System Manual:

symbol location\_[3][3];

This creates a 2D vector that holds 9 different memory locations that can be used to create the tic tac toe board.

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

This function checks to see if the move is valid by seeing if there is a symbol there already. If there is something in the spot it will return false. If there is nothing in the spot the memory location equals O or X depending on what m is. This will return true if there is nothing in the memory location.

bool game\_over();

This function checks to see if winner()=X or if winner()=O or if the winner()=BLANK while all of the memory locations are filled. If the winner()=X,O, or BLANK while the locations are all filled, it will return true. If none of those conditions apply it will return false and the game will not end.

symbol winner();

This function checks for all of the possible winning combinations. There are 8 conditions in which someone can win. It checks to see if X or O satisfied any of the conditions. If one of the conditions is satisfied then it will return X or O depending on which symbol it is. If none of this winning conditions are met, BLANK will be returned.

string getMyBoard()const;

This function outputs the board and all of the memory locations. This function will also be used in ostream & operator<<(ostream& os, const tBoard& myTable).

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

This function returns the game board by using myTable.getMyBoard().