CMPS 5J
Introduction to Programming in Java
Programming Assignment 5 (100 Points)

In this assignment you will write a Processing program that uses loops (and possibly functions) to emulate a program posted on the course webpage at

https://classes.soe.ucsc.edu/cmps005j/Fall15/Examples/MouseInBox.html

The above link takes you to a Processing program that runs in your browser. Again no source code is provided, and your task is to reverse-engineer the program.

The program draws a 25 by 25 grid of square boxes in a window that is 500 pixels wide and 500 pixels high. Each square box is therefore 20 pixels on a side. This grid is easily drawn by means of a pair of nested (for or while) loops. Good examples to study for this grid are ColorLoop2, ColorLoop3, and Shade, all posted in the examples section of the webpage.

Observe that when you press the mouse inside one of the boxes, a pattern appears in which some previously white boxes are colored black.

![Pattern Illustration](image)

This pattern includes the box containing the mouse pointer. When you release the mouse, the pattern remains. You can create the pattern anywhere on the grid by pressing the mouse in any box. If you hold down the mouse and drag across the window, the pattern is replicated and a swath of boxes is colored black. To erase the image and return all boxes to their original white state, press any key.

Your program should have the following general form.

```java
// userid
// cmps 5J
// pa5

// global variable declarations

void setup(){
    // initialization commands
    // draw the initial grid here using nested loops
}

void draw(){
    // commands that are executed once per frame
    // write another pair of nested loops that colors
    // certain squares black if the mousePressed system variable is true
}

void keyPressed(){
    // commands that are executed once per keyPress
    // re-draw the initial grid all white
}
```
// optional function
boolean mouseInBox(int i, int j){
    // return true if and only if the mouse is in the
    // square with upper left corner at (20*i,20*j)
}

// optional function
void drawPattern(int i, int j){
    // draw pattern centered at the box with upper left corner
    // at (20*i,20*j)
}

The above optional functions are recommended, but not necessary for full credit. As always start early, formulate questions and get help. Call your program MouseInBox. Submit MouseInBox.pde as an attachment to pa5 in eCommons.