Colin Kincaid
CS 106A

Final Exam
August 17, 2018

CS 106A Final Exam – Answer Booklet

This is an closed-textbook, closed-note exam. You may use two double-sided sheets of notes, but otherwise you may not use any printed paper resources or computing devices of any kind, including calculators, laptops, cell phones, or other devices.

Unless otherwise indicated, you will be graded on functionality only – but good style helps graders understand what you were attempting. **Remember that you must write the solution to a problem in that problem's provided answer pages in the answer booklet. Answers to a problem outside of its designated answer pages in the answer booklet will receive no credit.** You do not need to write any import statements. Unless otherwise specified, you may write helper methods to implement required behavior if you like. Do not abbreviate code; you must write out all code that is part of your answer. Note whether a question requires you to write a complete Java program, or just individual methods.

You have 3 hours to complete this exam. We recommend looking over all problems before starting. **Remember that you do not have to complete problems in order.**

First Name: ______________________

Last Name: ______________________

SUNET ID (e.g. jdoe) ______________________

Section Leader / Grader: ______________________

I accept the letter and spirit of the Honor Code. I have never given nor received any unpermitted aid on this exam, and promise to address/report anyone who fails to follow the Honor Code. I also promise to write more neatly than I have in my entire life.

(sign here) ______________________

```
// don't worry about importing packages

public class FourInARowBoard extends GCanvas {

    private static final int DIAMETER = 100;
    private static final Color EMPTY = Color.WHITE;

    private GOval[][] pieces;

    /* === Board creation === */

    public FourInARowBoard(int nRows, int nCols) {
        pieces = createPieces(nRows, nCols);
        displayBoard();
    }

    private GOval[][] createPieces(int nRows, int nCols) {
        // Assume this returns a 2D array of GOvals with the EMPTY color
    }

    private void displayBoard() {
        // Assume this displays a graphical representation of the board
        // Note that this method is responsible for adding all of the
        //      GOvals in the pieces array to the board.
        // (You can see this program in action when solutions are posted)
    }
```

```
/* === Making moves === */
```

Part A

```
    public String validateMove(int row, int col) {
        // TODO: your code for Part A here!









    }
```

Part B

```
    public void playPiece(int row, int col, Color player) {
        // TODO: your code for Part B here!

    }
```

```
/* === Ending the game === */

    public boolean gameOver() {
        return isDraw() || diagonalWin();
    }

    private boolean isDraw() {
        // Assume this returns true when no more moves can be made
    }
```

Part D

```
    private boolean diagonalWin() {
        // TODO: your code for Part D here!















    }
}
```

```
public class FourInARowGame extends ConsoleProgram {

    private FourInARowBoard board;

    public void init() {
        int nRows;
        int nCols;
        do {
            nRows = readInt("How many rows should the board have? ");
        } while (nRows < 4);
        do {
            nCols = readInt("How many columns should the board have? ");
        } while (nCols < 4);
        board = new FourInARowBoard(nRows, nCols);
        add(board);
    }

    public void run() {
        while (!board.gameOver()) {
            takeTurn(Color.RED);
            if (!board.gameOver()) {
                takeTurn(Color.BLACK);
            }
        }
        println("Game over!");
    }
```

Part C

```
    private void takeTurn(Color player) {
        // TODO: your code for Part C here!











    }
}
```

Problem 2, Part A

Problem 2, Part B

Problem 2, Part C

1.

2.

3.

Scratch Paper for Problem 2

Problem 3

```
// don't worry about importing packages

public class FurnitureCatalog {

    // TODO any instance variables you need go here









    /* Creates a new furniture catalog
     * given a file of initial items and their prices.
     */
    public FurnitureCatalog(String filename) {
        // TODO implement this constructor
    }

    /* Returns a list of all the prices in this catalog for the given item.
     */
    public ArrayList<Double> getPriceOptions(String furnitureItem) {
        // TODO implement this method

    }

    /* Returns a list of all the items that only appear once in this catalog.
     */
    public ArrayList<String> getMustBuys() {
        // TODO implement this method
    }

    /* Adds an item and its price to this catalog. The item may be new,
     * or it may already be in the catalog. Does nothing if the catalog
     * already has an entry with this price for the given item.
     */
    public void addItem(String item, double price) {
        // TODO implement this method









    }
}
```

Problem 4

```
// don't worry about importing packages

public class BakerRank extends GraphicsProgram {

      private static final String[] BAKERS = {/* fixed-length array of baker
names will be provided here (no action required) */};
      private static final int DELAY = 1000;

      // TODO any instance variables you need go here










      public void init() {
            // TODO implement this method























      }
```

```
public void run() {
      // TODO implement this method











}
```

```
    // TODO write method to respond to button pressing here
```

```
}
```