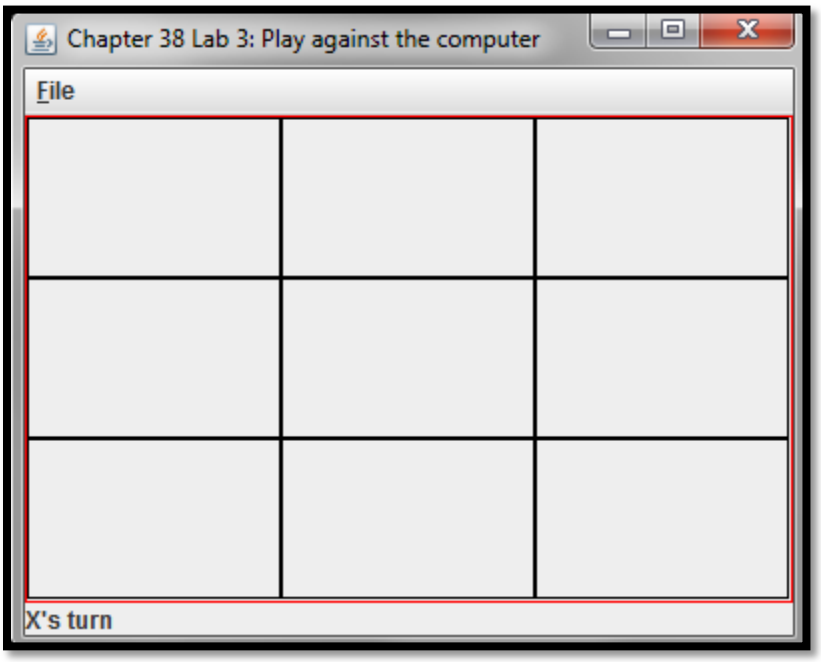
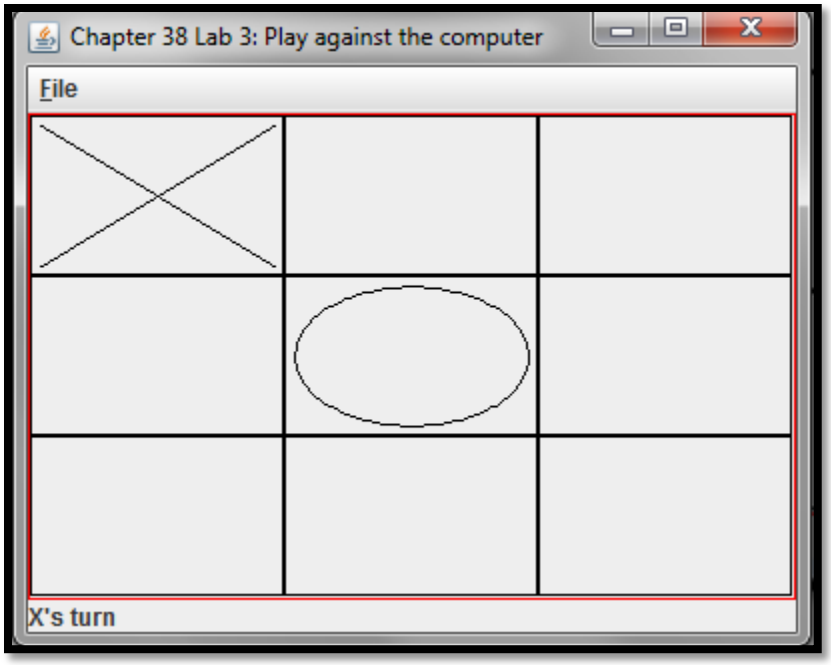
CIT 249: Java II
Chapter 38 Lab3

