

IQRA NATIONAL UNIVERSITY

OBJECT ORIENTED PROGRAMMING

ASSIGNMENT. 2

NAME: NOORIYA ILYAS

ID: 15770

SEMESTER: 2nd

BS(SE)

SUBMITTED TO: SIR AYUB

QUESTION: Create the game Tic tac toe using java and explain in detail.

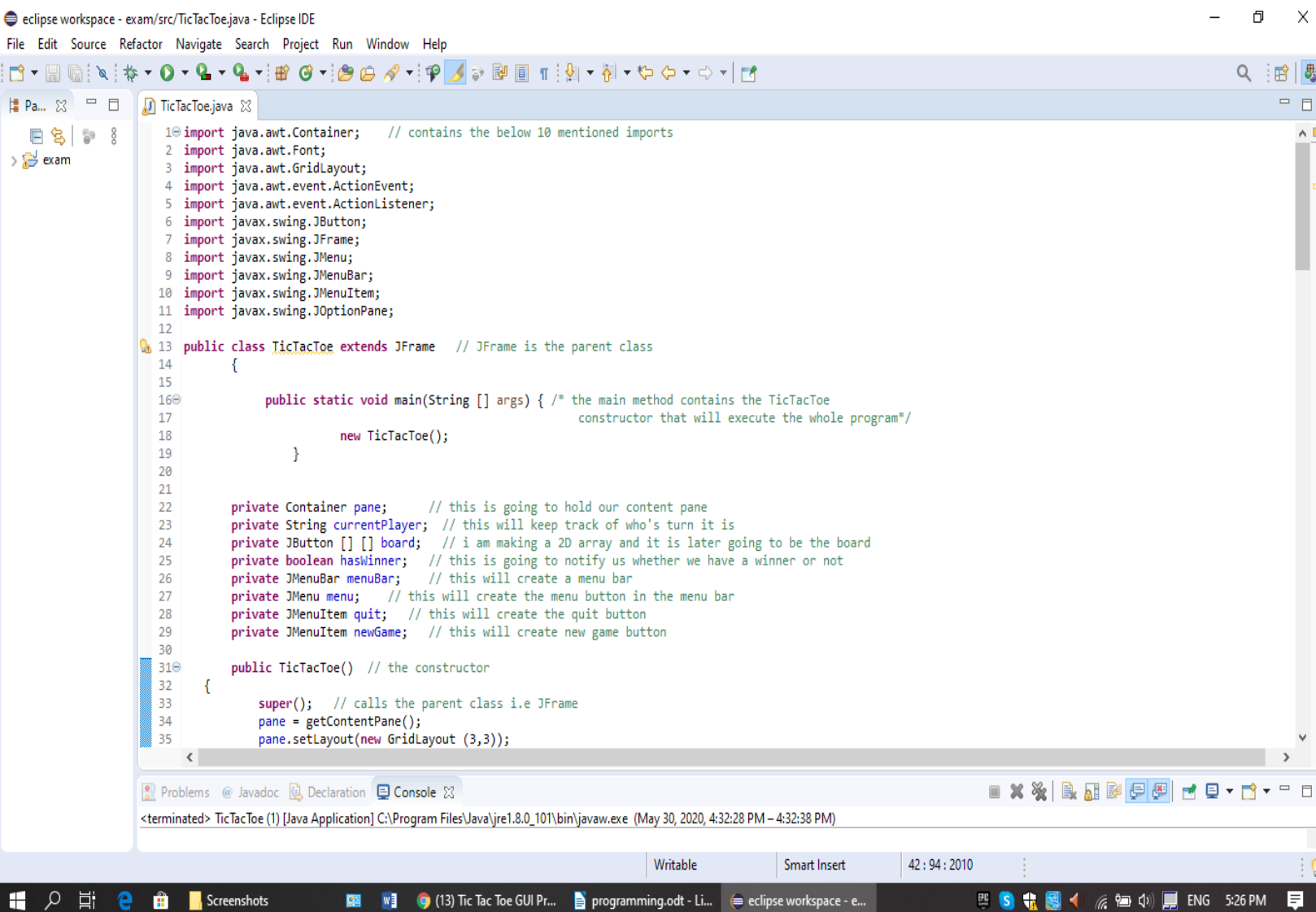
ANSWER: In the following program, I created the Tic tac toe game using java and Swing. Swing is a package that lets you create applications that use a graphical user interface or GUI instead of a dull console interface. The Swing API provides programmers many different classes for creating various types of user interface elements. A java interface is an abstract type used to select a set of abstract methods for classes to implement. When a class implements an interface, it must inherit all of the abstract methods declared within the interface.

I also wrote comments within the code to explain it better.

