**Department of Enterprise and Digital Innovation**

**Bachelor of Information and Communication Technologies**

**Best Programming Practices in .NET**

**BCPR283**

**Assignment 1 Part B Practical Project**

Semester 2 2019

Due date: Monday 25 November

Time: 09:00

Instructions:

Create and provide supporting documentation for running Sudoku as described in this document.

**TOTAL MARKS: 100**

Student Name/ID

*Ara Institute of Canterbury and its division members reserve the right to use electronic means to detect and help prevent plagiarism. Students agree that when submitting this assignment, it may be subject to submission for textual similarity review to Turnitin.com.*

Submissions received late will be subject to a penalty of 10% of the student’s mark per working day.

This assignment is worth 50% of the total marks for this course.

This paper has four(4) pages including the cover sheet.

**OVERVIEW**

You are to complete the writing a program that will enable a user to solve a Sudoku puzzle. This will be written in C#. Naturally you will be expected to use good design, programming and testing practices. For an overview of Sudoku, try the puzzles at [www.websudoku.com](http://www.websudoku.com) , or use Google.

**OUTSIDE THE SCOPE**

* Creating puzzles programmatically
* Solving the puzzles programmatically
* Checking if a given puzzle state can be solved

**PROVIDED FILES**

You have been provided with a set of Interface files as below:

* IGame for initialization purposes
* IGet which retrieves a cell value from a game board in various ways
* ISet which sets a cell value into a game board in various ways
* ISerialize which reads/writes values for a whole game board and implies knowledge of the way the data is stored, wither in CSV or in an appropriate class.

You MUST use these.

**REQUIRED STRUCTURES**

* Game will be serialized into a CSV format. You can decide (and document) this format
* The game board must be stored as a 1-d array
* The row, column and square numbering is 0-based.

**DELIVERABLES PART A (40 marks maximum)**

You need to write your Sudoku program so that it can be used.

* Your <<MODEL>> must meet the requirements of the assignment due in the previous term. Acknowledge any sources.
* Your <<VIEW>> should be implemented as some type of Windows form. This could be WinForms, or Windows Presentation Framework or any other that you may wish to investigate. Remember that the board size will be dynamic depending on the size of the game boing loaded.

**MUST (5 marks each)**

The following user features MUST be implemented: (5 marks each)

* Load a Game from CSV. The game that is loaded will determine how the size of the board. There should be more than one game available.
* Time the game.
* Do not accept an invalid number.
* Show if a row is complete.
* Show if a column is complete.
* Show if a square is complete.
* Do not override a default value (one that is set at the load).
* Show game complete.

If you do not have ALL of the MUST features implemented, none of the rest of the assignment will be marked.

**DELIVERABLES PART B (30 marks maximum)**

You may wish to enhance your program as follows:

**COULD (5 marks each)**

The following extra features COULD be implemented.

* Save/Load partially played games
* Design/Implement/Persist a scoring system for different levels
* Undo/Redo feature
* Hints for any square of all possible values
* X Sudoku, where the diagonals must also have the complete set of unique numbers.
* Different methods of storing games that are not CSV
* Other feature with approval of the tutor. Any feature suggestion should be put in the appropriate Moodle forum for approval.

If you do not have ALL of the MUST features implemented, you will get no marks for the COULD features. If you do not have at least 15 marks for Part C, you will get no marks for the COULD features.

**Marking Guide For Practical**

Each practical element will be marked on a scale of 0-5, taking into account Model implementation, View implementation, documentation, and elegance of implementation.

**DELIVERABLES PART C (30 marks maximum)**

* Discuss a selection of the following items that you will have dealt with in the creation of your program. Your discussion should relate to what you did do, compared to what you may perceive as best practice. Remember the paper is “Best Practices in .NET”:

1. Describe and use common Design Patterns, Algorithms and programming language Idioms.
2. Use appropriate software life-cycle models and software construction steps.
3. Design programs.
4. Design user interfaces which conform to recognised usability criteria.
5. Code programs in the specified language using the prescribed standards.
6. Produce and execute testing strategies at the systems level using a unit testing framework.
7. Debug and test programs to the systems level.
8. Provide all appropriate systems level documentation.
9. Maintain existing programs and update documentation.
10. Compare and contrast the features and uses of different programming languages.

* The theory discussion should cover:

1. Statement of the theory
2. Give credible references that establishes that it is widely accepted theory
3. Show/say what you have done / used / implemented
4. Show code or a working application or a scan of a design / sketch or other work product
5. Compare and contrast what you did with other possible approaches
6. Evaluation of the effectiveness of your approach
7. Say how you could have improved what you did

NOTE: You will lose 1/2 mark per thing for each of these things done inadequately

**Marking Guide For Theory**

In marking the theory component, I will consider both the correctness of the theory AND the amountof evidence you present to show that you ‘did stuff’.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 |
| CODE examples | No code | Presents OTHER PEOPLE’S CODE | code examples from your own code, not related to SUDOKU | 1-2 code examples from your own <<MODEL>> code, each of 10+ lines | 1-2 code examples from your own <<VIEW>> code |
| THEORY | Get it wrong | Minor / trivial point(s) | Valid point, but not that important | 1 Important / major point – ‘the most important thing’ | 2 important / major points |

**The marks for a question = min{ marks(Code examples), marks(Theory) }**

**HAND IN**

* A 10-minute MAXIMUM (optionally narrated) PowerPoint
  + For practical elements you should highlight the feature you have,
  + For theory elements, see the marking schedule