goal, therefore, is to get the tests for Box and GameBoard to pass. However, this program should help you debug matters.

**WARNING: Make sure that your SITNET ID does NOT appear in the .gitignore file!** Sometimes after changing the package name, IntelliJ will update .gitignore as well.

### 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
- 10% - UML Diagram
- 85% - Programming assignments