```java
/**
Board can represents 2D 3*3 array for TicTacToe game.
It can check if someone wins or a cat's game.
It can check if a square has been chosen.
It can also mark an X or O from the player's choice.
*/

class Board
{
    private int [][] myBoard = new int [3][3];

//  Create a 3 by 3 array and use for a tic tac toe board.
    public Board()
    {
        for(int row = 0; row < 3; row++)
        {
            for(int column = 0; column < 3; column++)
            {
                myBoard [row] [column] = 0;
            }
        }
    }

    public int [] [] copyBoard()
    {
        return myBoard;
    }

    /*
    markFirst makes places a 2 accumulation for X
    */
    public void markFirst(int row, int column)
    {
        myBoard [row] [column] = 2;
    }
```

```
36
37      /*
38      markSecond makes places a 1 accumulation for O
39      */
40      public void markSecond(int row, int column)
41      {
42          myBoard [row] [column] = 1;
43      }
44
45      /*
46      elementMarked returns a true if the space has been taken
47      */
48      public boolean elementMarked(int row, int column)
49      {
50          if(myBoard [row] [column] == 0) return false;
51          else return true;
52      }
53
54      /*
55      Win constructor checks if someone wins.
56      Here are the meanings of each return type
57      'N' means no winner;
58      'X' means X won;
59      'O' means O won;
60      'C' means a C's game.
61      */
62      public char win()
63      {
64          char winner = 'N';
65          int catCheck = 1;
66
67          // Check the columns
68          for(int column = 0; column < 3; column++)
69          {
70              int accumulation = myBoard [0] [column] * myBoard [1] [column] * myBoard [2] [column];
```

```
71              if(accumulation == 8) // 2*2*2 = 8, a win for X
72              {
73                  winner = 'X';
74                  break;
75              }
76              if(accumulation == 1) // 1*1*1 = 1, a win for O
77              {
78                  winner = 'O';
79                  break;
80              }
81          }
82
83          if(winner != 'N') return winner;
84
85          // Check the rows
86          for(int row = 0; row < 3; row++)
87          {
88              int accumulation = myBoard [row] [0] * myBoard [row] [1] * myBoard [row] [2];
89              if(accumulation == 8)
90              {
91                  winner = 'X';
92                  break;
93              }
94              if(accumulation == 1)
95              {
96                  winner = 'O';
97                  break;
98              }
99          }
100
101         if(winner != 'N') return winner;
102
103         // Check one diagonal
104         int accumulation = myBoard [0] [0] * myBoard [1] [1] * myBoard [2] [2];
105         if(accumulation == 1) winner = 'O';
```

```
106         if(accumulation == 8) winner = 'X';
107 
108         // Check the other diagonal
109         accumulation = myBoard [0] [2] * myBoard [1] [1] * myBoard [2] [0];
110         if(accumulation == 1) winner = 'O';
111         if(accumulation == 8) winner = 'X';
112 
113         // If nobody's won, Check for a cat's game
114         if(winner == 'N')
115         {
116             for(int row = 0; row < 3; row++)
117             {
118                 for(int column = 0; column < 3; column++)
119                 {
120                     catCheck *= myBoard [row] [column];
121                 }
122             }
123             // any empty space is a zero. So product is zero if there is space left.
124             if(catCheck != 0) winner = 'C';
125         }
126 
127         return winner;
128     }
129 
130     /*
131     toString enables printing out of the board
132     */
133     public String toString()
134     {
135         String printBoard = "";
136         char XorO;
137         int position = 49;   // In ASCII, 49 stands for number 1
138 
139         for(int row = 0; row < 3; row++)
140         {
```

```
141             for(int column = 0; column < 3; column++)
142             {
143                 if(myBoard[row] [column] == 1)
144                     // In ASCII, 79 stands for an O  (78+1)
145                     XorO = (char) (myBoard [row] [column] + 78);
146                 else
147                     if(myBoard[row] [column] == 2)
148                         // In ASCII, 88 stands for an X  (86+2)
149                         XorO = (char) (myBoard [row] [column] + 86);
150                     else
151                         XorO = (char) (position);
152 
153                     position++;
154 
155                 printBoard = printBoard + XorO + " ";
156             }
157             printBoard = printBoard + "\n" ; // starts a new line at the end of a row
158         }
159         return printBoard;
160     }
161 }
162 
```