

Code Observer Pattern

```
public class ConcreteObserver1 extends Observer{  
    public ConcreteObserver1(CricketMatch1 subject , CricketMatch2 subject2){  
        this.subject = subject;  
        this.subject.attach(this);  
        this.subject2 = subject2;  
        this.subject2.attach(this);  
    }  
    @Override  
    public void update() {  
        System.out.println( "The Name of team is " + subject.getTeam() );  
        System.out.println( "The score of team is " + subject.getScore() );  
        System.out.println( "The Name of team2 is " + subject2.getTeam1() );  
        System.out.println( "The score of team2 is " + subject2.getScore1() );  
    }  
}  
  
public class ConcreteObserver2 extends Observer{  
    public ConcreteObserver2(CricketMatch1 subject , CricketMatch2 subject2){  
        this.subject = subject;  
        this.subject.attach(this);  
        this.subject2 = subject2;  
        this.subject2.attach(this);  
    }  
    @Override  
    public void update() {  
        System.out.println( "The Name of team is " + subject.getTeam() );  
        System.out.println( "The score of team is " + subject.getScore() );  
        System.out.println( "The Name of team2 is " + subject2.getTeam1() );
```

```
System.out.println( "The score of team2 is " + subject2.getScore1() );  
    }  
}
```

```
public class Cricket {
```

```
    public static void main(String[] args) {  
        CricketMatch1 subject = new CricketMatch1();  
        CricketMatch2 subject2 = new CricketMatch2();  
        ConcreteObserver1 ob =new ConcreteObserver1(subject,subject2);  
        ConcreteObserver2 ob1 =new ConcreteObserver2(subject,subject2);  
        //subject.detach(ob);  
        // System.out.println("Cricket Match 1 ");  
        // System.out.println("Score of Match 1 is:");  
        subject.setScore(150);  
        subject.setTeam("Pakistan");  
        subject2.setScore1(200);  
        subject2.setTeam1("India");  
    }  
}
```

```
public class CricketMatch1 {
```

```
    private List<Observer> observers = new ArrayList<Observer>();  
    private int score;  
    private String team;  
    public int getScore() {  
        return score;  
    }  
    public void setScore(int score) {  
        this.score = score;  
        notifyAllObservers();  
    }  
    public String getTeam() {
```

```

        return team;
    }
    public void setTeam(String team) {
        this.team = team;
        notifyAllObservers();
    }
    public void attach(Observer observer){
        observers.add(observer);
    }
    public void detach(Observer observer){
        observers.remove(observer);
    }
    public void notifyAllObservers(){
        for (Observer observer : observers) {
            observer.update();
        }
    }
}

public class CricketMatch2 {
    private List<Observer> observers = new ArrayList<Observer>();
    private int score1;
    private String team1;
    public int getScore1() {
        return score1;
    }
    public void setScore1(int score) {
        this.score1 = score;
        notifyAllObservers();
    }
    public int getTeam1() {

```

```
        return score1;
    }
    public void setTeam1(String team1) {
        this.team1 = team1;
        notifyAllObservers();
    }
    public void attach(Observer observer){
        observers.add(observer);
    }
    public void detach(Observer observer){
        observers.remove(observer);
    }
    public void notifyAllObservers(){
        for (Observer observer : observers) {
            observer.update();
        }
    }
}

public abstract class Observer {
    protected CricketMatch1 subject;
    protected CricketMatch2 subject2;
    public abstract void update();
}
```