Name – Pranav Salunkhe

TY CS-D

Batch-1

Roll no – 09

Assignment 1 A

Code –

import java.util.Scanner;

public class XoGame {

    private char[][] board;

    private char currentPlayer;

    private int moves = 9;

    public XoGame(char first) {

        board = new char[3][3];

        currentPlayer = first;

        for (int i = 0; i < 3; i++) {

            for (int j = 0; j < 3; j++) {

                board[i][j] = ' ';

            }

        }

    }

    public void playGame() {

        Scanner sc = new Scanner(System.in);

        boolean gameEnded = false;

        System.out.println("Welcome to Tic-Tac-Toe!");

        while (!gameEnded) {

            System.out.println("\nCurrent board:");

            printBoard();

            System.out.println("Player " + currentPlayer + ", enter your move 1-9:");

            int input=0;

            input = sc.nextInt();

            input--;

            int row = input/3;

            int col = input%3;

            if (isValidMove(row, col)) {

                board[row][col] = currentPlayer;

                moves--;

                if (hasPlayerWon()) {

                    System.out.println("\nPlayer " + currentPlayer + " wins!");

                    gameEnded = true;

                } else if (isBoardFull()) {

                    System.out.println("\nThe game is a draw!");

                    gameEnded = true;

                } else {

                    currentPlayer = (currentPlayer == 'X') ? 'O' : 'X';

                }

            } else {

                System.out.println("\nInvalid move! Please try again.\nInput number between 1-9 only");

            }

        }

        System.out.println("\nFinal board:");

        printBoard();

        sc.close();

    }

    public boolean isValidMove(int row, int col) {

        if (row < 0 || row >= 3 || col < 0 || col >= 3)

            return false;

        return board[row][col] == ' ';

    }

    public boolean hasPlayerWon() {

        for (int i = 0; i < 3; i++) {

            if (board[i][0] == currentPlayer && board[i][1] == currentPlayer && board[i][2] == currentPlayer)

                return true;

        }

        for (int j = 0; j < 3; j++) {

            if (board[0][j] == currentPlayer && board[1][j] == currentPlayer && board[2][j] == currentPlayer)

                return true;

        }

        if ((board[0][0] == currentPlayer && board[1][1] == currentPlayer && board[2][2] == currentPlayer) ||

            (board[0][2] == currentPlayer && board[1][1] == currentPlayer && board[2][0] == currentPlayer))

            return true;

        return false;

    }

    // public boolean isBoardFull() {

    //     for (int i = 0; i < 3; i++) {

    //         for (int j = 0; j < 3; j++) {

    //             if (board[i][j] == ' ')

    //                 return false;

    //         }

    //     }

    //     return true;

    // }

    public boolean isBoardFull() {

        return moves == 0;

    }

    public void printBoard() {

        for (int i = 0; i < 3; i++) {

            System.out.println("-------------");

            for (int j = 0; j < 3; j++) {

                System.out.print("| " + board[i][j] + " ");

            }

            System.out.println("|");

        }

    }

    public static void main(String[] args) {

        XoGame game = new XoGame('O');

        game.playGame();

    }

}

Output

