

# OBJECT ORIENTED PROGRAMMING

ANSWER SCRIPT

SIKANDAR AYOUB | BS.SE | ID: 16524

DATE: 9<sup>TH</sup>/06/2020

## Q 1.

ANSWER:

### INTRODUCTION OF TIC-TAC-TOE GAME:

Tic-Tac-Toe is a very common game that is fairly easy to play. The rules of the game are simple and well-known. Because of these things, Tic-Tac-Toe is fairly easy to code up.

### CODE OF TIC TAC TOE GAME JAVA:

```
import java.util.Scanner;

/*
 * Tic-Tac-Toe
 */

public class TicTacToe {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        int a = 1;

        int n = 0;

        Boolean player1 = false;

        Boolean player2 = false;

        Boolean positionMessage = false;

        Boolean quadFound = false;

        String answer;

        String player;
```

```

String quadrants[] = { "Q1", "Q2", "Q3", "Q4", "Q5", "Q6", "Q7", "Q8", "Q9" };
String targetQuads[] = { " ", " ", " ", " ", " ", " ", " ", " ", " " };

System.out.println("---*** TIC-TAC-TOE ***---");
while (a <= 9) {
    // Resets boolean variables for each iteration
    quadFound = false;
    positionMessage = false;

    // Alternate player turns
    if (n % 2 == 0) {
        player = "x";
        System.out.println("It's your turn Player 1. (X)");
    } else {
        player = "o";
        System.out.println("It's your turn Player 2. (O)");
    }

    // Input player quadrant
    System.out.println("What quadrant do you want? (Q1-Q9)");
    answer = input.nextLine();

    int i;
    outerloop:
    for (i = 0; i < quadrants.length; i++) {
        if (answer.equals(quadrants[i]) && " " !=
targetQuads[i].intern()) {

```

```

        positionMessage = true;
        quadFound = true;
        a--;
        break outerloop;
    }
    if (answer.equalsIgnoreCase(quadrants[i])) {
        targetQuads[i] = player;
        quadFound = true;
    }
}
/*
 * Checks player win conditions
 * after every turn of the game
 */

// Horizontal wins P1
if (targetQuads[0] == "x" && targetQuads[1] == "x" &&
targetQuads[2] == "x") {
    player1 = true;
    a = 9;
}

if (targetQuads[3] == "x" && targetQuads[4] == "x" &&
targetQuads[5] == "x") {
    player1 = true;
    a = 9;
}

```

```

        if (targetQuads[6] == "x" && targetQuads[7] == "x" &&
targetQuads[8] == "x") {
            player1 = true;
            a = 9;
        }
        // Vertical wins P1
        if (targetQuads[0] == "x" && targetQuads[3] == "x" &&
targetQuads[6] == "x") {
            player1 = true;
            a = 9;
        }
        if (targetQuads[1] == "x" && targetQuads[4] == "x" &&
targetQuads[7] == "x") {
            player1 = true;
            a = 9;
        }
        if (targetQuads[2] == "x" && targetQuads[5] == "x" &&
targetQuads[8] == "x") {
            player1 = true;
            a = 9;
        }
        // Diagonal wins P1
        if (targetQuads[0] == "x" && targetQuads[4] == "x" &&
targetQuads[8] == "x") {
            player1 = true;
            a = 9;
        }

```

```

        if (targetQuads[2] == "x" && targetQuads[4] == "x" &&
targetQuads[6] == "x") {

            player1 = true;

            a = 9;

        }

// Horizontal wins P2

        if (targetQuads[0] == "o" && targetQuads[1] == "o" &&
targetQuads[2] == "o") {

            player2 = true;

            a = 9;

        }

        if (targetQuads[3] == "o" && targetQuads[4] == "o" &&
targetQuads[5] == "o") {

            player2 = true;

            a = 9;

        }

        if (targetQuads[6] == "o" && targetQuads[7] == "o" &&
targetQuads[8] == "o") {

            player2 = true;

            a = 9;

        }

// Vertical wins P2

        if (targetQuads[0] == "o" && targetQuads[3] == "o" &&
targetQuads[6] == "o") {

            player2 = true;

            a = 9;

        }