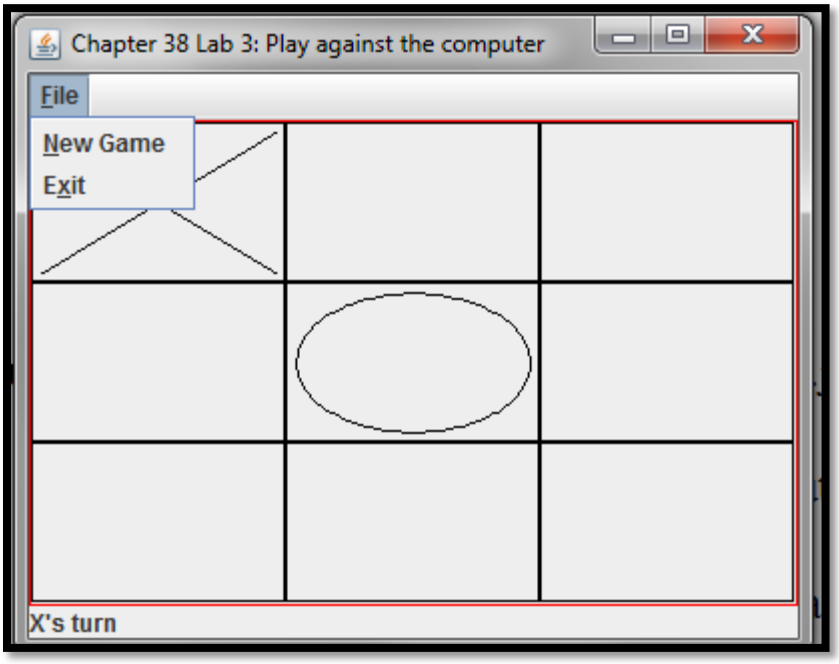
In this lab we will a program that enables a player to play Tic Tac Toe against the computer. Add a File menu with two items, New Game and Exit. When you click New Game, it displays a dialog box. From this dialog box, you can decide whether to let the computer go first. When run the application will display as:



If you click on a square you will put an X on the square and the program will decide on a place for O. As in:



We also have a small menu here as in:



Let's get started!

1. Open a new document window and save the file as Ch38Lab3.java. Purpose is:

   /* Purpose is to create a Tic-Tac-Toe game against the computer */

2. Type the import statements that will import all classes in the java.awt, java.awt.event, javax.swing.border and javax.swing packages.
3. Type the class header and opening brace having the class extend JApplet and implement the ActionListener interface.
4. First we will create our components and variables by typing:

```
//declare and/or construct variables and objects
public JLabel jlblStatus = new JLabel("X's turn");
public char turn = 'X';
public Cell[][] c = new Cell[3][3];
public char com = ' ';
public char hum = ' ';
```

   - Notice that we created a multi-dimensional array of Cell objects. We'll create the Cell class later.

5. Next we start our init() method by typing:

```
public void init()
{

   JPanel p = new JPanel(new GridLayout(3,3));
   p.setBorder(new LineBorder(Color.red));
```

   - We create a new panel with 3 rows and 3 columns using GridLayout
   - We set the border color to red.

6. We construct each element in our multi-dimensional array by typing:

```
for(int row = 0; row < 3; row++)
{
    for(int col = 0; col < 3; col++)
    {
      p.add(c[row][col] = new Cell());
    }
```

```
}
```

- We also add each to the panel.

7. We create our menu by typing:

```
JMenuBar mb = new JMenuBar();
this.setJMenuBar(mb);

JMenu file = new JMenu("File");
file.setMnemonic('f');
mb.add(file);

JMenuItem newM = new JMenuItem("New Game");
newM.setMnemonic('n');
newM.addActionListener(this);
file.add(newM);

JMenuItem exit = new JMenuItem("Exit");
exit.setMnemonic('x');
exit.addActionListener(this);
file.add(exit);
```

- The JMenuBar creates the bar where the menu resides.
- The JMenu creates a new menu on the bar. You can create as many as you wish.
- The JMenuItem creates a menu under the JMenu it is clicked. These display as drop-down menus.
- The setMnemonic() method sets the key that is pressed, along with either the Alt or Ctrl key, as a shortcut.
- For the “exit” and “newM” JMenuItem we add a listener. The listener will work the same as when clicking a regular button.
- We add “newM” and “exit” to the JMenu, so they will display when this item is clicked.

8. We add the panel and the label to the frame by typing:

```
add(p, BorderLayout.CENTER);
add(jlblStatus, BorderLayout.SOUTH);
```

9. We set controls for our array by typing:

```
for(int i = 0; i < 3; i++)
{
    for(int j = 0; j < 3; j++)
    {
     c[i][j].token = ' ';
     c[i][j].repaint();
     c[i][j].removeMouseListener(c[i][j]);
     c[i][j].addMouseListener(c[i][j]);
    }
  }
```

- Using loops we first set the token used, which in this case is an empty single quotes. There is a space between ' '
- We repaint the screen to prevent any shadowing.
- We remove MouseListener's from certain elements in the array.
- We add a MouseListener to elements.

