



**Faculty of Engineering and Technology**  
**Electrical and Computer Engineering Department**  
*Linux Laboratory, ENCS3130*  
 Python Project

## Project Description

Have you ever tried to play the Sudoku game? Sudoku is a puzzle and excellent brain game. More information about the game and how to play it can be found at <https://sudoku.com/>.

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

Figure 1

In this project, you will write a python code to create and solve a Sudoku puzzle. A sample puzzle is shown in figure 1. To start the game the python script should setup the puzzle. The player has the option to **load a puzzle from a text file** or **start a random puzzle**. For the option of loading the Sudoku game from a text file, the format of the text file will be according to figure 2 which represents the puzzle shown in figure 1.

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

Figure 2

For the option of the random puzzle, the python script should ask about the level of difficulty which corresponds to the number of cells that are automatically filled. In this project, we will have three levels of difficulties as presented in Table 1.

Table 1

Level of difficulty	Percentage of cells automatically filled
Easy (E)	~ 40%
Medium (M)	~ 25%

Difficult (D)	~ 10%
---------------	-------

After puzzle setup (**load a puzzle from a text file** or **start a random puzzle**), the game can be played in two modes; One Player Mode, Two Players Mode.

**One Player Mode:** in this mode, the player will have the following options:

- **Fill:** fill a specific cell (row, column) by a specific value (value). In this case, the player will provide the input as a tuple (row, column, value). The python should check if this is a valid input. If a valid input, it will be added to the puzzle and the player gets one point. However, if not a valid input, it will not be added to the puzzle and the player loses one point.
- **Hint:** ask the game for a hint, in this case, the python script will fill one cell and print on the screen a tuple (row, column, value) that represents the filled cell. Also, the player will lose 2 points.
- **Solve:** ask the python script to solve the whole puzzle.

To compute the score of the player

- Determine the number of points achieved by the player ( $N_p$ ).
- Determine the number of the cells in the puzzle ( $M$ ). In this project,  $M = 9 \times 9$ .
- Determine the total time consumed by the player to solve the puzzle in seconds ( $T_p$ ).

$$Score_p = \begin{cases} \frac{N_p}{M} \times \left( \frac{3600}{T_p} \right) & N_p > 0 \\ 0 & N_p \leq 0 \end{cases}$$

**Two Players Mode:** in this mode, players play in turn and have the following options:

- **Fill:** Fill a specific cell (row, column) by a specific value (value). In this case, the player will provide the input as a tuple (row, column, value). The python should check if this is a valid input. If a valid input, it will be added to the puzzle and the player gets one point. However, if not a valid input, it will not be added to the puzzle and the player loses one point.
- **Pass:** Pass the turn to the other player. For every requested pass option, the player loses one point. If the Pass option is requested by the two players for 2 times in sequence (player 1 requested a Pass, then the player 2 requested a Pass, then the player 1 requested a Pass again, and then the player 2 requested a Pass), the hint option (explained in one player mode) is executed.
- **Solve:** Ask the python script to solve the whole puzzle.

To compute the score of each player

- Determine the number of the cells in the puzzle ( $M$ ). In this project,  $M = 9 \times 9$ .
- Determine the number of points achieved by each player ( $N_{p1}$ ,  $N_{p2}$ )
- Determine the total time of the turns of each player in seconds ( $T_{p1}$ ,  $T_{p2}$ ).

$$Score_{p1} = \begin{cases} \frac{N_{p1}}{M} \times \left( \frac{T_{p1} + T_{p2}}{T_{p1}} \right) & N_{p1} > 0 \\ 0 & N_{p1} \leq 0 \end{cases}$$

$$Score_{p2} = \begin{cases} \frac{N_{p2}}{M} \times \left( \frac{T_{p1} + T_{p2}}{T_{p2}} \right) & N_{p2} > 0 \\ 0 & N_{p2} \leq 0 \end{cases}$$

## Components:

- 1- Main script to execute the puzzle game.
- 2- Object-oriented coding: build your code by defining proper classes.
  - a. **Abstract Class:** you need to implement an abstract class with all the methods that must be created within any child class built from the abstract class.
  - b. **One Player Mode (OM) Class:** this class must be inherited from the abstract class. In this class you must implement all methods needed in the one player mode including puzzle setup and score calculation.
  - c. **Two Player Mode (TM) Class:** this class must be inherited from the OM class. Make sure to override only the methods different from the OM Class such as score calculation.
  - d. All Class members must be either **Private or Protected**.

## Submission and notes:

1. Submit a running Python Code through ITC.
2. Write the python code to satisfy the requirements described above.
3. Make sure your code is clean and well indented; variables have meaningful names, etc.
4. Make sure your code has enough comments inserted to add clarity.
5. Work in groups of at most two students
6. Deadline: 30 August, 2021 at 11:59pm.
7. This project is per group effort: instances of cheating will result in you failing the lab.
8. The project discussion date will be decided later.