Tic Tac Toe Problem

```
Create a Tic Tac Toe Program
Sample output from your program should look like the following:


================================e
      ---a------b------c---
      |      |      |     |
      ---d------e------f---
      |      |      |     |
      ---g------h------i---
      |      |      |     |
      ---------------------
Entering input for: player1: O
enter in cell  [a, b, ... i]e
================================e
      ---a------b------c---
      |      |      |     |
      ---d------e------f---
      |      |  O   |     |
      ---g------h------i---
      |      |      |     |
      ---------------------
Entering input for: player2: X
enter in cell  [a, b, ... i]b
================================e
      ---a------b------c---
      |      |  X   |     |
      ---d------e------f---
      |      |  O   |     |
      ---g------h------i---
      |      |      |     |
      ---------------------
Entering input for: player1: O
enter in cell  [a, b, ... i]i
================================e
      ---a------b------c---
      |      |  X   |     |
      ---d------e------f---
      |      |  O   |     |
      ---g------h------i---
      |      |      |  O  |
      ---------------------
Entering input for: player2: X
enter in cell  [a, b, ... i]a
================================e
      ---a------b------c---
      |  X   |  X   |     |
      ---d------e------f---
      |      |  O   |     |
      ---g------h------i---
      |      |      |  O  |
      ---------------------
Entering input for: player1: O
```

```
enter in cell  [a, b, ... i]c
===============================e
      ---a------b------c---
      |  X   |  X   |  O  |
      ---d------e------f---
      |      |  O   |     |
      ---g------h------i---
      |      |      |  O  |
      ---------------------
Entering input for: player2: X
enter in cell  [a, b, ... i]f
===============================e
      ---a------b------c---
      |  X   |  X   |  O  |
      ---d------e------f---
      |      |  O   |  X  |
      ---g------h------i---
      |      |      |  O  |
      ---------------------
Entering input for: player1: O
enter in cell  [a, b, ... i]g
********  We have a winner: player1: O
===============================e
      ---a------b------c---
      |  X   |  X   |  O  |
      ---d------e------f---
      |      |  O   |  X  |
      ---g------h------i---
      |  O   |      |  O  |
      ---------------------

Do you want to play Tic-Tac-Toe (y/n)?y
===============================e
      ---a------b------c---
      |      |      |     |
      ---d------e------f---
      |      |      |     |
      ---g------h------i---
      |      |      |     |
      ---------------------
Entering input for: player1: O
enter in cell  [a, b, ... i]e
===============================e
      ---a------b------c---
      |      |      |     |
      ---d------e------f---
      |      |  O   |     |
      ---g------h------i---
      |      |      |     |
      ---------------------
Entering input for: player2: X
enter in cell  [a, b, ... i]a
===============================e
      ---a------b------c---
      |  X   |      |     |
      ---d------e------f---
      |      |  O   |     |
      ---g------h------i---
      |      |      |     |
      ---------------------
Entering input for: player1: O
enter in cell  [a, b, ... i]c
===============================e
      ---a------b------c---
      |  X   |      |  O  |
      ---d------e------f---
      |      |  O   |     |
      ---g------h------i---
      |      |      |     |
      ---------------------
Entering input for: player2: X
enter in cell  [a, b, ... i]g
===============================e
      ---a------b------c---
      |  X   |      |  O  |
      ---d------e------f---
      |      |  O   |     |
      ---g------h------i---
      |  X   |      |     |
      ---------------------
Entering input for: player1: O
enter in cell  [a, b, ... i]d
===============================e
      ---a------b------c---
      |  X   |      |  O  |
      ---d------e------f---
      |  O   |  O   |     |
      ---g------h------i---
      |  X   |      |     |
      ---------------------
Entering input for: player2: X
enter in cell  [a, b, ... i]f
===============================e
      ---a------b------c---
      |  X   |      |  O  |
      ---d------e------f---
      |  O   |  O   |  X  |
      ---g------h------i---
      |  X   |      |     |
      ---------------------
Entering input for: player1: O
enter in cell  [a, b, ... i]h
===============================e
      ---a------b------c---
      |  X   |      |  O  |
      ---d------e------f---
      |  O   |  O   |  X  |
      ---g------h------i---
      |  X   |  O   |     |
      ---------------------
Entering input for: player2: X
enter in cell  [a, b, ... i]b
===============================e
      ---a------b------c---
      |  X   |  X   |  O  |
      ---d------e------f---
      |  O   |  O   |  X  |
      ---g------h------i---
      |  X   |  O   |     |
      ---------------------
Entering input for: player1: O
enter in cell  [a, b, ... i]i
===============================e
      ---a------b------c---
      |  X   |  X   |  O  |
      ---d------e------f---
      |  O   |  O   |  X  |
      ---g------h------i---
      |  X   |  O   |  O  |
      ---------------------
********  Game Ends in Draw

Do you want to play Tic-Tac-Toe (y/n)?n
Bye
```

