

# Data Structures

## Color Game WQU - 10 Course Points

The purpose of this lab is to understand how constructors initialize an instance of an object, as well as gain experience testing code.

**Start your assignment early!** You need time to understand the assignment and to answer the many questions that will arise as you read the description and the code provided.

Check our [Programming Assignment FAQ](#) for questions about VSCode, Java, the terminal, and more.

The assignment has one component, which is coding submitted through [Autolab](#).

## Overview

The ColorGameWQU class contains the find() and union() methods for a Weighted Quick Union implementation. You are tasked with writing the constructor for this implementation, so that the find() and union() methods properly work.

This Weighted Quick Union implementation is used to activate the Driver, which allows you to play the color memorization game Simon (also known as Genius). By unioning all pixels that are the same color (red, green, yellow, or blue), it can then use find() in order to determine what color you have selected when you click.

## Directions

- DO NOT add new import statements.
- DO NOT change any of the method's signatures.

You will work on the ColorGameWQU.java class, and implement its constructor.

The constructor is simply a method, which is only called when a new object of that class type is created.

- i.e "ClassName temp = new ClassName()" calls the public ClassName() constructor. This constructor is expected to initialize class attributes and do any other work that is needed for the "temp" variable to be used.
- Constructors can take in parameters, just like a method. You can also create multiple constructors for a single class, with different parameters in each. You do not need to create any additional constructors for this lab.
- For example, the constructor for the "Integer" (the wrapper class for int) has two constructors. The first takes in an int and sets the Integers value to that. And the second takes in a String, converts it to an int, and then sets the Integers value to that.

You are provided with a completed `Coordinate.java` class, which represents a single (row, col) coordinate. When using 2D arrays, a Coordinate at (4,3) would be `array[4][3]`. On an XY Grid, this would be equivalent to (3,4) (since row represents height, and col represents width)

To complete the `ColorGameWQU` constructor:

- 1) First instantiate the "parent" and "size" arrays. These are already declared for you as class attributes (aka fields).
  - To access and modify these class attributes, use "this.parent" and "this.size". While the "this." prefix is not always required, it is good practice to always use it when referring to class attributes

Set both arrays equal to a new array of their corresponding array types:

- The "parent" 2D array is of type "`Coordinate[][]`"
- The "size" 2D array is of type "`int[][]`", it is NOT "`Integer[][]`"
- The size of both arrays is `[row][col]`, which is given by the constructor parameters

- 2) Finally, set the default values for both arrays:

- You can accomplish this using only a double for-loop (nested), since the height (number of rows) and width (number of cols) of the two arrays are the same.
- For each index in the arrays:
  - Set the parent array at that index equal to a new `Coordinate`. Pass in the (row,col) values to the `Coordinate` constructor. This sets the parent of each cell to be itself by default.
  - Set the size array at that same index equal to 1, since it is its own parent.

Once you have initialized the class attribute arrays, and set their default values, the `Color Memorization Game` should be functional when you compile and run the `ColorGameWQU` class.

## Testing Your Code

Since this WQU implementation contains public methods, and represents a generalized 2D grid of coordinates, we can test it independently of the driver using a test class.

In the main project folder, there exists a "tests" folder next to the "src" folder. This contains JUnit tests, which can run and test pieces of your code.

To compile and run these tests, you must install the following JUnit package:

- [Test Runner for Java](#)

You are provided with a prewritten JUnit test class for `ColorGameWQU`. You can use this to test your constructors code.

See the [Debugging Guide](#) to learn more about debugging code.

## VSCode Extensions

You can install VSCode extension packs for Java. Take a look at [this tutorial](#). We suggest:

- [Extension Pack for Java](#)
- [Project Manager for Java](#)
- [Debugger for Java](#)
- [Test Runner for Java](#)

## Importing VSCode Project

1. Download ColorGameWQU.zip from [Autolab Attachments](#).
2. Unzip the file by double-clicking.
3. Open VSCode
  - Import the folder to a workspace through **File > Open Folder**

## Executing and Debugging

- You can run your program through VSCode or you can use the Terminal to compile and execute. We suggest running through VSCode because it will give you the option to debug.
- [How to debug your code](#)
- If you choose the Terminal:
  - first navigate to **ColorGameWQU** directory/folder
    - to compile: **javac -d bin src/wqu/\*.java**
    - to execute: **java -cp bin wqu.ColorGameWQU**

## Before submission

**COMMENT all printing statements you have written from ColorGameWQU.java**

## Collaboration policy.

Read our collaboration policy [here](#).

**Submitting the assignment.** Submit *ColorGameWQU.java* separately via the web submission system called Autolab. To do this, click the *Labs and Assignments* link from the course website; click the *Submit* link for that assignment.

## Getting help

If anything is unclear, don't hesitate to drop by office hours or post a question on Piazza.

- Find instructors office hours [here](#)
- Find tutors office hours on Canvas -> Tutoring
- Find head TAs office hours [here](#)
- In addition to office hours we have the [Coding and Social Lounge \(CSL\)](#) , a community space staffed with lab assistants which are undergraduate students further along the CS major to answer questions.

**By Colin Sullivan**