Electronics and
Computer Science

UNIVERSITY OF
Southampton

Further Assessment

Module:	COMP1206: Programming 2
Assignment:	Further Assessment
Lecturer:	Dr Sebastian Stein (ss2@ecs.soton.ac.uk)
Deadline:	2nd Oct 2020
Feedback:	1st Nov 2020

Instructions

The Further Assessment consists of three separate components, each corresponding to one of the original assessments on the module (the coursework, the labs and the CFA):

	Original Assessment	Further Assessment
1)	Coursework	Hitori Application
2)	Labs	JUnit and C Exercises
3)	CFA	Blackboard CAA

These components are detailed on the following pages.

IMPORTANT: You only need to complete those components that you did not pass on your first attempt (or that you did not submit anything for), i.e., those where you original mark was less than 40%.

For example, if you received 50% on the coursework, 25% across all labs and did not submit the CFA, you should only complete further assessments 2 (JUnit and C Labs) and 3 (Blackboard CAA).

If you failed or did not submit some of the labs, but still achieved an average of 40% or more across all labs, then you do not need to complete further assessment 2.

You must pass each component that you are required to take (i.e., achieve at least 40% of the available marks on it).

If you are in doubt about which components you should complete, please contact the ECS student office at ecs-studentoffice@soton.ac.uk

If you have questions about the assessment instructions, please contact Seb Stein at ss2@ecs.soton.ac.uk

Further Assessment 1: Hitori Application

Hitori is a puzzle game where the player is presented with a square grid of cells. Each cell initially contains a number. The aim of the game is to eliminate cells (by blacking them out) until the following three constraints all hold:

1) There are no duplicate numbers in each row and column.
2) There are no blacked out cells next to each other (horizontally or vertically).
3) All white cells are connected to each other in a single component.

Below is an example of a Hitori puzzle (Figure 1) and its solution (Figure 2):

4	8	1	6	3	2	5	7
3	6	7	2	1	6	5	4
2	3	4	8	2	8	6	1
4	1	6	5	7	7	3	5
7	2	3	1	8	5	1	2
3	5	6	7	3	1	8	4
6	4	2	3	5	4	7	8
8	7	1	4	2	3	5	6

Figure 1: Hitori Puzzle

■	8	■	6	3	2	■	7
3	6	7	2	1	■	5	4
■	3	4	■	2	8	6	1
4	1	■	5	7	■	3	■
7	■	3	■	8	5	1	2
■	5	6	7	■	1	8	■
6	■	2	3	5	4	7	8
8	7	1	4	■	3	■	6

Figure 2: Solution of Puzzle

In this assessment, your objective is to design a JavaFX application for playing Hitori.

The application must meet the following **basic** requirements:

1) **Initial Puzzle**: On starting the application, an initial puzzle must be displayed to the user (for example the grid in Figure 1).
2) **Elimination**: The user must be able to eliminate (i.e., black out) cells by left-clicking on them.
3) **Reactivation**: The user must be able to reverse the elimination action by right-clicking on a blacked-out cell. This cell should then become white again.
4) **Reset**: There must be a button to reset the puzzle to the starting state (where no cells are blacked out). When clicking the button, a confirmation dialog must be shown to the user first, allowing the user to confirm or cancel this action.
5) **Mistake Detection**: When the user has violated constraints 2 or 3 through their actions, this should be highlighted immediately and clearly to the user.
6) **Win Detection**: When the puzzle has been solved (i.e., constraints 1-3 hold), the user should be notified immediately and clearly.

The application should also meet the following **extended** requirements:

7) **Resizability**: The application should be resizable, with the grid resizing appropriately to fill the available space.
8) **Loading**: There should be functionality to load new puzzles from a file. Files should be assumed to contain a text representation (in UTF-8) of the puzzle, where each line in the file corresponds to a line in the puzzle and adjacent cells are separated by spaces. For example, the puzzle in Figure 1 would be represented as:

```
4 8 1 6 3 2 5 7
3 6 7 2 1 6 5 4
4 8 1 6 3 2 5 7
3 6 7 2 1 6 5 4
2 3 4 8 2 8 6 1
4 1 6 5 7 7 3 5
7 2 3 1 8 5 1 2
3 5 6 7 3 1 8 4
6 4 2 3 5 4 7 8
8 7 1 4 2 3 5 6
```

You can download this file at:
https://secure.ecs.soton.ac.uk/notes/comp1206/fe/example.txt

Mark Scheme

10 marks are awarded for each of Requirements 1-7. Up to 30 marks are awarded for Requirement 8. The final mark is out of 100.

Submission

Please make sure you submit only the .java files required to run the application. You should not rely on third-party packages (except for JavaFX). Please include a readme.txt file with brief instructions on how to compile and run the application, as well as any additional information that may help us run and test your application.

Further Assessment 2: JUnit and C Exercises

For this assessment, you should complete the original exercise sheets 9 and 10, on JUnit and C programming, respectively.

- Exercise Sheet 9 (JUnit): https://secure.ecs.soton.ac.uk/notes/comp1206/sheet9fe.html

- Exercise Sheet 10 (C Programming):
https://secure.ecs.soton.ac.uk/notes/comp1206/sheet10fe.html

We will award a mark out of 100 to each exercise sheet and your total mark for this component will be the average mark of the two exercise sheets.

You may re-use and/or revise solutions that you previously submitted for these exercise sheets, as long as they are entirely your own work.

Further Assessment 3: Blackboard CAA

This assessment is a time-limited online test that will be available on Blackboard. The specific day will be communicated to you later, but this will be during the week 28th September – 2nd October. The assessment can be started at any time during the designated day. Once started, you will have 2 hours to complete it (unless you are permitted extra time).

You will be able to access the assessment at this link: https://go.soton.ac.uk/cv1