I made a public class TicTacToe (child class) , which extends to JFrame (parent class). In java, it is possible to inherit attributes and methods from one class to another. And to inherit from a class, we use the **extend** keyword. Basically, the keyword **extends** is used to extend the functionality of a class. The class name before **extends** identifies the child class and the class name after it identifies the parent class.

The main method consists of the instance/constructor of the class TicTacToe which contains the whole code written to make the game. Our main method would eventually execute the program. I declared 8 variables that we would need to use later in the code.

The keyword **super** refers to parent class objects. It is used to call parent class methods, and to access the parent class constructor.



```
eclipse workspace - exam/src/TicTacToe.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
TicTacToe.java
1 import java.awt.Container; // contains the below 10 mentioned imports
2 import java.awt.Font;
3 import java.awt.GridLayout;
4 import java.awt.event.ActionEvent;
5 import java.awt.event.ActionListener;
6 import javax.swing.JButton;
7 import javax.swing.JFrame;
8 import javax.swing.JMenu;
9 import javax.swing.JMenuBar;
10 import javax.swing.JMenuItem;
11 import javax.swing.JOptionPane;
12
13 public class TicTacToe extends JFrame // JFrame is the parent class
14 {
15
16     public static void main(String [] args) { /* the main method contains the TicTacToe
17                                     constructor that will execute the whole program*/
18         new TicTacToe();
19     }
20
21
22     private Container pane; // this is going to hold our content pane
23     private String currentPlayer; // this will keep track of who's turn it is
24     private JButton [] [] board; // i am making a 2D array and it is later going to be the board
25     private boolean hasWinner; // this is going to notify us whether we have a winner or not
26     private JMenuBar menuBar; // this will create a menu bar
27     private JMenu menu; // this will create the menu button in the menu bar
28     private JMenuItem quit; // this will create the quit button
29     private JMenuItem newGame; // this will create new game button
30
31     public TicTacToe() // the constructor
32     {
33         super(); // calls the parent class i.e JFrame
34         pane = getContentPane();
35         pane.setLayout(new GridLayout (3,3));

```

Problems @ Javadoc Declaration Console

<terminated> TicTacToe (1) [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (May 30, 2020, 4:32:28 PM - 4:32:38 PM)

Writable Smart Insert 42 : 94 : 2010

Windows Taskbar: Screenshots, (13) Tic Tac Toe GUI Pr..., programming.odt - Li..., eclipse workspace - e...

