

Gebze Technical University
Department of Computer Engineering
CSE 241/505
Object Oriented Programming
Fall 2019
Homework # 3
N-Puzzle Using OOP
Due date Nov 3rd 2019

In this homework, you will rewrite our N-Puzzle program using classes. Your overall program will behave very similar to your previous homework but your program structure will be considerably different.

Your main class will be named **NPuzzle**. It will have the following public functions

Fuction Name	Explanation
print	Prints the current configuration on the screen by sending it to cout
printReport	Prints a report about how many moves have been done since reset and if the solution is found
readFromFile	Reads the current configuration from the file given as function parameter. The file format is defined as in HW2.
writeToFile	Writes the current configuration to the file given as function parameter
shuffle	Makes N random moves to shuffle the board. N is given as a function parameter.
reset	Resets the current configuration to the solution.
setsize	Sets the board size to given values. The values are given as parameters and they can be at most 9x9. After setting the size, the boards should be reset.
moveRandom	Makes a valid random move
moveIntelligent	Makes an “intelligent” move
move	Makes a move according to the given char parameter. If the parameters is ‘L’ then, the blank tiles moves left, ..., etc, as defined in HW1.
solvePuzzle	Makes an attempt to solve the puzzle using your own algorithm from HW2.

Your **NPuzzle** class defines and uses a private inner class named **Board**, which represents the board configuration using a private C type 2D array. This class defines the following functions

Fuction Name	Explanation
print	Prints the board on the screen by sending it to cout
readFromFile	Reads the board from the file given as function parameter. The file format is defined as in HW2.
writeToFile	Writes the board to the file given as function parameter
reset	Resets the board to the solution.

setSize	Sets the board size to given values. The values are given as parameters and they can be at most 9x9. After setting the size, the boards should be reset.
move	Makes a move according to the given char parameter. If the parameters is 'L' then the blank tiles moves left, ..., etc, as defined in HW1.
isSolved	Returns true if the board is a solution

Your program will use object or objects of **NPuzzle** to perform what we did previously in HW1 and HW2. Your command line options and your user interface is the same. The following table is repeated here just for convenience.

Input	Action
V	Solves the problem from the current configuration using the intelligent algorithm.
T	Prints a report about how many moves have been done and if the solution is found
E	Asks a file name and saves the current board configuration as a loadable shape file.
O	Asks a file name and loads the current board configuration from the shape file.
L	moves the empty cell left if there is room
R	moves the empty cell right if there is room
U	moves the empty cell up if there is room
D	moves the empty cell down if there is room
I	makes an "intelligent" move for you.
S	Shuffle- takes the board to the final solution, and makes size*size random moves to shuffle the board.

Notes:

- Do not use any functions from the standard C library (like **printf**), use `cout`, `cin`, etc.
- Your program should have no global functions other than `main`. Do not use any global variables.
- Do not use any C++ features that we did not learn during the lectures.
- You will use all the object oriented techniques that we learned in the class including **const**, **inline**, **decltype**, **auto** keywords.
- Do not forget to indent your code and provide comments.
- Check the validity of the user input.
- **Test your programs very carefully at least with 10 different runs and submit your result files for each.**
- You should submit your work to the moodle page and follow all the submission rules that will be posted.