cricketobserver

```
public class cricketobserver {
    void updatecricketobserver(String sport, String message);
}
```

Footballobserver

```
public class footballobserver {
    void updatefootballobserver(String sport, String message);
}
```

Hockeyobserver

```
public class hockeyobserver {
    void updatehockeyobserver(String sport, String message);
}
```

Sportsobserver

```
import java.util.ArrayList;
import java.util.List;
import java.util.concurrent.Flow.Subscriber;

public interface Sportsobserver {
    void subscribe(SportsSubsricption  subscriber);
    void unsubscribe(SportsSubsricption  subscriber);
    void notifysubscriber(SportsSubsricption  subscriber);
}
```

Sportssubscription

```
//sportssubscription.java

import java.util.concurrent.Flow.Subscriber;

class SportsSubsricption implements Sportsobserver {
private List<Subscriber> subscribers = new ArrayList<>();

 public void sendSportsUpdate(String sport, String message){
  notifySubscriber(sport,message);
}
@Override
public void subscribe(Subscriber subscriber){
  subscribers.add(subscriber);
}

@Override
public void unsubscribe(Subscriber subscriber){
  subscribers.remove(subscriber);
}
@Override
public void notifySubscribers(String sport, String message){
  for(Subscriber subscribe : subscribers){
   subscriber.update(sport,message);

  }
}

}
```

Sportsusbcriptiondemo

```
public class SportsSubsricptiondemo {
    public static void main(String[] args) {
        SportsSubsricption subscricption = new  SportsSubsricption();
        User user1 =new User ("");
        User user2 =new User ("");

        subscricption.subscribe(user1);
        subscricption.subscribe(user2);

        subscricption.sendSportsUpdate("cricket", "match rescheduled");
        subscricption.sendSportsUpdate("football", "match rescheduled");
        subscricption.sendSportsUpdate("cricket", "match lost by pakistan");


    }
}
```

Subscriber

```
public interface subscriber {
void update(String sport, String message);

}
```

User

```
public class User implements subscriber {
    private String name;
    public User(String name){
     this.name = name;

    }

  @Override
  public void update(String sport, String message){
   System.out.println("name" + sport + message);
  }

}
```

Volleyballobserver

```
public class volleyballobserver {
    void updatevolleyballobserver(String sport, String message);
}
```