****

In your JH7_worksheet.txt file, I want you to paste in at least 2 games played. One should have a winner, and another should be a draw. At the end, you should terminate tic_tac_toe.

A starting point for your program is provided (if you feel overly constrained by this and have a better idea on structuring this game you can create your own structure -- main should not have very much code):

```
package tictactoe;

import java.util.*;

class TicTacToe
{
    char ttt[][] = new char[3][3];
    static final char player1 = 'O';
    static final char player2 = 'X';
    Scanner scan  =new Scanner(System.in);


    String playerID(char player)
    {
    // Prints the identity of the player
    // For example:
    // player2: X

        if (player == player1)
            return "player1: "+player;
        else
            return "player2: "+ player;
    }

    void getPlayerInput(char player)
    {
    // ******* Details to fill in ************

       // Prompt the user for a cell input.  Make sure its a legal
       // cell designator.  Also make sure the cell doesn't already
       // have a player in it.  Prompt the user again in the case
       // of any problem.  Once a valid spot has been found,
       // you will issue a statement like:

         ttt[row][col]=player;

    }

    boolean gameIsDraw()
    {
     // ******* Details to fill in ************

        // If all ttt entries have been taken return true
        // otherwise return false
    }

    boolean winner(char player)
    {
     // ******* Details to fill in ************

     // Check to see if the parameter player has won
     // Winning means that player shows up in a line of
     // three.  The line can be a column, row or a diagonal
     // Return true if player is a winner, otherwise return false.
    }

    void displayBoard()
    {
     // ******* Details to fill in ************

     // If you want to skip the fun of this, I have a candidate displayBoard
down below

    }


    void newgame()
    {
        char currPlayer = player1;
        for(int i=0; i<3; i++)
            for(int j=0; j<3; j++)
                ttt[i][j] =' ';

        boolean continueFlag = true;
        while (continueFlag)
        {
            displayBoard();
            if (gameIsDraw())
            {
                System.out.println("Game Ends in Draw");
                continueFlag = false;
            }
            else
            {
                getPlayerInput(currPlayer);
                if (winner(currPlayer))
                {
                    System.out.println("We have a winner: " +
playerID(currPlayer));
                    displayBoard();
                    continueFlag = false;
                }
                else
                {
                    if (currPlayer == player1) currPlayer = player2;
                        else currPlayer = player1;
                }
             }
        }

    }


    public static void main(String[] args)
    {
        TicTacToe game = new TicTacToe();
        String str;
        do
        {
            game.newgame();

            System.out.println("Do you want to play Tic-Tac-Toe (y/n)?");
            str= game.scan.next();
        } while ("y".equals(str));

        System.out.println("Bye");
    }
}
```

By the way, I'm going to give a version of displayBoard(). This routine is a bit of a pain and I'm not sure how much learning happens by going through the pain. Also, I'm not real happy with my version, but it at least works. So... if you want to use my version, here it is:

```
void displayBoard()
    {
        System.out.println("********************************");
        System.out.println("      ---a------b------c---");

        for (int i=0; i<3; i++)
        {
            for (int j=0; j< 3; j++)
            {
              if (j == 0) System.out.print("      |  ");
              System.out.print(ttt[i][j]);
```

```
          if (j < 2) System.out.print( "   |  ");
          if (j==2)  System.out.print("  |");
        }
        System.out.println();
        switch (i)
        {
        case 0:
            System.out.println("      ---d------e------f---");
            break;
        case 1:
            System.out.println("      ---g------h------i---");
            break;
        case 2:
            System.out.println("      ---------------------");
            break;
        }
    }
}
```