**Code Lab Midterm**

**public interface CricketMatch** {

void attach(CoverageObserver observer);

void detach(CoverageObserver observer);

void notifyAllObservers();

}

**public class Cricket\_Match1 implements CricketMatch** {

private List<CoverageObserver> observers = new ArrayList<CoverageObserver>();

private int score;

private String team1;

private String team2;

public int getScore(){

return score;

}

public void setScore(int score){

this.score=score;

}

public String getTeam1(){

return team1;

}

public String getTeam2(){

return team2;

}

public void setTeams(String team1, String team2){

this.team1=team1;

this.team2=team2;

}

@Override

public void attach(CoverageObserver observer) {

observers.add(observer);

}

@Override

public void detach(CoverageObserver observer) {

observers.remove(observer);

}

@Override

public void notifyAllObservers(){

for (CoverageObserver observer : observers) {

observer.update();

}

}

}

**public class Cricket\_Match2 implements CricketMatch**{

private List<CoverageObserver> observers = new ArrayList<CoverageObserver>();

private int score;

private String team1;

private String team2;

public int getScore(){

return score;

}

public void setScore(int score){

this.score=score;

}

public String getTeam1(){

return team1;

}

public String getTeam2(){

return team2;

}

public void setTeams(String team1, String team2){

this.team1=team1;

this.team2=team2;

}

@Override

public void attach(CoverageObserver observer) {

observers.add(observer);

}

@Override

public void detach(CoverageObserver observer) {

observers.remove(observer);

}

@Override

public void notifyAllObservers(){

for (CoverageObserver observer : observers) {

observer.update();

}

}

}

**public abstract class CoverageObserver** {

protected Cricket\_Match1 match1;

protected Cricket\_Match2 match2;

public abstract void update();

}

**public class BallByBallCverage extends CoverageObserver** {

BallByBallCverage(Cricket\_Match1 match1 , Cricket\_Match2 match2){

this.match1 = match1;

this.match1.attach(this);

this.match2 = match2;

this.match2.attach(this);

}

public void SelectMatch(Cricket\_Match1 match1 , Cricket\_Match2 match2){

System.out.println("Select Matches..");

}

@Override

public void update() {

System.out.println( "Team :" + match1.getTeam1() );

System.out.println( "Score :" + match1.getScore() );

System.out.println( "Team :" + match1.getTeam2() );

}

public void displayBallbyBallCoverage(){

System.out.println( "Team :" + match1.getTeam1() );

System.out.println( "Score :" + match1.getScore() );

}

}

**public class LiveCricketCoverage extends CoverageObserver**{

LiveCricketCoverage(Cricket\_Match1 match1 , Cricket\_Match2 match2){

this.match1 = match1;

this.match1.attach(this);

this.match2 = match2;

this.match2.attach(this);

}

public void SelectMatch(Cricket\_Match1 match1 , Cricket\_Match2 match2){

System.out.println("Select Matches..");

}

@Override

public void update() {

System.out.println( "Team :" + match2.getTeam1() );

System.out.println( "Team :" + match2.getTeam2() );

System.out.println( "Score :" + match2.getScore() );

}

public void displayBallbyBallCoverage(){

System.out.println( "Team :" + match1.getTeam1() );

System.out.println( "Score :" + match1.getScore() );

}

}

**public class MainObserver** {

public static void main(String[] args) {

Cricket\_Match1 match1 = new Cricket\_Match1();

Cricket\_Match2 match2 = new Cricket\_Match2();

BallByBallCverage ballByBallCverage = new BallByBallCverage(match1, match2);

LiveCricketCoverage liveCricketCoverage = new LiveCricketCoverage(match1, match2);

//match1.detach(ballByBallCverage);

match1.setScore(350);

match1.setTeams("Pakistan", "England");

match2.setScore(200);

match2.setTeams("Australia", "India");

}

}