```

```

        if (targetQuads[1] == "o" && targetQuads[4] == "o" &&
targetQuads[7] == "o") {

            player2 = true;

            a = 9;

        }

        if (targetQuads[2] == "o" && targetQuads[5] == "o" &&
targetQuads[8] == "o") {

            player2 = true;

            a = 9;

        }

        // Diagonal wins P2

        if (targetQuads[0] == "o" && targetQuads[4] == "o" &&
targetQuads[9] == "o") {

            player2 = true;

            a = 9;

        }

        if (targetQuads[2] == "o" && targetQuads[4] == "o" &&
targetQuads[6] == "o") {

            player2 = true;

            a = 9;

        }

        // Prints array field
        for (i = 0; i < targetQuads.length; i++) {

            System.out.print(targetQuads[i] + ",");

            if ((i + 1) % 3 == 0)

                System.out.println("");

```

```

    }

    // Prints out error messages if the player does not follow rules

    if(positionMessage == true) {
        System.out.println("That position is already used.");
    }

    if(quadFound == false && positionMessage == false) {
        System.out.println("The string " + "" + answer + "" + " did
not match any quadrant number. (Q1-Q9)");
        a--;
    }

    n++;
    a++;
}

// Checks the winner
if (player1) {
    System.out.println("Player 1 wins!");
    System.exit(0);
}

if (player2) {
    System.out.println("Player 2 wins!");
    System.exit(0);
}

