

#### ST MARY UNIVERSITY

# DEPARTMENT OF COMPUTER SCIENCE

## COURSE: INTRODUCTION COMLUTER PROGRAMING

### **GROUP NUMBER 4**

TITLE: TIC-TAC-TOE GAME

GROUP MEMBERS	ID NUMBER
	D CD (0000 /0015
1) Abel Baysa	RCD/0229/2017
2) Adoniyas Melaku	RCD/0232/2017
3) Ahmed yesuf	RCD/0234/2017
4) Dawit ketsela.	RCD/0248/2017
5) Kirubel mekonnen	RCD/0265/2017
6) Hryakos Gorfu	ECD/1262/2017
7) Haileyesus Abera	RCD/2572/2016
8) Eman Ali.	RCD/0253/2017

### Overview and functionality.

This code is a C++ implementation of a two-player Tic-Tac-Toe game that runs in the console. The program allows two players to take turns marking spaces on a 3x3 grid, with one player using 'X' and the other using 'O'.

The game starts by welcoming the players and displaying an empty board with positions numbered 1 through 9. Player 1 is prompted to choose whether they want to be X or O. Player 2 is automatically assigned the remaining symbol.

Players take turns entering numbers 1-9 to place their symbol on the corresponding board position. The game checks for several conditions:

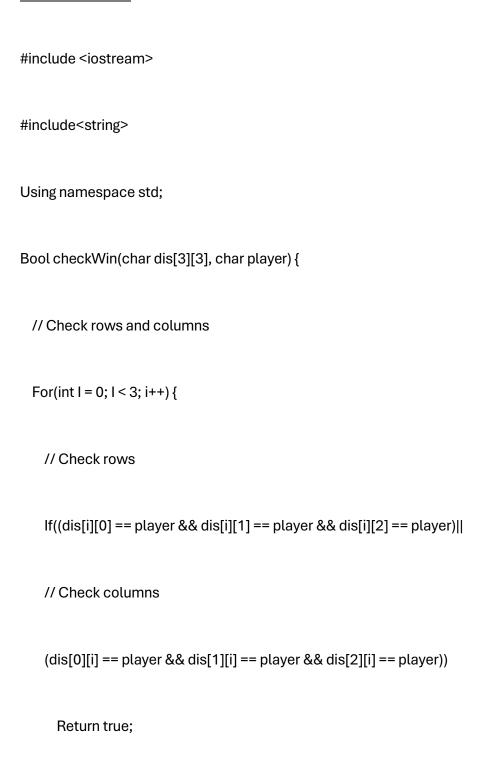
- Valid position input (must be 1-9)
- Position availability (cannot overwrite existing marks)
- Win conditions (3 in a row horizontally, vertically or diagonally)
- Draw condition (board filled with no winner)

After each move, the updated board is displayed. If a player wins, the game announces the winner and ends. If all spaces are filled without a winner, it declares a draw.

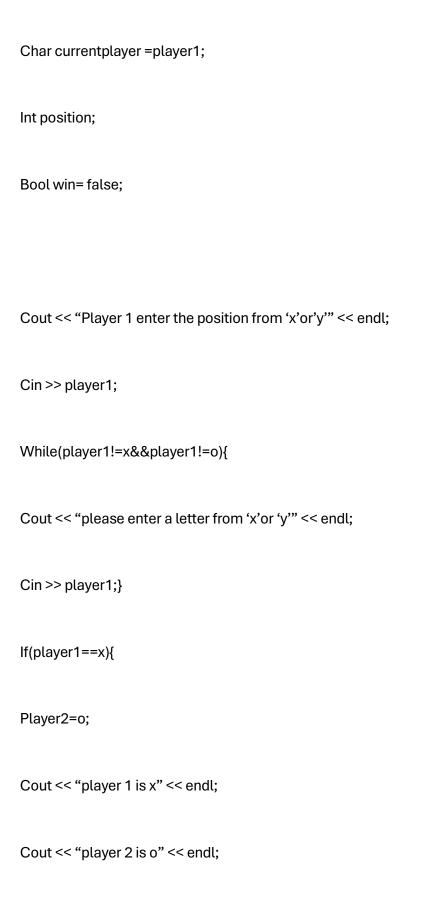
The code uses a 3x3 character array to represent the game board and includes functions to check for win conditions and handle player input. The game alternates turns between players until a win or draw occurs.

