Gebze Technical University
Department of Computer Engineering
CSE 241/505
Object Oriented Programming
Fall 2022
Homework # 5
Inheritance and Templates
Due date Dec 25th 2022

In this homework, you will design and implement a class hierarchy for the Tetris class. You will also write global functions to use these classes.

Your **`AbstractTetris`** class represents the game configuration as you did in the HW3 homework assignments. You may use the sample HW2 file that we have provided.

Fuction Name	Explanation
`draw`	Draw function to draw the Tetris board. It will optionally start the drawing from the top. See how to move your cursor to the top of your screen on a UNIX terminal at https://en.wikipedia.org/wiki/ANSI_escape_code#CSI_sequences. Note that your code will not be portable because it will work only on certain consoles.
`readFromFile`	Reads the game from the file given as function parameter. The file format will be decided by you.
`writeToFile`	Writes the game to the file given as function parameter
`operator +=`	Adds a Tetromino to the board. The new Tetromino will be added at the top row in the middle.
`animate`	Animate function to animate the added tetromino dropping to the bottom of the board. The animation will be repetition of four steps: 1. Draw the board with Tetromino at the top 2. Ask the user rotation direction and rotation count 3. Ask the user move direction and count 4. Rotate and move the Tetromino 5. Draw the board 6. Sleep 50 milliseconds 7. Lower the Tetromino one level and go to step 5 until it hits the bottom.
`fit`	Do not implement any fit functions!
`lastMove`	Returns the last move, if there is no last move, throws exception.
`numberOfMoves`	Returns the number of steps (moves) this board made

Many of the functions above cannot be implemented because your do not know how the board is represented in this abstract base class. You will derive 2 new concrete classes and 1 new templated class from this class that represent the boards in different ways:

- **`TetrisVector`**: The Board is represented using an STL vector of STL vectors.
- **`TetrisArray1D`**: The Board is represented using a one-dimensional dynamic C array.
- **`TetrisAdapter`**: This class takes a template parameter and behaves like an adaptor class just like the stack or queue class of the STL. The parameter classes can be any STL class with a random access iterator.

You will submit two driver source files, each will include a main function. Your first driver code will test each member function of three concrete Tetris classes.

Your second driver code will do the following

1. Ask the user the size of the Tetris board and the type of the Tetris class (V for vector, 1 for array1D, A for adaptor)
2. Ask the Tetromino type (I, O, T, J, L, S, Z). User may enter R for random Tetromino, Q for quit.
3. Add the asked Tetromino to the board and animate
4. Go to 2

Notes:

- Use all the OOPL rules we learned in the class.
- **Throw exceptions and write code to test them.**
- Make your own namespace, use separation of interface and implementation rules.
- Test each function of each class at least once by writing driver code.
- Test the global function at least 5 times with different number of types of board pointers.
- Submit all your source files and a MAKEFILE that compiles and runs your project.