```

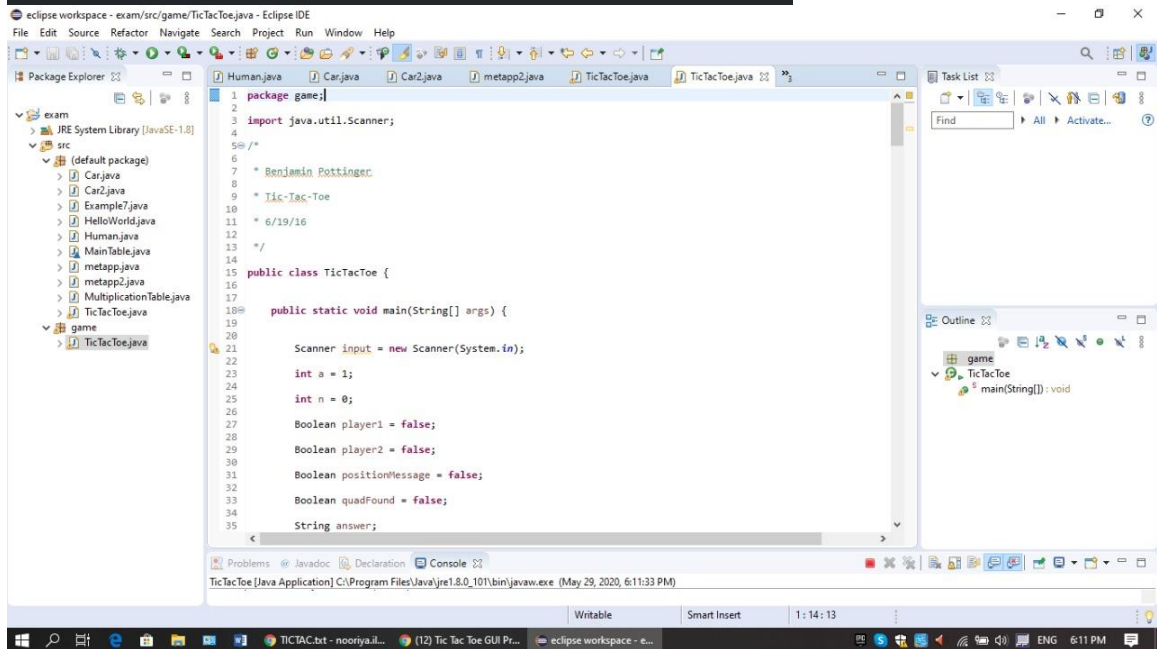


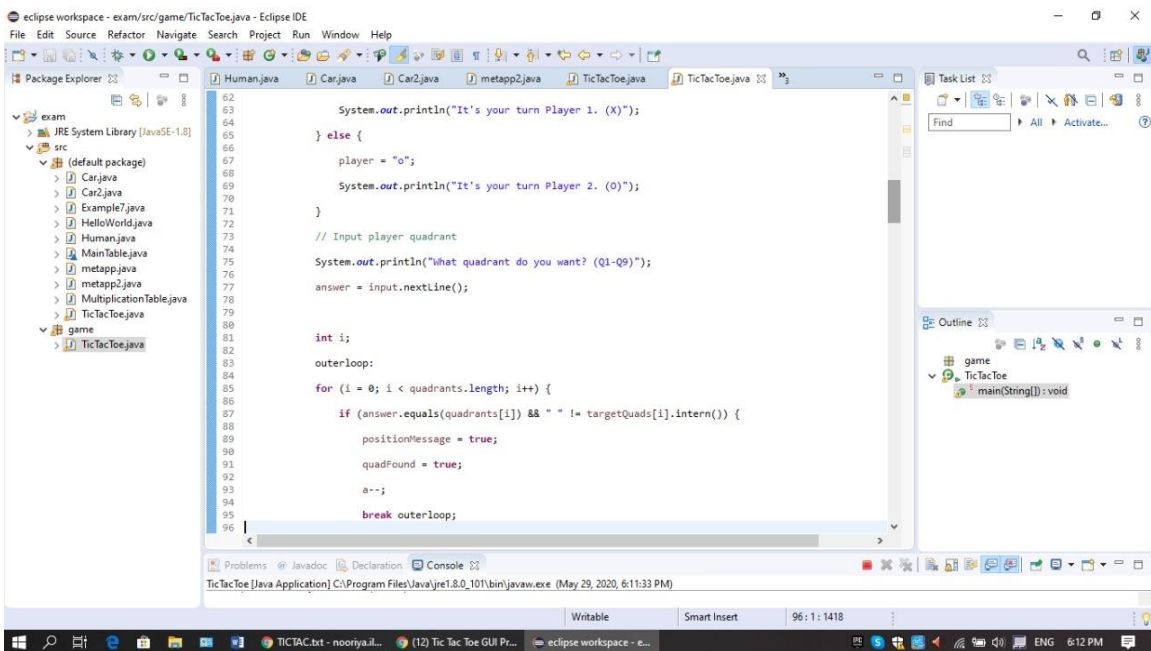