10. We set the text of the label and set values for the variables by typing:

```
  jlblStatus.setText("X's turn");
  turn = 'X';
  com = 'O';
  hum = 'X';
 }
```

- We also closed the init() method.

11. The main method is similar to what we have been doing all along. Type:

```
public static void main(String[] args)
{
  Ch38Lab3 applet = new Ch38Lab3();
  JFrame frame = new JFrame();
  frame.setTitle("Chapter 38 Lab 3: Play against the computer");
  frame.add(applet, BorderLayout.CENTER);
  applet.init();
  applet.start();
  frame.setSize(400,320);
```

```
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setLocationRelativeTo(null); // Center the frame
    frame.setVisible(true);
  }
```

12. Next we start our actionPerformed() method for the events. Type:

```
public void actionPerformed(ActionEvent e)
{
    if (e.getSource() instanceof JMenuItem)
    {
      if(e.getActionCommand() == "New Game")
      {
        JOptionPane pane = new JOptionPane("Do you wish to go first?",
                                JOptionPane.QUESTION_MESSAGE,
                                JOptionPane.YES_NO_OPTION);
        JDialog dialog = pane.createDialog(this, "2P");
        dialog.show();
```

- Here check to make certain the source is an instance of the JMenuItem.
- If the menu item with the text of "New Game" is clicked, make sure this is exactly as the text on the item; we display a pop-up message asking if you want to go first. If you go first you're X.
- We create a new JDialog object that creates a dialog of 2P and we invoke the show() method of the JDialog class. Note: From the beginning of JOptionPane through the first semicolon, keep the code on the same line allowing the software to do the wrapping.

13. Let's continue by typing:

```
Object selectedValue = pane.getValue();
   if(selectedValue.equals(new Integer(JOptionPane.YES_OPTION)))
  {
      for(int i = 0; i < 3; i++)
       {
        for(int j = 0; j < 3; j++)
        {
          c[i][j].token = ' ';
          c[i][j].repaint();
          c[i][j].removeMouseListener(c[i][j]);
          c[i][j].addMouseListener(c[i][j]);
        }
       }
```

- We create an Object object that retrieves the value from the pane.
- If the selected value equals a new integer returned by the JOptionPane YES_OPTION, which means that you received the value of 0.
- In for loops we repaint the screen and remove and add MouseListeners.

14. We set the text on the label and change the values of our variables by typing:

```
   jlblStatus.setText("X's turn");
   turn = 'X';
   com = 'O';
   hum = 'X';
}
```

15. The next code is similar to what we already typed, so type:

```
else
{
 for(int i = 0; i < 3; i++)
 {
  for(int j = 0; j < 3; j++)
  {
   c[i][j].token = ' ';
   c[i][j].repaint();
   c[i][j].addMouseListener(c[i][j]);
  }
 }
 jlblStatus.setText("X's turn");
 turn = 'X';
 com = 'X';
 hum = 'O';
```

16. Let's finish this else statement by typing:

```
 int r = (int)(3*Math.random());
 int col = (int)(3*Math.random());
 c[r][col].token = com;
 c[r][col].repaint();
 turn = hum;
 jlblStatus.setText(turn + "'s turn");
 }
}
```

- We create two random integers that represent the row and column.
- We set the token for a row and column to *com*.
- We repaint the row and column.
- We change the value of *turn*.
- Set change the text on the label.

17. We finish the actionPerformed method by typing:

```
    else if (e.getActionCommand() == "Exit")
    {
     System.exit(1);
    }
  }
}
```

- Notice that we use 1 instead of 0 in the System.exit? We are terminating the program, but not naturally.

