TicTacToe Design

Task 1: Print the board every after each character is inputted by the user

- Functions:
  - PrintBoard();
    - Variables:
      - int i
      - int j
      - vector<vector<char> > board
- Procedures:
  - Nested for loop that prints out the board
- Output
  - game board

Task 2: Get correct coordinates from the user

- Functions:
  - getLocation()
    - Variables
      - vector<vector<char> > board
      - int row
      - int col
      - char currLetter
      - bool locChecker

  - locationInRange();
    - Variables:
      - vector<vector<char> > board
      - int row
      - int col
      - bool checkloc

  - locationTaken
    - Variables:
      - vector<vector<char> > board
      - int row
      - int col
      - bool checkLoc
      - bool locCheckek
- Procedures:
  - main calls getLocation
  - getLocation() gets the input from the user depending if currLetter is set to either ‘X’ or ‘O’
  - when the coordination is inputted, getLocation() calls locationTaken()
  - locationTaken() checks to see if the location inputted is within the dimension of the game board. If not, then the user gets asked for different coordinates. If it is, then locationTaken() checks to see if

the coordinate inputted has an existing character. If it does, then the user is asked for different coordinates. If the location is not taken, row and col variables get set to the inputted values

- main runs this 9 times, alternating with currLetter set as 'X' or 'O' to switch players.

- Output:
    - game board with the characters placed at valid locations

Task 3: Check for winner

- Functions:
    - checkWin()
        - Variables:
            - char winCheck

Procedures:

- Main calls checkWin();
- Tests all individual cases for winner, which means the board is checked for characters placed adjacently 3-in-a-row. If one is found, then the winning character is returned.
- In main, checkWin will break out of the while loop that runs 9 turns if a winner is found.

- Output:
    - X won!!!
    - O won!!!
    - Cat!!!