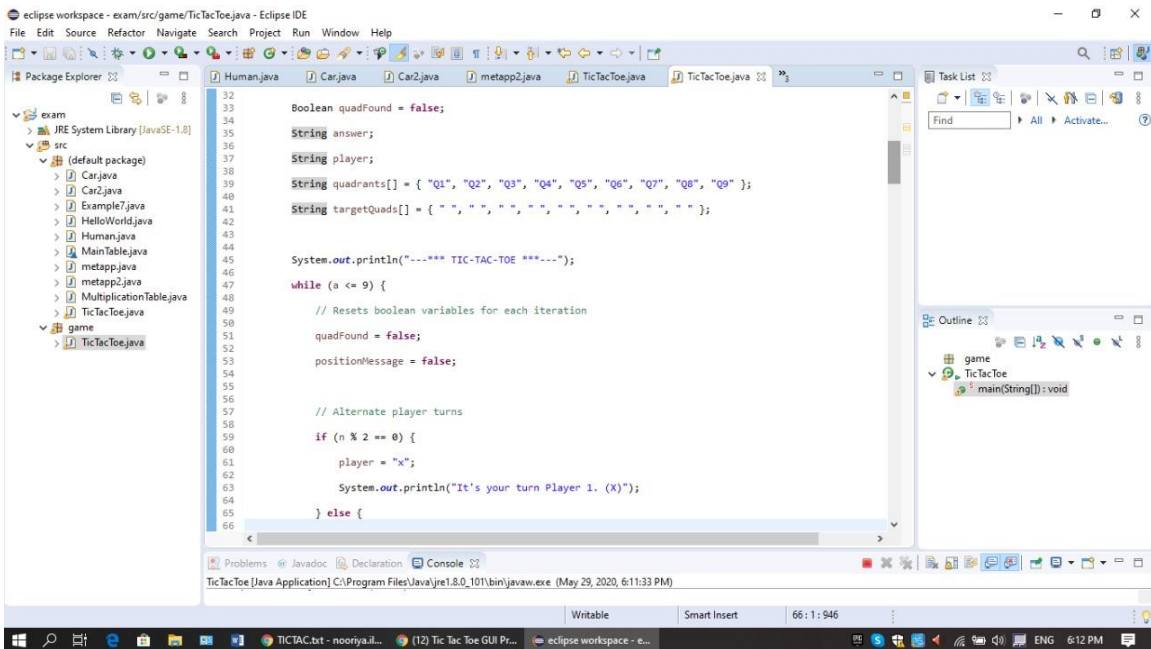
```

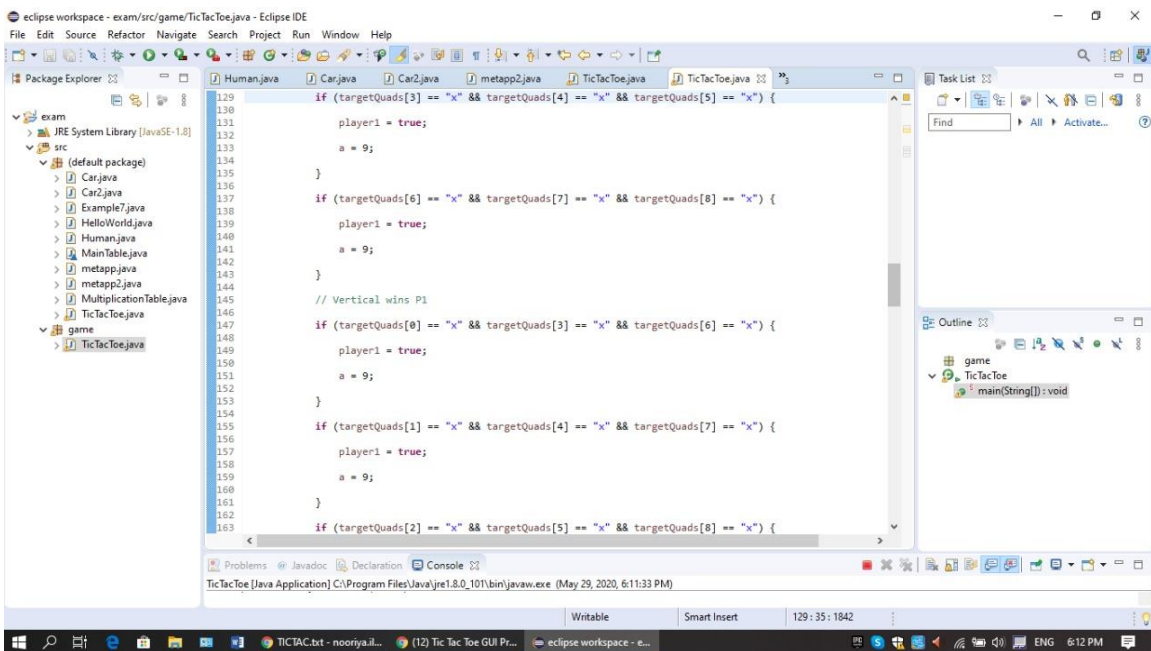
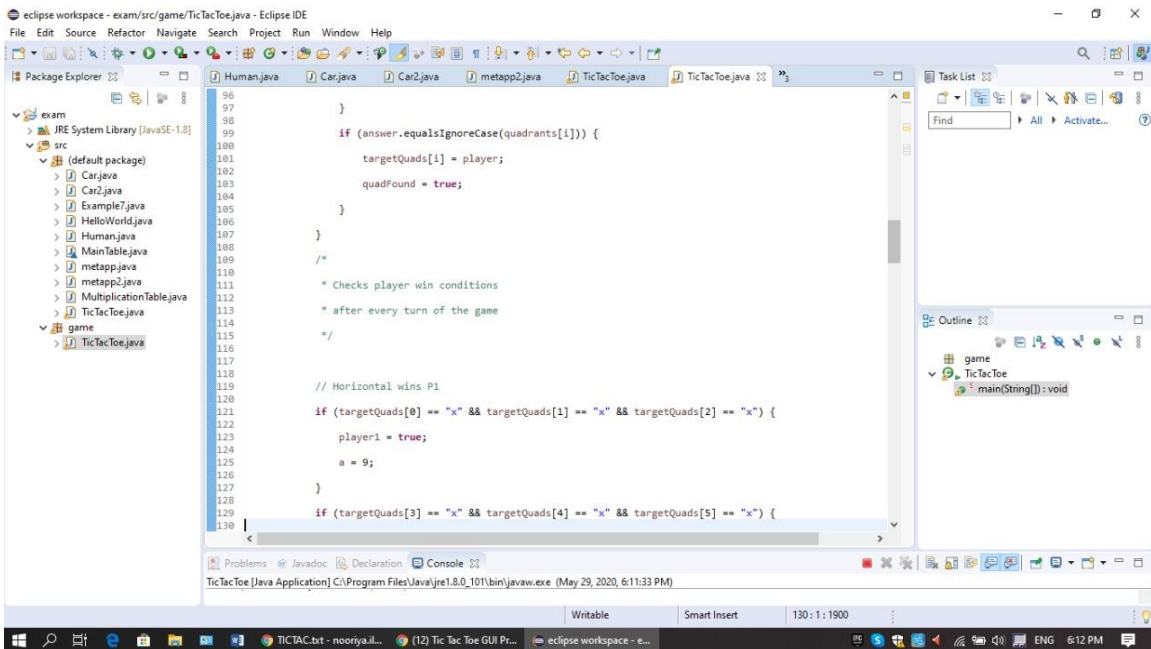
    }
else {
    System.out.println("It's a draw!");
    System.exit(0);
}
}
}
}