18. The next method determines whose turn it is and proceeds accordingly. Since we have quite a few if statements I will be including many on separate pages so you can keep track of which you have already typed. Start by typing:

```
public void comTurn()
{
   boolean found = false;

   if(!found)
   {
    for(int row=0;row<3;row++)
    {
     if(found)
       break;
     if((c[row][0].token == com) && (c[row][1].token == com))
     {
      if(c[row][2].token == ' ')
      {
        c[row][2].token = com;
        c[row][2].repaint();
        found = true;
      }//close last if
     }//close 2nd to last if
    }//close for loop
   }//close other if
```

- If the variable *found* IS NOT true:
    - We use a for loop to go through the rows, up to 3 since there are only 3 rows.
    - If found is true:
        - Break from the loop
        - It will check to see if the token in two are rows are the same by seeing if they equal the variable *com*.
        - If the array with an element of the index number of [row][2] (3rd column) token has not value, equals ' ' (yes there is a space)
            a. Set the array to equal the variable *com*, repaint the screen and set *found* to true.

19. The next code is a repeat of the previous except for different cells. Type:

```
if(!found)
 {
  for(int row=0;row<3;row++)
  {
   if(found)
    break;
   if((c[row][1].token == com) && (c[row][2].token == com))
   {
    if(c[row][0].token == ' ')
    {
     c[row][0].token = com;
     c[row][0].repaint();
     found = true;
    }//close last if
   }//close 2nd to last if
  }//close for
 }//close outer if
```

```
 if(!found)
{
  for(int row=0;row<3;row++)
  {
    if(found)
      break;
    if((c[row][0].token == com) && (c[row][2].token == com))
    {
      if(c[row][1].token == ' ')
      {
        c[row][1].token = com;
        c[row][1].repaint();
        found = true;
      }//closes if statement
    }//closes next if statement
  }//closes for loop
}//closes outer if statement
```

```
 if(!found)
{
  for(int col = 0; col < 3;col++)
  {
    if(found)
      break;
    if((c[0][col].token == turn) && (c[1][col].token == turn))
    {
      if(c[2][col].token == ' ')
      {
        c[2][col].token = com;
        c[2][col].repaint();
        found = true;
      }//closes last if statement
    }//closes first inner if statement
  }//closes for loop
}//closes outer if statement
```

```
 if(!found)
{
  for(int col=0;col<3;col++)
  {
    if(found)
      break;
    if((c[1][col].token == turn) && (c[2][col].token == turn))
    {
      if(c[0][col].token == ' ')
      {
        c[0][col].token = com;
        c[0][col].repaint();
        found = true;
      }//closes last if statement
    }//closes inner if statement
  }//closes for loop
}//closes outer if statement
```

```
 if(!found)
{
  for(int col=0;col<3;col++)
  {
    if(found)
      break;
    if((c[0][col].token == turn) && (c[2][col].token == turn))
    {
      if(c[1][col].token == ' ')
      {
        c[1][col].token = com;
        c[1][col].repaint();
        found = true;
      }//closes last if statement
    }//closes second if statement
  }//closes for loop
}//closes outer if statement
```

```
 if(!found)
{
  if((c[0][0].token == com) && (c[1][1].token == com))
  {
    if(c[2][2].token == ' ')
    {
      c[2][2].token = com;
      c[2][2].repaint();
      found = true;
    }//closes inner if statement
  }//closes second if statement
}//closes outer if  statement
```

```
 if(!found)
{
  if((c[2][2].token == com) && (c[1][1].token == com))
  {
    if(c[0][0].token == ' ')
    {
      c[0][0].token = com;
      c[0][0].repaint();
      found = true;
    }//closes last if statement
  }//closes 2nd if statement
}//closes outer if statement
```

```
 if(!found)
{
  if((c[0][0].token == com) && (c[2][2].token == com))
  {
    if(c[1][1].token == ' ')
    {
      c[1][1].token = com;
      c[1][1].repaint();
      found = true;
    }//last if
  }//second if
}//outer if

if(!found)
{
  if((c[0][2].token == com) && (c[2][0].token == com))
  {
    if(c[1][1].token == ' ')
    {
      c[1][1].token = com;
      c[1][1].repaint();
      found = true;
    }//last if
  }//second if
}//outer if
```

```
if(!found)
{
  if((c[2][0].token == com) && (c[1][1].token == com))