System Tray: ENG 5:26 PM

```
eclipse workspace - exam/src/TicTacToe.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

TicTacToe.java
34 pane = getContentPane();
35 pane.setLayout(new GridLayout (3,3));
36 setTitle("Tic Tac Toe");
37 setSize(500,500);
38 setDefaultCloseOperation(); // this brings up a notification when a player wins
39 setVisible(true); // enables us to see the game board
40 currentPlayer = "x"; // the first player will be "x"
41 board = new JButton[3][3];
42 hasWinner = false; // the game has just started, so there is no winner yet
43 initializeBoard(); // calling the methods
44 initializeMenuBar();
45 }
46 private void initializeMenuBar() // this method is for initializing the menu bar
47 {
48     menuBar = new JMenuBar();
49     menu = new JMenu("File");
50     newGame = new JMenuItem("New Game"); // making the new game button functional
51     newGame.addActionListener (new ActionListener()
52     {
53         public void actionPerformed (ActionEvent e)
54         {
55             resetBoard(); // calling the reset board method
56         }
57     });
58 }
59
60 quit = new JMenuItem("Quit"); // quit game button will just shut down the game
61 quit.addActionListener(new ActionListener()
62 {
63     public void actionPerformed(ActionEvent e)
64     {
65         System.exit(0); // this will terminate the program
66     }
67 }
68 });

Problems @ Javadoc Declaration Console
<terminated> TicTacToe (1) [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (May 30, 2020, 4:32:28 PM - 4:32:38 PM)

Writable Smart Insert 33: 60 : 1504
```

```
eclipse workspace - exam/src/TicTacToe.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

TicTacToe.java
64
65     System.exit(0); // this will terminate the program
66 }
67 });
68 });
69
70 menu.add(newGame); // adding newGame to the menu
71 menu.add(quit); //adding quit to the menu
72 menuBar.add(menu); //adding menu to the menu bar
73 setJMenuBar(menuBar); // setting the menu bar to the JFrame
74 }
75 private void resetBoard() // this method will reset the game the game board when the user will click the new game button
76 {
77     currentPlayer = "x";
78     hasWinner = false;
79     for (int i = 0; i < 3; i++) { // resetting the JButtons
80         for(int j = 0; j < 3; j++) {
81             board[i][j].setText(""); // empty string, because we want the board to be empty
82         }
83     }
84 }
85
86 private void initializeBoard()
87 { //this method is for initializing the game board
88     for(int i = 0; i < 3; i++) {
89         for(int j = 0; j < 3; j++) {
90             JButton btn = new JButton ();
91             btn.setFont(new Font (Font.SANS_SERIF,Font.BOLD,100)); // here i set the font and size of the "x" and "0" in the game
92             board[i][j] = btn;
93             btn.addActionListener(new ActionListener()
94             {
95                 public void actionPerformed(ActionEvent e)
96                 {
97                     if(((JButton)e.getSource()).getText().contentEquals("") &&
98                     hasWinner == false) {
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

Problems @ Javadoc Declaration Console
<terminated> TicTacToe (1) [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (May 30, 2020, 4:32:28 PM - 4:32:38 PM)

Writable Smart Insert 33: 60 : 1504
```

```
eclipse workspace - exam/src/TicTacToe.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

TicTacToe.java
95
96
97
98     public void actionPerformed(ActionEvent e)
99     {
100         if(((JButton)e.getSource()).getText().contentEquals("") &&
101             hasWinner == false) {
102             btn.setText(currentPlayer);
103             hasWinner();
104             togglePlayer();
105         }
106     }
107     pane.add(btn);
108 }
109
110 private void togglePlayer() // this method will switch the player after 1 move
111 {
112     if(currentPlayer.contentEquals("X"))
113         currentPlayer = "O";
114     else
115         currentPlayer = "X";
116 }
117
118 private void hasWinner() // this method will determine if we have a winner
119 {
120     if(board[0][0].getText().equals(currentPlayer) && board [1][0]. getText().equals(currentPlayer) && board [2][0]. getText() .equals(currentPlayer)) {