```

## SCREENSHOTS OF PROFRAM:





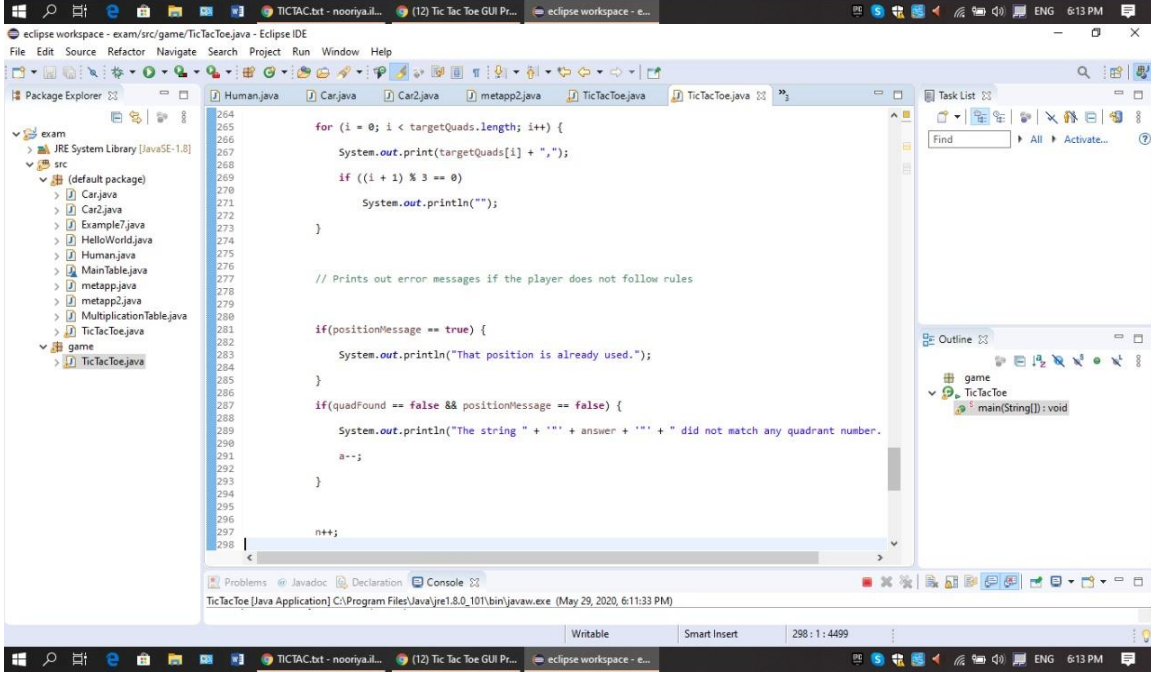
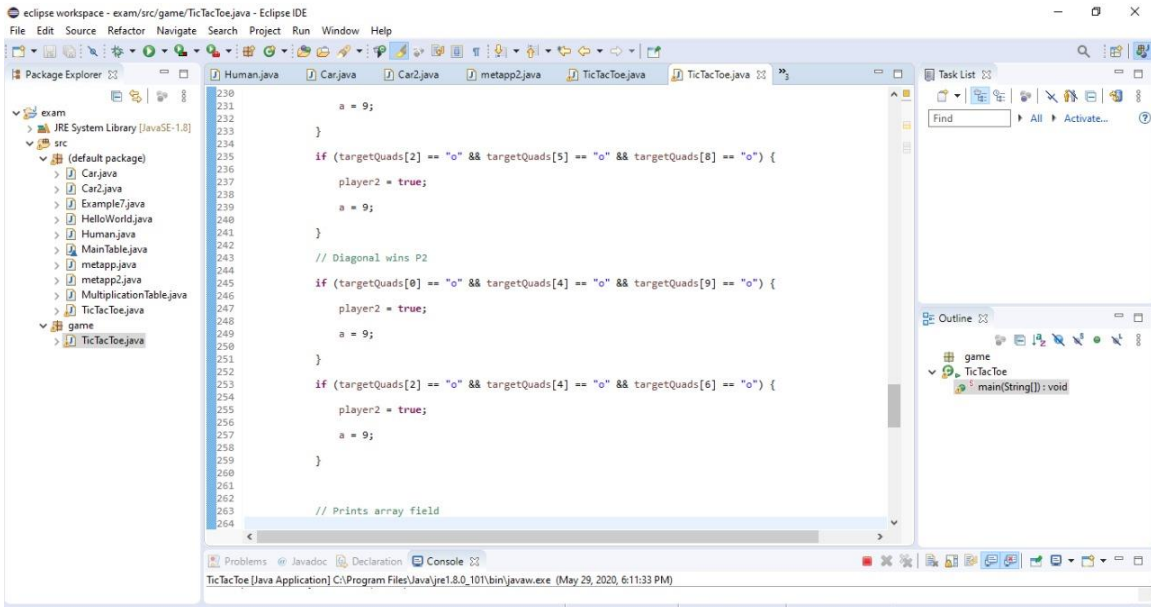


