# SER 210 – Software Engineering Design and Development

## Assignment 1

## Objectives

- Refresh your java programming skills
- Practice developing an Android UI.
- Use intents.
- Manage application configuration.

## Overview

**This is an individual assignment.**

In this assignment, you will create a Four-in-a-row Android game in which the user plays against the computer. You will start by implementing a java console-version to refresh your java coding skills.

## Game Rules

Recall the game rules:

- The boards size 6 by 6.

- A winner must have 4 consecutive same color discs in a row

## Part 1:

Use the attached code to implement a **console version of the game**. This part of the assignment is a java refresher. Download the attached project and open in **Eclipse**.

The attached code includes the `IGame` interface. **You are NOT allowed to modify the interface.**

Instructions:

- Finish the implementation of the class `FourInARow`.
- Use the two-dimensional `board` array to represent the game board.
- Focus on the AI or computer move in this version.
- The `FourInARow` class provides an implementation of two methods for displaying the board.
- Use `FIRConsole` class to run the game.

**Show a demo in class on Thurs 2/4. Submit your work by Thursday midnight**

**This part is worth 30 points.**

## Part 2:

Implement an Android version of Four in a row..

## Game Requirements

The game should start with a splash screen including a text explaining the rules of the game and scoring. This splash screen will ask the player to input their name. Player will transition to the play board screen using a Start button or on enter.

The play board should display the Four-in-a-row board, the username and text indicating whose turn is it. It should also display a win/lose/tie message if the game ends as well as a reset button to start a new game. This button should be disabled until the game ends.

## Implementation Requirements

Use the interface IGame and FourInARow classes implementation in Part 1 as the backend of your code. Declare an instance of type IGame in your FourInARow MainActivity class to represent the game.

Use a GridLayout or TableLayout to represent your board.

Your application should maintain its status if the user flips the phone to horizontal orientation.

A through documentation of your code include the author name on each file is expected.

**+ 5 points for extra features and creativity.**

## Due dates:

**Part 1** is due Thru 2/4th. An in-class demo is required.

**Part 2** is due Friday 2/23rd

## Submission guideline for the Android game

1- Record a video demo of your working app. You demo must show different scenarios in running the game. At least winning and losing. Submit a link to your recording.
2- Submit a Github link of your code repository. Code must include both the console version and Android version

## Grading Criteria for Part 2.

| Criteria | Points |
|---|---|
| **Game play :** | |
| Game play is correct | /10 |
| IGame interface implemented correctly | /5 |
| **Basic Requirements** | |
| Game has two different activities | /5 |
| Game uses a GridLayout or TableLayout | /5 |
| Game does not lose state when the phone is flipped | /5 |
| An Intent is used to communicate data between activities | /5 |
| **UI Requirements:** | |
| Startup screen accepts username, displays instructions and has a start button | /5 |
| UI component placement is correct | /5 |
| Text control displays username | /5 |
| Icons or images are used for X and O | /5 |
| Text control is updated correctly during play and at the end of the game | /5 |
| Reset Button | /5 |
| **Code Quality** | |
| Clean code including the use of constants, no repetitions, etc. (see java guidelines on BB) | /5 |
| Code is well commented including your name in each header, assignment name and date | /4 |
| Following submission guidelines | /1 |
| **Total** | **/90** |
| **Bonus up to 5 points** | |