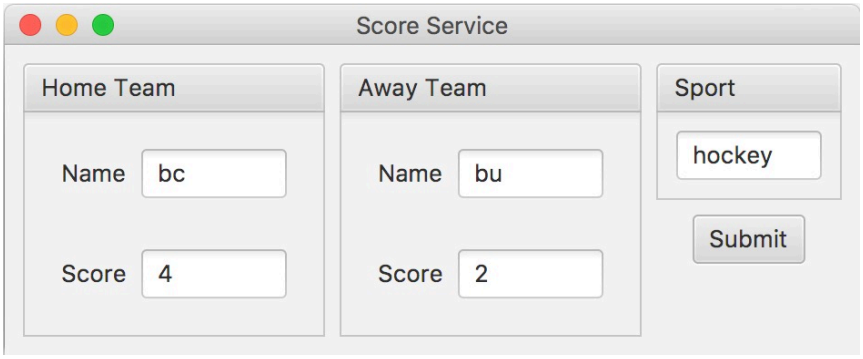
CSCI 3353 Object Oriented Design
Homework Assignment 4
Due Monday, April 30

This assignment will give you practice using JavaFX and the observer pattern. The idea is to emulate a service that supplies the results of sporting events to interested clients, similar to an RSS feed. Each time the service receives the score of an event, it notifies each of its clients. A client can then do what it likes with this information.

To make things more specific, your program will have a service class *ScoreService* and two client classes *SportClient* and *TeamClient*. These classes should be organized using the observer pattern. There will be one *ScoreService* object, which is the observable. There can be arbitrarily many client objects, who are the observers. Each client object registers itself as an observer of the score service.

The following screenshot illustrates what the program needs to look like.



Each score has five component values: the sport, the names of the home team and away team, and the scores of the home team and away team. The user reports a new score by filling in the five text boxes and clicking the Submit button. The service then notifies its observers.

The observer interface is called *Client*. Here is its code.

```
public interface Client {
   void newScore(String sport, String hometeam,
                 String awayteam, String homescore,
                 String awayscore);
}
```

Note that the scores are transmitted as strings. A client is responsible for converting the strings to numbers, if desired.

A *SportClient* object is interested only in scores for a specific sport. For example, the statement

```
Client c1 = new SportClient("hockey");
```

creates a new client that is interested only in scores of hockey games. A *TeamClient* object is interested only in scores for a specific team. For example, the statement

```
Client c2 = new TeamClient("bc");
```

creates a new client that is interested only in scores of games involving BC teams. For simplicity, we shall assume that if a client receives the score of an event that it finds interesting, the client will print that score on the console. For example, if c1 and c2 are the observers when the above Submit button is clicked, each will print a message on the console. In my program, the messages look like this:

```
New hockey score: bc 4, bu 2
New score in bc hockey: bc 4, bu 2
```

The main class is *HW10Program*. Its code looks like this:

```
public class HW10Program extends Application {
   public void start(Stage stage) {
      ScoreService service = new ScoreService();
      Client c1 = new SportClient("chess", service);
      Client c2 = new SportClient("golf", service);
      Client c3 = new TeamClient("bc", service);

      stage.setScene(new Scene(service.getRoot()));
      stage.setTitle("Score Service");
      stage.show();
   }

   public static void main(String[] args) {
      Application.launch(args);
   }
}
```

From this, you can see that the *ScoreService* class is responsible for constructing the node hierarchy for the window. The program calls its public method *getRoot* to construct the JavaFX *Scene* object.

What you can't see is how *ScoreService* maintains its observer list and how it notifies its observers. Those issues are for you to decide. The only requirement is that your code must work with the above *HW10Program* code, and that you use *Client* as the observer interface.

Note: Your JavaFX window doesn't have to look like mine. My only requirement is that it is easy to read and to use. I found it helpful to look at the JavaDoc documentation for JavaFX, so I could investigate what kind of controls and panes are available, and how to use them. The URL for the online documentation is:

https://docs.oracle.com/javase/8/javafx/api/toc.htm

WHAT TO SUBMIT: Please submit the following five files: *Client.java*, *SportClient.java*, *TeamClient.java*, *ScoreService.java*, and *HW10Program.java*. The Java files must be in the package "hw10". Create a zip file containing these files and submit it to Canvas.