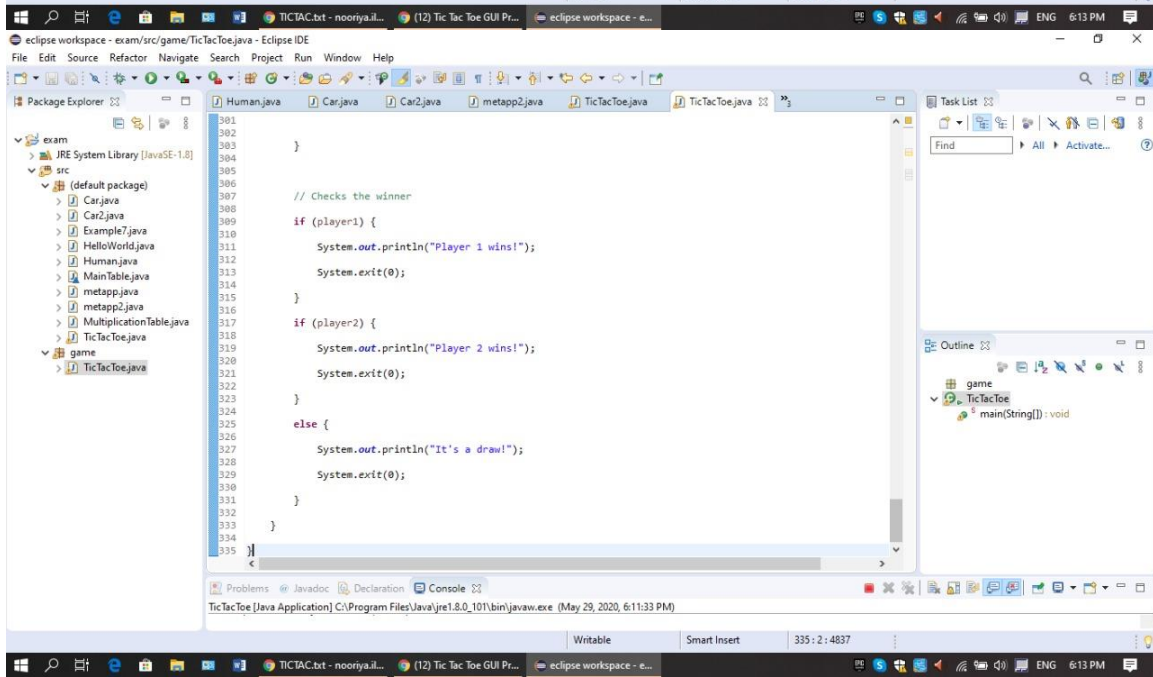
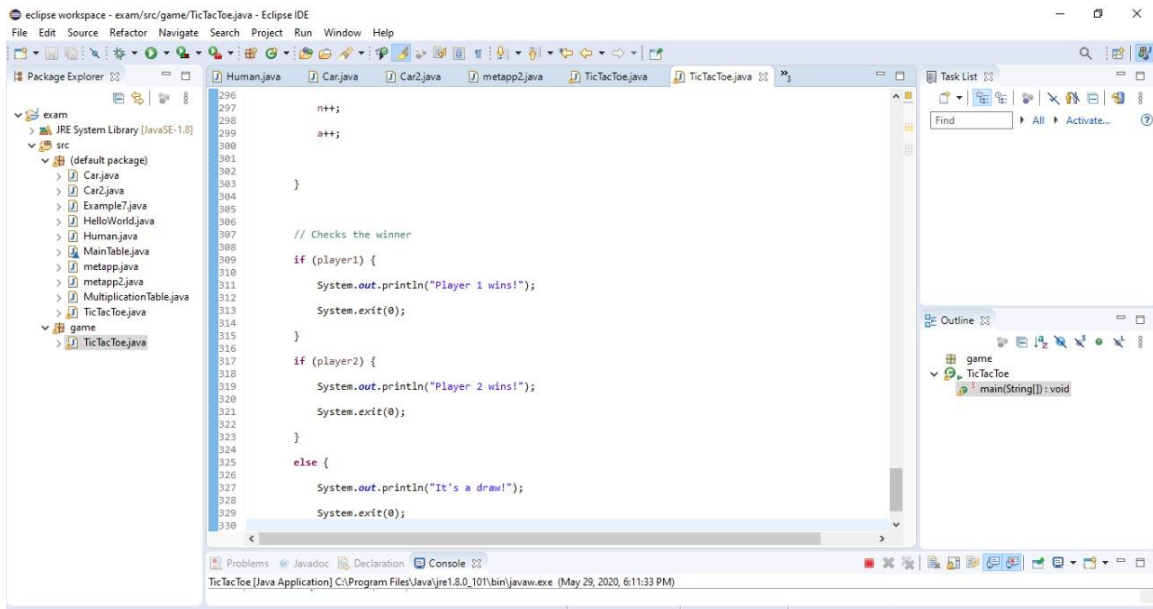
The screenshot shows the Eclipse IDE with the following code in TicTacToe.java:

```
163     if (targetQuads[2] == "x" && targetQuads[5] == "x" && targetQuads[8] == "x") {
164         player1 = true;
165     }
166     a = 9;
167 }
168 // Diagonal wins P1
169
170     if (targetQuads[0] == "x" && targetQuads[4] == "x" && targetQuads[8] == "x") {
171         player1 = true;
172     }
173     a = 9;
174 }
175 // Horizontal wins P2
176
177     if (targetQuads[2] == "x" && targetQuads[4] == "x" && targetQuads[6] == "x") {
178         player1 = true;
179     }
180     a = 9;
181 }
182 // Horizontal wins P1
183
184     if (targetQuads[0] == "o" && targetQuads[1] == "o" && targetQuads[2] == "o") {
185         player2 = true;
186     }
187     a = 9;
188 }
189 // Horizontal wins P2
190
191     if (targetQuads[0] == "o" && targetQuads[1] == "o" && targetQuads[2] == "o") {
192         player2 = true;
193     }
194     a = 9;
195 }
196 // Horizontal wins P1
197
```

