```java
import java.util.Scanner;
import java.util.Random;

/**
 * Pre-conditions
 * The user selects a valid symbol (either 'x' or 'o').
 * The board is initialized and displayed with 9 empty spaces (from "1"
to "9").
 * The game starts with the player's turn.
 */


public class TicTacToe {
    public static void minigameplayer() {
        // Has the logic code and then place the drawing into it
        // Program going to use 9 diff variables for spaces, depending
on what is placed code will randomize its next choice.
        // top left= tl top middle= tm top right= tr and so on.
        // 1.  user chooses to play as x or o, they will choose a box
(1-9).
        // 2.  computer should choose next and so on till one person
wins

        //Initializing game components
        Scanner170 console = new Scanner170(System.in); //Initialized
scanner object 'console' for user input using Scanner170.
        Random random = new Random(); //Initializes random object for
computers random moves that allow it to play the user in console.
        String[] board = {"_1_", "_2_", "_3_", "_4_", "_5_", "_6_",
"_7_", "_8_", "_9_"}; //Creates a board with 9 spaces (1-9).
        char playerSymbol = ' ';
        char computerSymbol = ' ';
        boolean isPlayerTurn = true;

        //Choose symbol x or o
        //Loop until user selects a valid symbol input into the
console.
        //Read input choice from user.
        while (true) {
            System.out.print("Choose x or o: "); //User chooses symbol
x or o.
            String choice = console.nextLine().toLowerCase(); //Custom
input handling
            if (choice.equals("x")) { //If console input is x, set
playerSymbol to x, and computerSymbol to o.
                playerSymbol = 'x';
                computerSymbol = 'o';
                break;
            } else if (choice.equals("o")) { //Else if the input is o,
set playerSymbol to and computerSymbol to x.
                playerSymbol = 'o';
                computerSymbol = 'x';
                break;
            } else { //Else print to the console that user input is
invalid, and to enter a valid input into the console.
                System.out.println("Not a valid letter, pick x or o.");
//Invalid output message to console.

            }
        }
        System.out.println("You are " + playerSymbol + "computer is " +
computerSymbol); //Display the computer's symbol and the user's symbol
in console.
        System.out.println("Now play"); //Print "Now play" to console
for user.

        //Game loop will alternate between player and computer's turn,
until loop ends.
        while (true) { //While loop allows the game to be continued
until the board is full or Tie, Winner, Loser condition is displayed to
the user in console.
            printBoard(board); //Prints the current game board.

            if (checkWinner(board, playerSymbol)) { // If
(checkWinner(board, playerSymbol)) = true, display user winner message.
                System.out.println("You Win!"); //User win condition
                break;
            } else if (checkWinner(board, computerSymbol)) { //Else if
checkWinner(board, computerSymbol) is = true, display user loss
message.
                System.out.println("Computer wins, you lose!"); //User
loss condition
                break;
            } else if (isBoardFull(board)) { //Else if the board is
full, and not winner is declared the tie condition message is displayed
to the user in the console.
                System.out.println("It's a tie!"); //Tie condition
                break;
            }
            //Break loop results in game ending.
            //Calling the function print board, will result in
displaying the current game board to the console.


            if (isPlayerTurn) { //If PlayerTurn = true
                System.out.print("Choose a space 1-9: "); //"Print
choose a space 1-9" to console.
                int move;
                while (true) { //Loop until the user selects a valid
space.
                    if (console.hasNextInt()) { //Reads the players
move.
                        move = console.nextInt() - 1; //Converts the
move to an index by subtracting 1.
                        //If move is a valid number (between 1-8), and
place is unoccupied, the board updates with the players symbol at that
position they chose.
                        if (move >= 0 && move < 9 &&
!board[move].contains("x") && !board[move].contains("o")) {
                            board[move] = "_" + playerSymbol + "_";
                            break; //Break out of the loop.
                        } else {
                            System.out.print("Enter a number 1-9");
//Tell user to select a valid and non-occupied space in console.
                        }
                    } else {
                        console.next();
                        System.out.println("Enter valid number.");
//Tell user to input a valid number into the console.
                    }
                }

            } else { //Computer's turn
                //Loop until the computer selects a valid, non-occupied
space on the game board.
                System.out.println("Computer's turn..."); //Display
Computer's turn to console.
                int move;
                while (true) { //Validates computer move.
                    move = random.nextInt(9); //Generates a random move
for the computer opponent (number between 0 and 8)
                    if (!board[move].contains("x") &&
!board[move].contains("o")) { //If the space in unoccupied, update the
board with the computer's symbol.
                        board[move] = "_" + computerSymbol + "_";
                        break; //Break out of the loop.
                    }
                }
            }
            isPlayerTurn = !isPlayerTurn; //Allows for switching turns
by setting isPlayerTurn to the opposite of its current value.
        } //Player and computer's turn is toggled.
        console.close(); //Closes the scanner.
    }

    public static void printBoard(String[] board) {
        System.out.println();
        for (int i = 0; i < 9; i += 3) {
            System.out.println(board[i] + board[i + 1] + board[i + 2]);
        }
        System.out.println();
    }

    //Game board set up.
    public static boolean checkWinner(String[] board, char symbol) {
        int[][] winConditions = { //User win conditions.
                {0, 1, 2}, {3, 4, 5}, {6, 7, 8}, //Rows
                {0, 3, 6}, {1, 4, 7}, {2, 5, 8},//Columns
                {0, 4, 8}, {2, 4, 6}// Diagonals
        };
        String winningString = "_" + symbol + "_"; //Game board
conditions.
        for (int[] condition : winConditions) {
            if (board[condition[0]].equals(winningString) &&
                    board[condition[1]].equals(winningString) &&
                    board[condition[2]].equals(winningString)) {
                return true;
            }
        }
        return false;
    }

    public static boolean isBoardFull(String[] board) {
        for (String space : board) {
            if (!space.contains("x") && !space.contains("o")) { //If
board is full boolean = false
                return false;
            }
        }
        return true;

/**
 * Post-conditions
 * The board is updated after each move (player or computer) and
displayed after every turn.
 * The game ends when there is a winner (either player or computer) or
when the board is full (resulting in a tie).
 * The final result (win/lose/tie) is displayed.
 * Scanner object is closed.
 */

    }
}
```