

YOUR NAME: _____ DATE: _____

LAST FOUR DIGITS OF YOUR UF-ID: ____ ____ ____ ____ Please Print Clearly (Block Letters)

YOUR PARTNER'S NAME: _____ DATE: _____

LAST FOUR DIGITS OF PARTNER'S UF-ID: ____ ____ ____ ____ Please Print Clearly

Date Assigned: 19 April 2013 IN CLASS

Date Due: 26 April 2013 E-SUBMISSION of Parts I through III

In this homework assignment you may work in groups of two persons only. You may not copy from others, and you may not copy code from the Internet, textbook, or other sources.

However, you may study with others or read your textbook to determine general solutions. Then you must complete the problems as your own work, not copying others' work.

Questions about this homework should be addressed to your TA first. You can find your TA's email, office hours, etc. at the class website: <http://www.cise.ufl.edu/~wchapman/COP2800/officehours.html>

This homework has three parts: (I) Vocabulary Questions, (II) Regular Program, (III) Advanced Program. There is no penalty for guessing.

Part I. Vocabulary Questions

[10 points total]

Vocabulary: (terms you need to know to discuss the subject intelligently) – Define the following terms using 1-3 sentences: **[2 points each]**

- a. *Java Applet*
- b. *Java JApplet*
- c. *Javascript*
- d. *Servlet* (in Java)
- e. *Java Swing*

Use your text editor (Notepad++) to generate a file called "PartI.txt". Include this file in the ZIP file along with the code for Parts II and answers for Part III.

You **must** have in the upper right-hand corner: (i) "COP2800-S13-HW6-PartI", (ii) your name, and (iii) last four digits of your UFID.

Part II. Java Program

[60 points total]

TASK: Create a JApplet for your Java Program that plays TicTacToe based on the GUI created in Assignment 5. Given the results of Assignment 5 (intelligent TicTacToe game with GUI), make a Java JApplet that makes the TicTacToe game run as a Java Applet on the Web. As in Assignments 3 through 5, we allow you to work in groups of two, so that should make it easier for your coding/testing effort.

PROGRAMMING PROCEDURE:

- (1) We will be using your Assignment 5 code as a starting point for this assignment. If your code did not work properly, you may download a correct solution to Assignment 5 from this link:

<http://www.cise.ufl.edu/~wchapman/COP2800/misc/Assignment5-solution.zip>

- (2) Make a public Java **Class** called **ITTTApplet**, which extends **JApplet** and implements method **public void init()** – which is the entry point to the Java applet. Recall that the hello world for JApplets is:

```
import javax.swing.*;
public class HelloWorld extends JApplet {
    public void init() {
        JLabel label = new JLabel("Hello World");
        add(label);
    }
}
```

The HTML code required to display the HelloWorld applet on a web page (helloworld.html) is:

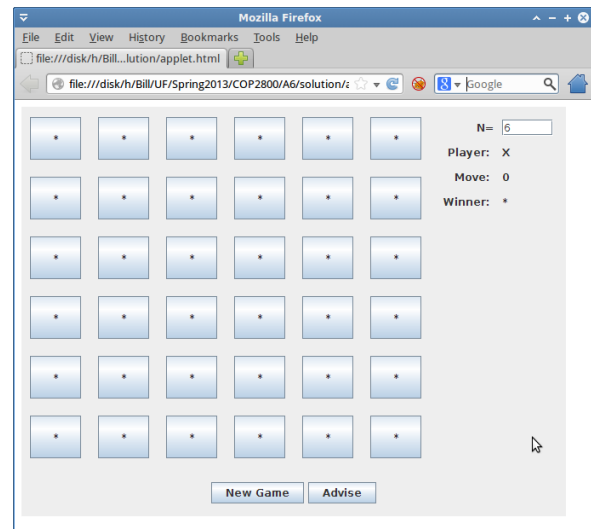
```
<html><head><title>My Applet</title></head><body>
<applet code="HelloWorld.class" width="640" height="480"></applet>
</body></html>
```

The command to display this web page using java's applet viewer is

```
appletviewer helloworld.html
```

- (3) Add elements to the JApplet to create the same user interface you implemented for Assignment 5. You do not need to implement an Exit button. The final product should look like the image displayed to the right:

Important Hint: Recall from Assignment 5 that we created our GUI by adding components (JPanels, JButtons, JLabels, and JTextFields) to the JFrame's content pane, using layout managers to arrange them on the screen. You can add all the same components to your JApplet using the same layout managers. In fact, nearly the entire solution to this assignment can be copied and pasted from your ITTTGUI.java file. The only major changes are: your main method is replaced by nonstatic method init(); pack() can no longer be used to resize your interface (in fact, your UI size is constant); and the new top level container is now JApplet itself rather than JFrame.getContentPane().



- (4) Create a file, applet.html, that loads your ITTTApplet.class file at size 640 x 480. Use the example HTML file shown above as a reference, and just modify it to load the correct class file.

- (5) Use the command “appletviewer applet.html” to test and debug your applet. **It is highly recommended** to use this command rather than debugging your applet in a web browser. Browsers will often cache the old class file, and may not update it after you recompile.
- (6) You should verify that your program works correctly by working through all the combinations of buttons and game board sizing.

Part III. Extra Credit (text file like Part I, submit online)

[15 points]

EC-1. Please describe how you would implement a Tic Tac Toe web interface that extends *Applet* and uses only the paint method (no Components). Please describe, in text (not code), how an Applet could (i) display a game board; (ii) display current game state information, including player, move, and winner; (iii) respond to user clicks over the game board, and determine what game board position was clicked (iv) allow the user to enter a new game board size by typing it with their keyboard.

Please do this part electronically.

Use your text editor (Notepad++) to generate a file called "Part3.txt". Include this file in the ZIP file along with the code for Part II.

You **must** have in the upper right-hand corner: (i) “COP2800-S13-HW6-PartIII”, (ii) your name, and (iii) last four digits of your UFID.

Hint: the motivation for this question is to get you to think about just how much harder GUI development would be without standard swing elements like JButton, JTextField, etc. A full credit solution will explain how the methods of java.awt.Graphics, java.awt.event.MouseListener, and java.awt.KeyListener can be used to manually (and tediously) re-implement the functionality provided by these elements. You do not need to write any code for this question. The code to do this would be very complicated.

Electronic Submission. Put all files you created in Parts I through III in a single ZIP file. Your ZIP file should contain all the files specified in Parts I through III. Submit this ZIP file electronically per the instructions at:

<http://www.cise.ufl.edu/~wchapman/COP2800/submit>

Part IV. Evaluation of Submitted Code

Grading: Code does not compile or run	= 0 points.
Code compiles and displays all or part of GUI but does not run the TicTacToe program	= < 30 percent of points.
Code, compiles, displays GUI, but runs TTT program with wrong results	= 31 to 60 percent of points.
Code runs with correct results but no documentation (e.g., comments in Part II)	= 61 to 85 percent of points.
Code compiles and runs, correct results, documentation present	= 86 to 100 percent of points.