  {
    if(c[0][2].token == ' ')
    {
      c[0][2].token = com;
      c[0][2].repaint();
      found = true;
    }//last if
  }//2nd if
}//outer if
```

```
 if(!found)
{
  if((c[0][2].token == com) && (c[1][1].token == com))
  {
    if(c[2][0].token == ' ')
    {
      c[2][0].token = com;
      c[2][0].repaint();
      found = true;
    }//last if
  }//2nd if
}//outer if
```

```
 if(!found)
{
  for(int row=0;row<3;row++)
  {
    if(found)
      break;
    if((c[row][0].token == hum) &&  (c[row][1].token == hum))
    {
      if(c[row][2].token == ' ')
      {
        c[row][2].token = com;
        c[row][2].repaint();
        found = true;
      }//last if
    }//2nd if
  }//for loop
}//outer if
```

```
 if(!found)
{
  for(int row=0;row<3;row++)
  {
    if(found)
      break;
    if((c[row][1].token == hum) && (c[row][2].token == hum))
    {
      if(c[row][0].token == ' ');
      {
        c[row][0].token = com;
        c[row][0].repaint();
        found = true;
      }//last if
    }//2nd if
  }//for loop
}//outer if
```

```
 if(!found)
{
 for(int row=0;row<3;row++)
 {
  if(found)
   break;
  if((c[row][0].token == hum) && (c[row][2].token == hum))
  {
   if(c[row][1].token == ' ') {
    c[row][1].token = com;
    c[row][1].repaint();
    found = true;
   }//last if
  }//2nd if
 }//for loop
}//outer if
```

```
 if(!found)
{
  for(int col=0;col<3;col++)
  {
    if((c[0][col].token == hum) && (c[1][col].token == hum))
    {
      if(c[2][col].token == ' ')
      {
        c[2][col].token = com;
        c[2][col].repaint();
        found = true;
      }//last if
    }//2nd if
  }//for
}//outer if
```

```
 if(!found)
{
  for(int col=0;col<3;col++)
  {
    if((c[1][col].token == hum) && (c[2][col].token == hum))
    {
      if(c[0][col].token == ' ')
      {
        c[0][col].token = com;
        c[0][col].repaint();
        found = true;
      }//last if
    }//2nd if
  }//for
}//outer if
```

```
if(!found)
{
  for(int col=0;col<3;col++)
  {
    if((c[0][col].token == hum) && (c[2][col].token == hum))
    {
      if(c[1][col].token == ' ')
      {
        c[1][col].token = com;
        c[1][col].repaint();
        found = true;
      }//last if
    }//2nd if
  }//for
}//outer if
```

```
if(!found)
{
  if((c[0][0].token == hum) && (c[1][1].token == hum))
  {
    if(c[2][2].token == ' ')
    {
      c[2][2].token = com;
      c[2][2].repaint();
      found = true;
    }//last if
  }//2nd if
}//outer if

if(!found)
{
  if((c[2][2].token == hum) &&  (c[1][1].token == hum))
  {
    if(c[0][0].token == ' ')
    {
      c[0][0].token = com;
      c[0][0].repaint();
      found = true;
    }//last if
  }//2nd if
}//outer if
```

```
if(!found)
{
  if((c[0][0].token == hum) && (c[2][2].token == hum))
  {
    if(c[1][1].token == ' ')
    {
      c[1][1].token = com;
      c[1][1].repaint();
      found = true;
    }//last if
  }//2nd if
}//outer if
```

```
 if(!found)
{
  if((c[0][2].token == hum) && (c[2][0].token == hum))
  {
    if(c[1][1].token == ' ')
    {
      c[1][1].token = com;
      c[1][1].repaint();
      found = true;
    }//last if
  }//2nd if
}//outer if

if(!found)
{
  if((c[2][0].token == hum) && (c[1][1].token == hum))
  {
    if(c[0][2].token == ' ')
    {
      c[0][2].token = com;
      c[0][2].repaint();
      found = true;
    }//last if
  }//2nd if
}//outer if
```

```
if(!found)
{
  if((c[0][2].token == hum) && (c[1][1].token == hum))
  {
    if(c[2][0].token == ' ')
    {
      c[2][0].token = com;
      c[2][0].repaint();
      found = true;
    }//close last if
  }//close 2nd if
}//close outer if

if(!found)
{
  if(c[1][1].token == ' ')
  {
    c[1][1].token = com;
    c[1][1].repaint();
    found = true;
  }//close inner if
}//close outer if
```

```
 if(!found)
{
  for(int i = 0; i<3;i++)
  {
    if(found)
      break;
    for(int j = 0; j<3; j++)
    {
      if(found)
        break;
      if(c[i][j].token == ' ')
      {
        c[i][j].token = com;
        c[i][j].repaint();
        found = true;
      }//close last if
    }//close inner for
  }//close outer for
}//close outer if
```

20. Wow that was a lot of code, but think about it. You have to check and compare each cell with the other cells and also check to see if you have 3 in a row for a win.

