Summary
In this assignment you will write a program to play the game Connect Four.

Deadline
This assignment is due on Friday, April 4 at 11:00 pm.

Description
Connect Four is a popular game played by dropping checkers into a vertical grid with six rows and seven columns. The checkers can be dropped in any of the seven columns (permitted they are not full). Once dropped, the checker falls to the lowest unoccupied space. The winner is the first player to occupy four consecutive spaces in a straight line, either horizontal, vertical, or diagonal.

Specifications
In this project you will write a program to represent the game Connect Four using a two-dimensional array of integers to represent the grid and a Picture object to display the game. You will create a class called `ConnectFour` to represent the game. The class `PlayConnectFour` (provided) will allow the user to play the game. When run, this program will display the game board and ask the user to specify a column. The user is always player 1 (red checkers) and goes first. After the user enters a column the opponent will select a column for their piece.

For this project you will create a simple naive opponent that selects a column at random. Then the program will update and repaint the board and ask the user to select another column. This will repeat until the game ends in a win, loss, or draw.

To help with this project a `Referee` class is provided. The referee will look at your grid and tell you whether the game is a win, loss, draw, or is not finished.
Principles of OO Programming

- **Encapsulation**
  - Objects can combine data and operations

- **Inheritance**
  - Classes can inherit properties from other classes

- **Polymorphism**
  - Objects can determine appropriate operations at execution time

Inheritance

- **Inheritance**
  - Allows a class to derive the behavior and structure of an existing class

- **Superclass or base class**
  - A class from which another class is derived

- **Subclass, derived class, or descendant class**
  - A class that inherits the members of another class

- **Benefits of inheritance**
  - It enables the reuse of existing classes
  - It reduces the effort necessary to add features to an existing object
Inheritance

- A subclass
  - Can add new members to those it inherits
  - Can override an inherited method of its superclass
    - A method in a subclass overrides a method in the superclass if the two methods have the same declarations

The subclass `Ball` inherits members of the superclass `Sphere` and overrides and adds methods.

Inheritance

- A subclass inherits private members from the superclass, but cannot access them directly
- Methods of a subclass can call the superclass’s public methods
- Clients of a subclass can invoke the superclass’s public methods
- An overridden method
  - Instances of the subclass will use the new method
  - Instances of the superclass will use the original method

An object invokes the correct version of a method.
Java Access Modifiers

- Membership categories of a class
  - Public members can be used by anyone
  - Members declared without an access modifier (the default) are available to
    - Methods of the class
    - Methods of other classes in the same package
  - Private members can be used only by methods of the class
  - Protected members can be used only by
    - Methods of the class
    - Methods of other classes in the same package
    - Methods of the subclass

Relationships

- Two basic kinds of relationships
  - Is-a relationship
  - Has-a relationship

Is-a Relationship

- Inheritance should imply an is-a relationship between the superclass and the subclass
- Example:
  - If the class Ball is derived from the class Sphere
    - A ball is a sphere
Is-a Relationship

- Object type compatibility
  - An instance of a subclass can be used instead of an instance of the superclass, but not the other way around

Has-a Relationships

- Has-a relationship
  - Also called containment
  - Cannot be implemented using inheritance
    - Example: To implement the has-a relationship between a pen and a ball
      - Define a data field `point` - whose type is `Ball` - within the class `Pen`