121         JOptionPane.showMessageDialog(null, "Player "+ currentPlayer+ " has won");
122         hasWinner = true;
123     }
124     else if (board[0][1].getText().equals(currentPlayer) && board [1][1]. getText().equals(currentPlayer) && board [2][1]. getText() .equals(currentPlayer))
125         JOptionPane.showMessageDialog(null, "Player "+ currentPlayer+ " has won");
126         hasWinner = true;
127     }
128     else if (board[0][2].getText().equals(currentPlayer) && board [1][2]. getText().equals(currentPlayer) && board [2][2]. getText() .equals(currentPlayer))
129         JOptionPane.showMessageDialog(null, "Player "+ currentPlayer+ " has won");
130 }
131 }
132 }
133 }
134 }
135 }
136 }
137 }
138 }
139 }
140 }
141 }
142 }
143 }
144 }
145 }
146 }
147 }
148 }
149 }
150 }
151 }
152 }
153 }
154 }
155 }
156 }
157 }
158 }

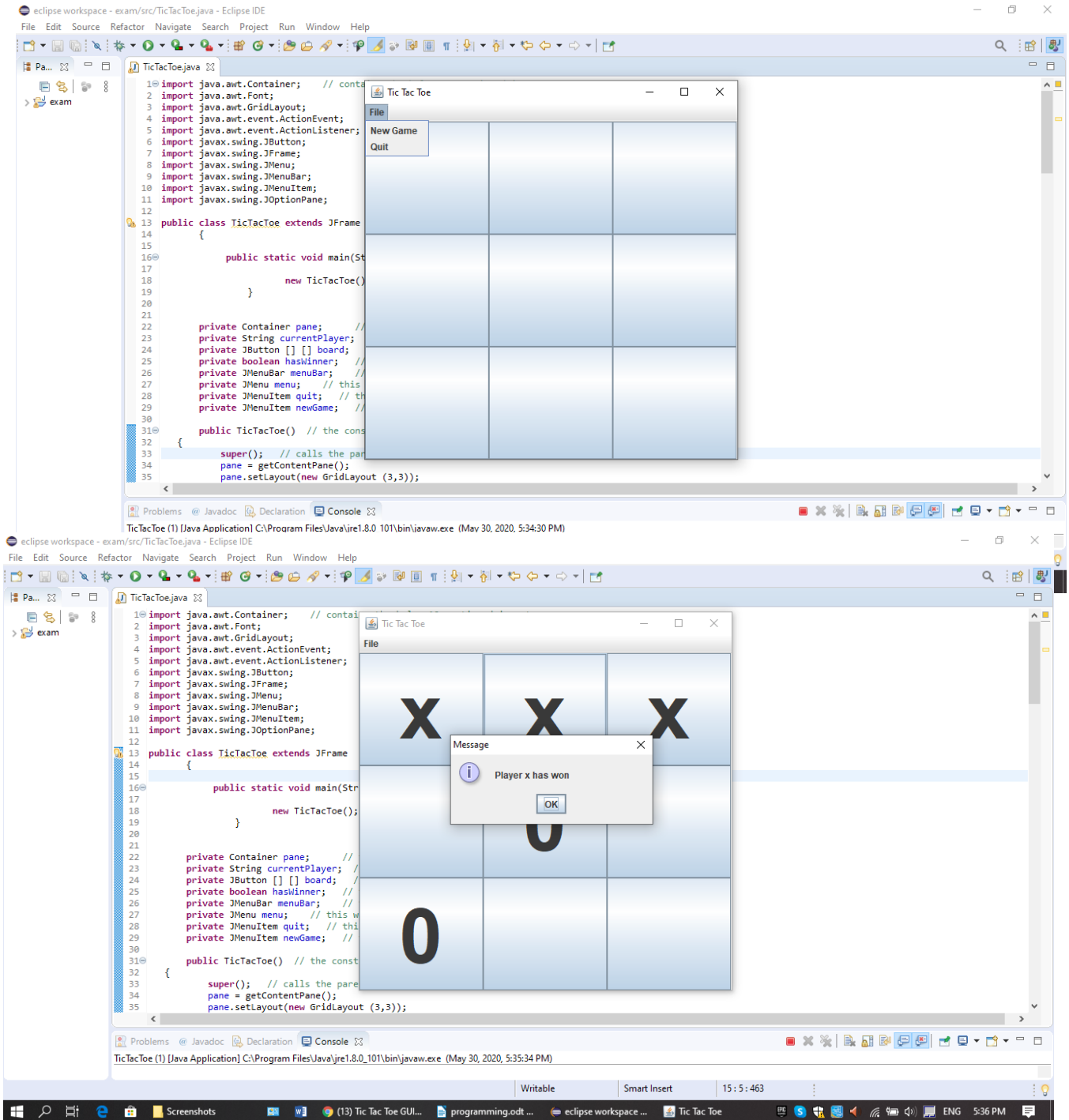
<terminated> TicTacToe (1) [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (May 30, 2020, 4:32:28 PM - 4:32:38 PM)
```

```
eclipse workspace - exam/src/TicTacToe.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

TicTacToe.java
124
125
126
127
128     else if (board[0][1].getText().equals(currentPlayer) && board [1][1]. getText().equals(currentPlayer) && board [2][1]. getText() .equals(currentPlayer))
129         JOptionPane.showMessageDialog(null, "Player "+ currentPlayer+ " has won");
130         hasWinner = true;
131     }
132     else if (board[0][2].getText().equals(currentPlayer) && board [1][2]. getText().equals(currentPlayer) && board [2][2]. getText() .equals(currentPlayer))
133         JOptionPane.showMessageDialog(null, "Player "+ currentPlayer+ " has won");
134         hasWinner = true;
135     }
136     else if (board[0][0].getText().equals(currentPlayer) && board [0][1]. getText().equals(currentPlayer) && board [0][2]. getText() .equals(currentPlayer))
137         JOptionPane.showMessageDialog(null, "Player "+ currentPlayer+ " has won");
138         hasWinner = true;
139     }
140     else if (board[1][0].getText().equals(currentPlayer) && board [1][1]. getText().equals(currentPlayer) && board [1][2]. getText() .equals(currentPlayer))
141         JOptionPane.showMessageDialog(null, "Player "+ currentPlayer+ " has won");
142         hasWinner = true;
143     }
144     else if (board[2][0].getText().equals(currentPlayer) && board [2][1]. getText().equals(currentPlayer) && board [2][2]. getText() .equals(currentPlayer))
145         JOptionPane.showMessageDialog(null, "Player "+ currentPlayer+ " has won");
146         hasWinner = true;
147     }
148     else if (board[0][0].getText().equals(currentPlayer) && board [1][2]. getText().equals(currentPlayer) && board [1][2]. getText() .equals(currentPlayer))
149         JOptionPane.showMessageDialog(null, "Player "+ currentPlayer+ " has won");
150         hasWinner = true;
151     }
152     else if (board[2][0].getText().equals(currentPlayer) && board [1][2]. getText().equals(currentPlayer) && board [2][1]. getText() .equals(currentPlayer))
153         JOptionPane.showMessageDialog(null, "Player "+ currentPlayer+ " has won");
154         hasWinner = true;
155     }
156 }
157 }
158 }

<terminated> TicTacToe (1) [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (May 30, 2020, 4:32:28 PM - 4:32:38 PM)
```

THE OUTPUT:



- A container is a component that holds and manages other components.
- Three of the useful container types I used are JFrame, JMenu and JOptionPane.
- The JFrame is the top level window on the display. JMenu is inside the menu bar and inside the JMenu, we have JMenuItem's I.e the new game option and quit option.
- The add() method of the container class adds a component to the container.
- Lastly, the ActionListener interface I used was to make the buttons functional. The action listener interface has only 1 method; actionPerformed(). The action listener is notified whenever we click on the button or a menu item. The actionPerformed() method is invoked automatically whenever we click on the registered component.(ActionEvent is a class, and e is the instance of that class which we can use to call it's methods.