Weekly Lab 6 – **Overriding the *equals* Method**

Maximum Points = 10

File *Player.java* contains a class that holds information about an athlete: name, team, and uniform number. File *ComparePlayers.java* contains a skeletal program that uses the Player class to read in information about two baseball players and determine whether or not they are the same player.

1. Fill in the missing code in ComparePlayers so that it reads in two players and prints "Same player" if they are the same, "Different players" if they are different. Use the *equals* method, which Player inherits from the Object class, to determine whether two players are the same. Are the results what you expect?

2. The problem above is that as defined in the Object class, *equals* does an address comparison. It says that two objects are the same if they live at the same memory location, that is, if the variables that hold references to them are aliases. The two Player objects in this program are not aliases, so even if they contain exactly the same information they will be "not equal." To make *equals* compare the actual information in the object, you can override it with a definition specific to the class. It might make sense to say that two players are "equal" (the same player) if they are on the same team and have the same uniform number.
   - Use this strategy to define an *equals* method for the Player class. Your method should take a Player object and return true if it is equal to the current object, false otherwise.
   - Test your ComparePlayers program using your modified Player class. It should give the results you would expect.

```
// ********************************************************
// Player.java
//
// Defines a Player class that holds information about an athlete.
// ********************************************************

import java.util.Scanner;

public class Player
{
    private String name;
    private String team;
    private int jerseyNumber;


    //-----------------------------------------------------------
    // Prompts for and reads in the player's name, team, and
    // jersey number.
    //-----------------------------------------------------------

    public void readPlayer()
    {
      Scanner scan = new Scanner(System.in);
      System.out.print("Name: ");
      name = scan.nextLine();
      System.out.print("Team: ");
      team = scan.nextLine();
      System.out.print("Jersey number: ");
      jerseyNumber = Scan.nextInt();
    }

}
```

```
// ************************************************************
// ComparePlayers
//
// Reads in two Player objects and tells whether they represent
// the same player.
// ************************************************************
import java.util.Scanner;

public class ComparePlayers
{
    public static void main(String[] args)
    {
        Player player1 = new Player();
        Player player2 = new Player();
        String again;
        String garbage;
        Scanner scan = new Scanner(System.in);

        again = "n";
        do
        {
        //Prompt for and read in information for player 1

        //Prompt for and read in information for player 2

        //Compare player1 to player 2 and print a message saying
        //whether they are equal


            System.out.print("Do you want to continue (y/n)?");
            again = scan.next();
            garbage = scan.nextLine();
        }
        while (again.equals("y") || again.equals("Y"));
    }
}
```

(Due before end of the day on Friday, February 18, 2011) Submit your .java files containing your program to the dropbox in WebCT.

Grades are determined using the following scale:

- Runs correctly.....:___/3
- Correct output.....:___/2
- Design of output.....:___/1
- Design of logic.....:___/2
- Standards.....:___/1
- Documentation.....:___/1

Grading Rubric (Word document)