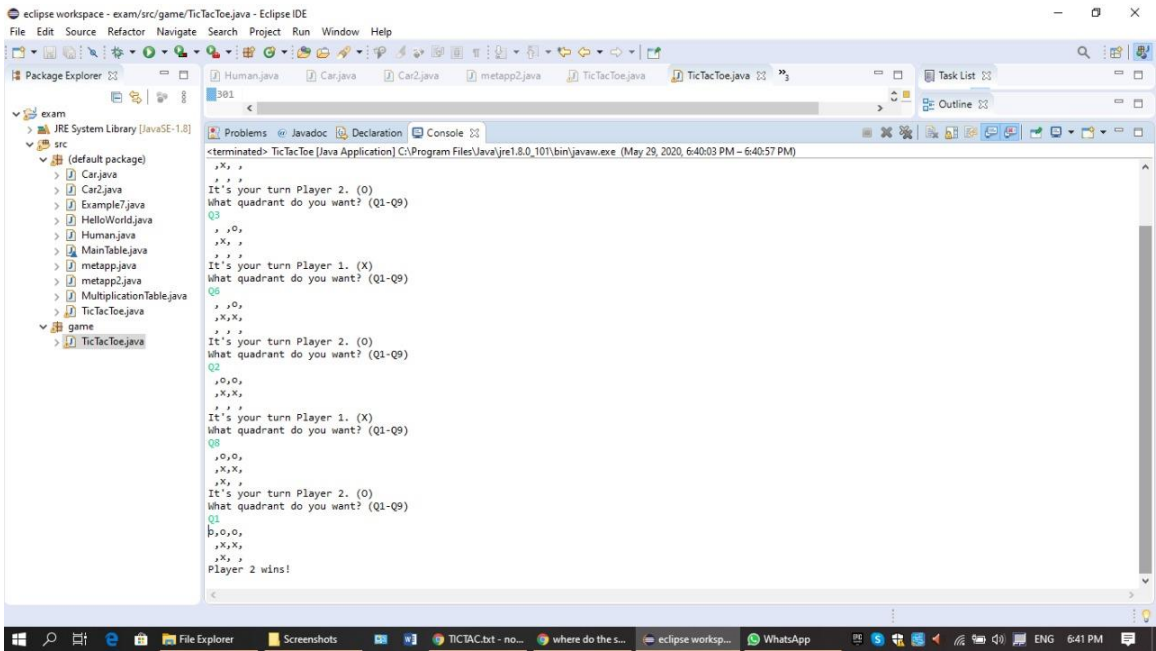
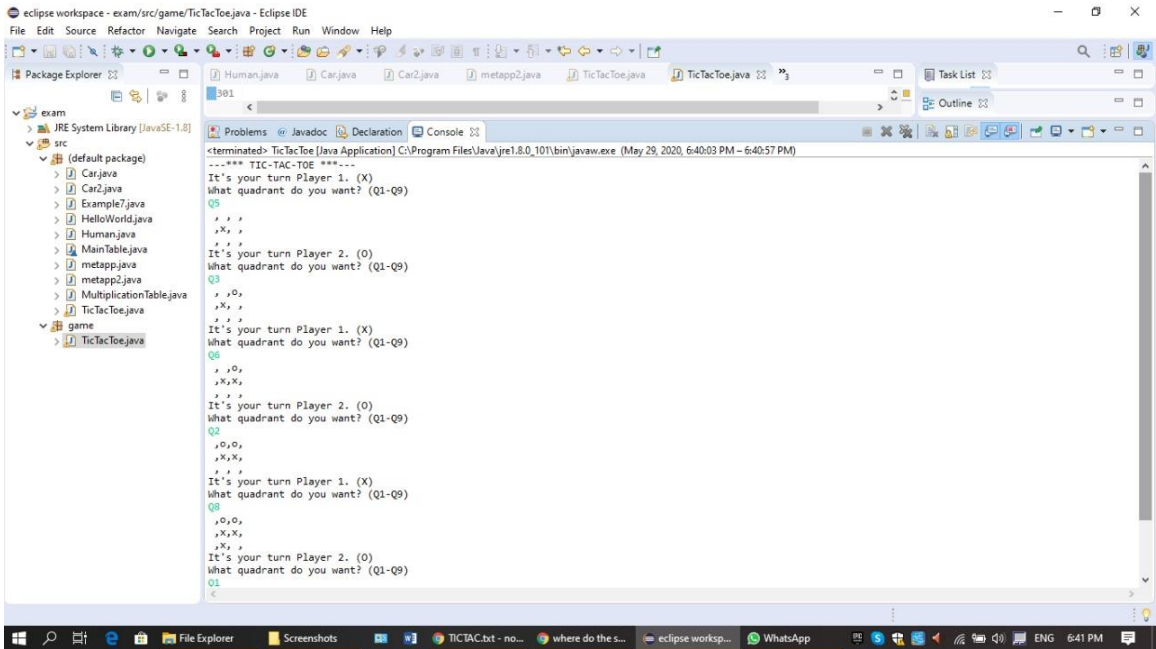
The screenshot shows the Eclipse IDE with the following code in TicTacToe.java:

```
196     a = 9;
197 }
198 // Vertical wins P1
199
200     if (targetQuads[3] == "o" && targetQuads[4] == "o" && targetQuads[5] == "o") {
201         player2 = true;
202     }
203     a = 9;
204 }
205 // Vertical wins P2
206
207     if (targetQuads[6] == "o" && targetQuads[7] == "o" && targetQuads[8] == "o") {
208         player2 = true;
209     }
210     a = 9;
211 }
212 // Vertical wins P1
213
214     if (targetQuads[0] == "o" && targetQuads[3] == "o" && targetQuads[6] == "o") {
215         player2 = true;
216     }
217     a = 9;
218 }
219 // Vertical wins P2
220
221     if (targetQuads[1] == "o" && targetQuads[4] == "o" && targetQuads[7] == "o") {
222         player2 = true;
223     }
224     a = 9;
225 }
226 // Vertical wins P1
227
228     if (targetQuads[1] == "o" && targetQuads[4] == "o" && targetQuads[7] == "o") {
229         player2 = true;
230     }
231     a = 9;
232 }
233 // Vertical wins P2
234
```





## SCREENSHOTS OF OUTPUT:



## OUTPUT DETAIL/DETAIL OF PROGRAM:

There are many ways to implement a game of Tic-Tac-Toe in Java. We will be coding a text-based version of Tic-Tac-Toe. Our Tic-Tac-Toe will start out by printing out “it’s your turn player1” and then asking for input from the first player (“what quadrant do you want?”) that will specify which quadrant to place that player’s mark. After placing the mark, we will then ask the other player for their move. That process will be continued until one player wins or the match draws. The input that will be taken in to specify where to place a mark will be in the format of Q1-Q9, which specify the quadrant where the mark is to be placed.

**END**