CSE 230 Problem Set 08

# Problem 24.1: Fragile

Consider a Position class representing a position on a chess board. Here the internal representation of the position is a single character. Note that a character has 256 possible values, but a chess board has 64 possible values. There are therefore plenty of bits to represent this position. To accomplish the mapping between the row and column values and the internal representation, the following code is provided:



Implement this class in C++. When you are finished, establish that your implementation has fragile robustness:

Code

Rationale

# Problem 24.2: Tested

From the Position class of Problem 24.1, create the assurances necessary to establish that the class has tested robustness. If any bugs are found in this process, please fix them and provide the new class definition.

Code of driver

Rationale

Code of Position Class

# Problem 24.3: Strong

From the Position class of Problem 24.1 and 24.2, create the assurances necessary to establish that the class has strong robustness. If any bugs are found in this process, please fix them and provide the new class definition.

Hint: You may need to create a simple test document to solve this problem and write some simple automation.

Test cases

Code of driver

# Problem 24.4: Resilient

From the Position class of Problem 24.1 – Problem 24.3, create the assurances necessary to establish that the class has resilient robustness. Provide only 3 test functions. If any bugs are found in this process, please fix them and present the updated class definition.

Code of tests