Homework Turnin

Email:	rgalanos@fcps.edu
Section:	6G
Course:	TJHSST APCS 2016-17
Assignment:	02-04
Receipt ID:	53b119918b56d14407c7fa461bf43352

Turnin Successful!

The following file(s) were received:

AreaFill.java (2884 bytes)

```
//name:
//date:

import java.util.Scanner;
import java.io.*;
public class AreaFill
{
   public static char[][] grid = null;
   public static String filename = null;

   public static void main(String[] args) throws FileNotFoundException
   {
      Scanner sc = new Scanner(System.in);
      while(true)
      {
         System.out.print("Filename: ");
         filename = sc.next();
         if(filename.equals("-1"))
         {
            sc.close();
            System.out.println("Good-bye");
            System.exit(0);
         }
         grid = read(filename);
         System.out.println( display(grid) );
         System.out.print("\nEnter ROW COL to fill from: ");
         int row = sc.nextInt();
         int col = sc.nextInt();
         grid = fill(grid, row, col, grid[row][col]);
         System.out.println( display(grid) );
          //  Extension
      // int count = fillAndCount(grid, row, col, grid[row][col]);
      // System.out.println( display(grid) );
      // System.out.println("count = " + count);
         System.out.println();
      }
   }
   public static char[][] read(String filename)throws FileNotFoundException
   {
      Scanner infile = new Scanner(new File(filename));
      int rows = infile.nextInt();
      int columns = infile.nextInt();
      char[][] chars = new char[rows][columns];
      infile.nextLine();

      for(int i=0;i<rows;i++)
      {
         String line = infile.nextLine();
         for(int a=0;a<columns;a++)
         {
            chars[i][a] = line.charAt(a);
```

```
            }
        }

        return chars;
    }

    public static String display(char[][] g)
    {
        String contents = "";
        for(int i=0;i<g.length;i++)
        {
            for(int a=0;a<g[0].length;a++)
            {
                contents+=g[i][a];
            }
            contents+="\n";
        }
        return contents;
    }
    public static char[][] fill(char[][] g, int r, int c, char ch) //recursive method
    {
        if(r<0||c<0||r>g.length-1||c>g[0].length-1)
        {
            return g;
        }
        else
        {
            if(r<g.length-1)
            {
                if(g[r+1][c]==ch){
                    g[r+1][c] = '*';
                    fill(g, r+1, c, ch);
                }
            }
            if(r>0)
            {
                if(g[r-1][c]==ch){
                    g[r-1][c] = '*';
                    fill(g, r-1, c, ch);
                }
            }
            if(c<g[0].length-1)
            {
                if(g[r][c+1]==ch){
                    g[r][c+1] = '*';
                    fill(g, r, c+1, ch);
                }
            }
            if(c>0)
            {
                if(g[r][c-1]==ch){
                    g[r][c-1] = '*';
                    fill(g, r, c-1, ch);
                }
            }
        }
        return g;
    }

    // Extension-- count and return the number of asterisks
    public static int fillAndCount(char[][] g, int r, int c, char ch)
    {
        return 0;
    }
}
```