21. Let's finish this method by typing:

```
if(winGame())
{
    jlblStatus.setText(turn + " wins the game.");
    removeListeners();
  }
  else if(drawGame())
  {
    jlblStatus.setText("Draw game, reset to play again.");
  }
  else
  {
    if(com == 'X')
    {
      turn = 'O';
    }
    else
    {
      turn = 'X';
    }
    jlblStatus.setText(turn + "'s turn");
  }//close else if
}//close outer if
```

- This is pretty clear as to what's going on. It starts with if the winGame() method returns true. Let me know if you cannot understand what is going on.

22. Now for the winGame() method. Type:

```
public boolean winGame()
{
   for(int row=0;row<3;row++)
   {
    if((c[row][0].token == turn) && (c[row][1].token == turn) &&  (c[row][2].token ==
      turn))
        return true;
   }
   for(int col=0;col<3;col++)
   {
    if((c[0][col].token == turn) && (c[1][col].token == turn) && (c[2][col].token ==
     turn))
        return true;
   }

   if((c[0][0].token == turn) && (c[1][1].token == turn) && (c[2][2].token == turn))
       return true;

   if((c[0][2].token == turn) && (c[1][1].token == turn) && (c[2][0].token == turn))
       return true;

   return false;
 }
```

- Here we check different rows and cells to see if it equals the value of the variable *turn*, the token. In each case we return true if the condition is true.
- Otherwise we return false.

23. Our removeListeners() method, does exactly what it says. Type:

```
public void removeListeners()
{
   for(int i=0;i<3;i++)
   {
     for(int j=0;j<3;j++)
     {
       c[i][j].removeMouseListener(c[i][j]);
     }//close inner for
   }//close outer for
  }
```

24. Our drawGame() method checks to see if it's a draw. Type:

```
public boolean drawGame()
{
   for(int i=0;i<3;i++)
   {
     for(int j=0;j<3;j++)
     {
       if(c[i][j].token == ' ')
         return false;
     }//close inner for
   }//close outer for

   return true;
  }
```

25. Finally we complete the Cell class which extends JPanel and is responsible for drawing, painting on the screen and having the program listen for mouse clicks. Type:

```
class Cell extends JPanel implements MouseListener
 {
  char token = ' ';


  public Cell()
  {
    this.setBorder(new LineBorder(Color.black, 1));
  }
```

```
 public void paintComponent(Graphics g)
{
  super.paintComponent(g);
  if (token == 'X')
  {
    g.setColor(Color.black);
    g.drawLine(5,5, getWidth()-5, getHeight()-5);
    g.drawLine(5, getHeight()-5, getWidth()-5, 5);
  }
  else if (token == 'O')
  {
    g.setColor(Color.black);
    g.drawOval(5,5, getWidth()-10, getHeight()-10);
  }
}
```

```
 public void mouseClicked(MouseEvent e)
{
 if(token == ' ')
 {
  this.token = turn;
  this.repaint();
  if(winGame())
  {
   jlblStatus.setText(turn + " wins the game.");
   removeListeners();
  }
  else if(drawGame())
  {
   jlblStatus.setText("Draw game, reset to play again.");
  }
  else
 {
   if(turn == 'X')
   {
    turn = 'O';
    jlblStatus.setText(turn + "'s turn");
   }
   else if (turn == 'O')
   {
    turn = 'X';
```

```
        jlblStatus.setText(turn + "'s turn");
      }

      if(turn == com)
      {
        comTurn();
      }//close last if
    }//close else
  }//close first if
}

//the following methods are required even if not used when using the MouseEvent
 public void mouseEntered(MouseEvent e)
{

}

public void mouseExited(MouseEvent e)
{
}

public void mousePressed(MouseEvent e)
{
}
```

```
    public void mouseReleased(MouseEvent e)
    {
    }
  }
}
```

26. I want you to go through this code, step-by-step, to make sure you understand what is occurring. By this time you should be able to read code and determine this.
27. Compile the program and fix any errors.
28. Run the program to test it out.
29. Compress all files into a single zip or rar file and submit.