**Tic-Tac-Toe Game**

**1. Introduction**

This is a simple console-based Tic-Tac-Toe game written in Java. The game allows two players to play against each other on a 3x3 grid. Each player selects a mark, either "X" or "O," and takes turns to place their mark in the desired position on the grid. The first player to align three of their marks in a row, column, or diagonal wins the game. If all positions are filled without a winner, the game ends in a draw.

**2. Features**

**Two-Player Game play:** The game supports two players, with each player choosing a unique mark ("X" or "O").

**Input Validation:** Ensures that players select either "X" or "O" as their mark.

**Move Validation:** The players are prompted to choose their move on an available cell by entering a number from 1 to 9, which corresponds to an empty position on the grid.

**Winning Conditions Check:** Checks for winning conditions after each move, including rows, columns, and diagonals.

**Display Board State:** Displays the board's state after each move to keep both players informed of the game's progress.

**End Conditions:** Declares a win if any player aligns three marks in a row, column, or diagonal, or a draw if all cells are filled without a winner.

**3. How to Play**

1. **Start the Game**
   1. Run the program, and it will prompt you to enter the names of the two players.
2. **Select a Mark**
   1. The first player is asked to choose a mark, either "X" or "O".
   2. Once the first player selects a mark, the other player is automatically assigned the opposite mark.
3. **Understand the Board Layout**
   1. The game board is displayed as a 3x3 grid with numbers from 1 to 9, representing each cell.
   2. Each number corresponds to a cell where players can place their mark.
4. **Take Turns**
   1. Players take turns entering a number from 1 to 9 to mark their chosen cell.
   2. Entering a number will replace that cell with the player’s mark (either "X" or "O").
5. **Win or Draw Check**
   1. After every move, the game checks for a win condition (three consecutive marks in a row, column, or diagonal).
   2. If a player achieves this, the game declares that player the winner and ends.
   3. If all cells are filled without any winning combination, the game declares a draw.
6. **End of Game**
   1. Once a win or draw is detected, the game displays the result and terminates.

**4. Example Output**

Enter the name of First player: Alice

Enter the name of Second player: Bob

Alice Select the mark (O or X): X

1 2 3

4 5 6

7 8 9

Alice Turn: 1

X 2 3

4 5 6

7 8 9

Bob Turn: 5

X 2 3

4 O 6

7 8 9

Alice Turn: 2

X X 3

4 O 6

7 8 9

Bob Turn: 3

X X O

4 O 6

7 8 9

Alice Turn: 6

X X O

4 O X

7 8 9

Bob Turn: 9

X X O

4 O X

7 8 O

Match is Draw

**5. Requirements**

* **Java**: The program is written in Java, so you need to have Java Development Kit (JDK) installed to compile and run it.
* **Text Editor/IDE**: Use any text editor or IDE like Eclipse, IntelliJ IDEA, or VS Code to write and run the code.
* Console or terminal to run the program.
* Basic understanding of Java syntax and 2D arrays.

**6. Error Handling for Invalid Moves**

Adding input validation for invalid moves. Currently, if a player selects a number that is already occupied or a number outside the range, the program may misbehave. Adding this check would make the game more robust and user-friendly.