tic tac toe board Class

Required Libraries

#include <iostream>

#include <string>

#include "symbol.h"

This class creates a tic tac toe board, places X’s and O’s using the symbol library, determines when the tic tac toe game is over, and declares a winner.

**Constructors:**

tBoard();

Creates a tBoard object and initializes all the symbols on the tic tac toe board to ‘BLANK’

Sample Usage:

tBoard myGame;

cout << myGame;

Creates

| 0 | 1 | 2 |

+-----------------+

0| | | |

+-----------------+

1| | | |

+-----------------+

2| | | |

+-----------------+

**Mutators:**

bool move(symbol m, int x, int y);

Changes the symbol(which should be blank) located the x, y to the specified symbol

Returns false if the symbol isn’t blank

Sample Usage:

myGame.move(X,0,0)

| 0 | 1 | 2 |

+-----------------+

0| X | | |

+-----------------+

1| | | |

+-----------------+

2| | | |

+-----------------+

bool game\_over();

Checks if the game is over by checking for one of the 8 winning moves in tic tac toe

Or if every space has been filled

Sample Usage:

if(myGame.game\_over())

cout << “Game Over” << endl;

symbol winner();

Determines the winning symbol X or O. If tie, returns BLANK

Sample Usage:

if(myGame.winner() == X)

cout << “Congratulations X!” << endl;

string toString() const;

The method that turns the game board into a string.

Sample Usage:

cout << myGame.toString();

ostream & operator<<(ostream& os, const tBoard& myTable);

This is what allows you to type something like cout << myGame; and it will write the game board to console.

Requires the help of:

void insert(ostream & out) const;

facilitates the overloading of the << operator

Sample Usage:

cout << myGame;