The implementation includes basic error handling for invalid inputs and already-taken positions, prompting players to enter valid moves. The board is redrawn after each move to show the current game state.

### **CODE SNIPPET**



```
If((dis[0][0] == player \&\& dis[1][1] == player \&\& dis[2][2] == player)||
  (dis[0][2] == player && dis[1][1] == player && dis[2][0] == player))
    Return true;
  Return false;
}
 }
Int main(){cout << " WELLCOM TO THE GAME " << endl;
Cout<<endl;
Char dis[3][3]={{'1','2','3'},{'4','5','6'},{'7','8','9'}};
Const char x='x',o='o';
Char player1;
Char player2;
```



```
Currentplayer=x;}
Else if(player1==o){
Player2=x;
Cout << "player 1 is o" << endl;
Cout << "player 2 is x" << endl;
Currentplayer=o;}
Cout << " Good luck" << endl;
For(int turn=0;turn<9;turn++){
Cout << " | | " << endl;
Cout << "`<< dis[0][0]<< "`|"<< dis[0][1]<< "`|"<< dis[0][2]<< "`"<< endl;
Cout << "___|___" << endl;
Cout << " | | " << endl;
Cout <<" "<<dis[1][0]<< " | "<<dis[1][1]<<" | "<<dis[1][2]<<" " << endl;
```

```
Cout << "__|___" << endl;
Cout << " | | " << endl;
Cout <<" "<<dis[2][0]<< " | "<<dis[2][1]<<" | "<<dis[2][2]<<" " << endl;
Cout << " | | " << endl;
//this is optional if you want we can jump code this but our program have a problem
/*while(true) {
     Cout << "Player " << (currentplayer == player1? "1": "2") << ", enter position (1-9): ";
     If(!(cin >> position)) { // If input fails (non-number)
       Cin.clear(); // Clear error flag
       Cin.ignore(numeric_limits<streamsize>::max(), '\n'); // Discard bad input
       Cout << "Please enter a number between 1-9!\n";
       Continue;
```

```
}
If(position < 1 || position > 9) {
  Cout << "Position must be between 1-9!\n";
  Continue;
}
Int row = (position - 1) / 3;
Int col = (position - 1) \% 3;
If(dis[row][col] == x \mid\mid dis[row][col] == o) \{
  Cout << "Position already taken!\n";
  Continue;
```

```
}
      Break; // Valid input
   }*/
Cout << "pleyer " << (currentplayer == player1?"1":"2")<<",enter the position:\n";
Cin>>position;
While(position < 1 || position > 9) {
      Cout << "Invalid position. Please enter 1-9: ";
      Cin >> position;}
Int row = (position - 1) / 3;
Int col = (position - 1) \% 3;
While(dis[row][col] == x \parallel dis[row][col] == 0) {
```

```
Cout << "Position already taken. Choose another: ";
    Cin >> position;
    Row = (position - 1) / 3;
    Col = (position - 1) \% 3; 
Dis[row][col]=currentplayer;
If(checkWin(dis, currentplayer)) {
    Cout << "\n | | " << endl;
    Cout << " " << dis[0][0] << " | " << dis[0][1] << " | " << dis[0][2] << " " << endl;
    Cout << "__|__|__" << endl;
    Cout << ``` << dis[1][0] << `` | `` << dis[1][1] << `` | `` << dis[1][2] << ``` << endl;
    Cout << "___|___" << endl;
    Cout << " " << dis[2][0] << " | " << dis[2][1] << " | " << dis[2][2] << " " << endl;
```

```
Cout << " | | " << endl;
      Cout << "CONGRATULATIONS! Player " << (currentplayer == player1 ? "1" : "2") << " (" <<
currentplayer << ") wins!\n";</pre>
      Win = true;
      Break;
   }
 Currentplayer=(currentplayer==player1)? Player2:player1;
}
  If(!win) {
    Cout << "\n | | " << endl;
    Cout << ``` << dis[0][0] << `` | `` << dis[0][1] << `` | `` << dis[0][2] << ``` << endl;
    Cout << "___|___" << endl;
```

```
Cout << " " << dis[1][0] << " | " << dis[1][1] << " | " << dis[1][2] << " " << endl;

Cout << " " << dis[2][0] << " | " << dis[2][1] << " | " << dis[2][2] << " " << endl;

Cout << " | | " << endl;

Cout << " | | " << endl;
```

Return 0;}