

Summary: Summary: In this week's assignment you will be writing a program to solve Sudoku puzzles. The puzzles will be loaded from a file. Solutions to the puzzles will be written to a file, or printed to the console if no file is specified. If you are not familiar with the sudoku puzzle, a nice description can be found here:

<https://en.wikipedia.org/wiki/Sudoku>

Sudoku Puzzle File We are defining a file format that will contain one or more sudoku puzzles for this assignment. The first line of this file is a format specifier followed by one puzzle per line.

The only format specifier supported is that for the 1.0 version of the file format which is the following.

```
#spf1.0
```

Each line will then encode a single puzzle. The encoding of the puzzle is 81 characters on a line with each character representing either a value or an empty box in the puzzle. Values are represented by the digits 1 through 9 and empty spaces are represented by the underscore character '_'. Finally the puzzle is encoded with the top row first followed by the next until the whole puzzle is encoded for example the following puzzle is represented by the sequence below.

			8		5			
	3			6				7
	9				3	8		
	4	7	9	5		3		
				7	1		9	
			2			5		
1					2	4	8	
		9					5	
					6			

```
__8_5__3_6__7_9__38__4795_3__71_9__2_5_1__248__9__5__6__
```

A file can contain any number of puzzles one per line after the format specifier.

Requirements While most of the design of the program is up to you there are some requirements for your program.

- You must have a class to represent the puzzle.
- You must overload the `std::ostream& operator<<` to pretty print the puzzle class.
- You must overload the `std::istream & operator>>` to load the puzzle class from a stream.
- You must be able to read and solve puzzles from a file in the defined format.
- You must allow the user to select a file to read puzzles from.
- You must allow the user to select a file to write the solutions. If a file is not selected by the user the solutions should be pretty printed to the screen.

The design of the pretty printing format is up to you.

You must use Catch2 to thoroughly test all your classes and as much functionality of your program as possible.

To complete your work you should use the following link to get a empty repo.

https://classroom.github.com/a/xXzVBkd_

Recommended Algorithm While we do not require you to use any particular algorithm we recommend that you use the following strategy.

When faced with solving an puzzle with an empty square in order check if each of the values 1-9 can be placed in the square and if so make that change and try to solve the puzzle from there. If none are possible return unsolvable.

This approach works quite efficiently if the tests for the validity of solutions prune invalid numbers early and should solve most puzzles in a few seconds.

Design and Style:

Much like we have been using the Google Java Style Guide for the Java code we have been writing we will be using the Google C++ Style Guide for C++. The style guide can be found at the following.

<https://google.github.io/styleguide/cppguide.html>

In this assignment we will be asking you to at least comply with the naming rules which can be found here.

<https://google.github.io/styleguide/cppguide.html#Naming>