

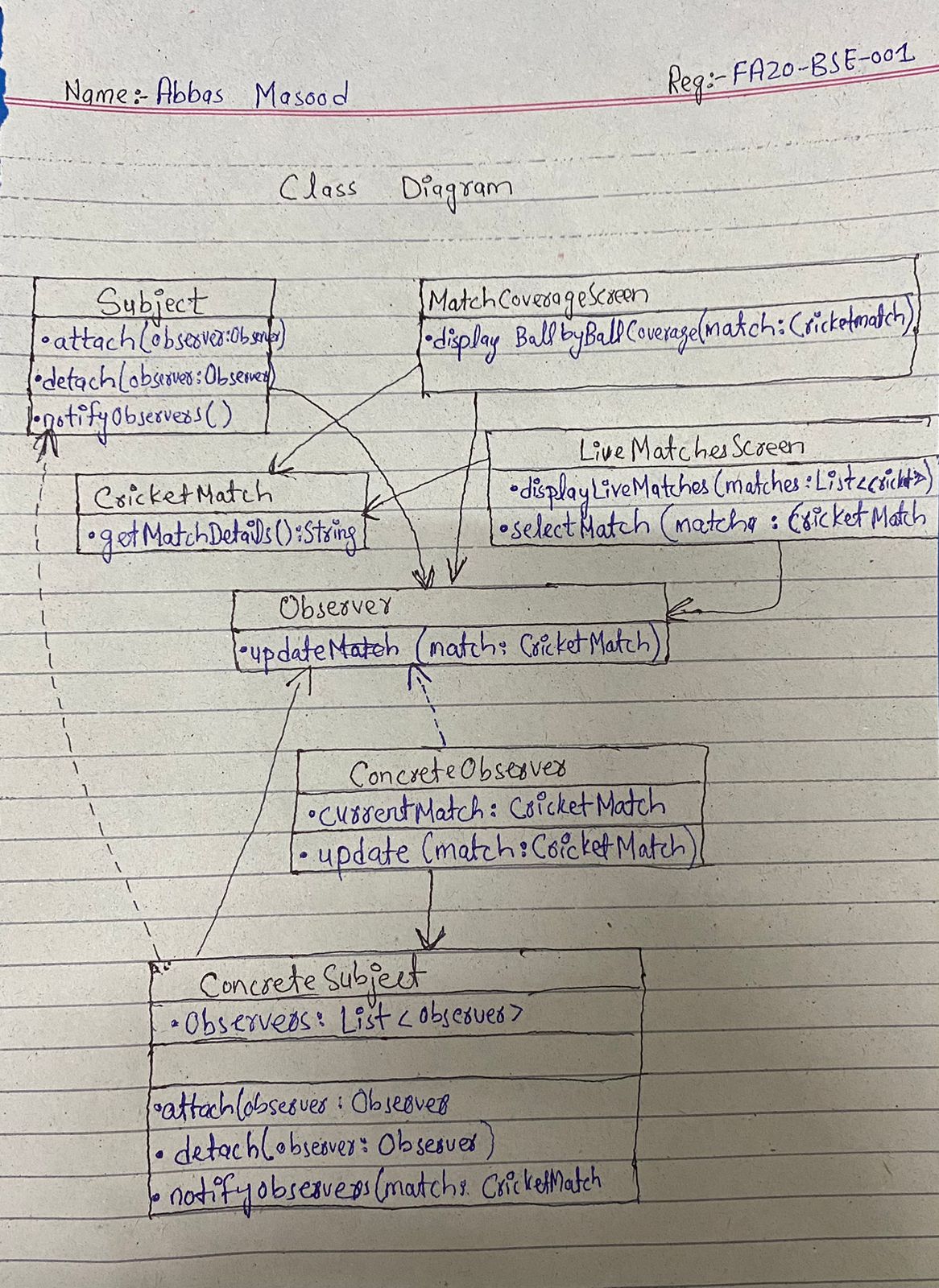
**NAME: - ABBAS MASOOD**

**REG: - FA20-BSE-001**

**LAB: - MID LAB**

**SUBJECT: - Design Patterns**

**Class Diagram: -**



**CODE: -**

import java.util.ArrayList;

import java.util.List;

interface Subject {

void attach(Observer observer);

void detach(Observer observer);

void notifyObservers();

}

class CricketMatch {

private String matchDetails;

CricketMatch(String matchDetails) {

this.matchDetails = matchDetails;

}

String getMatchDetails() {

return matchDetails;

}

}

interface Observer {

void update(CricketMatch match);

}

class ConcreteSubject implements Subject {

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

private CricketMatch currentMatch;

@Override

public void attach(Observer observer) {

observers.add(observer);

}

@Override

public void detach(Observer observer) {

observers.remove(observer);

}

@Override

public void notifyObservers() {

for (Observer observer : observers) {

observer.update(currentMatch);

}

}

public void setMatch(CricketMatch match) {

this.currentMatch = match;

notifyObservers();

}

}

class ConcreteObserver implements Observer {

private CricketMatch currentMatch;

@Override

public void update(CricketMatch match) {

this.currentMatch = match;

// Update the observer's display or perform any other necessary actions

System.out.println("Observer updated: New match details received - " + match.getMatchDetails());

}

}

class LiveMatchesScreen implements Observer {

@Override

public void update(CricketMatch match) {

System.out.println("LiveMatchesScreen: Exciting update! A new match is here - " + match.getMatchDetails());

}

}

class MatchCoverageScreen implements Observer {

@Override

public void update(CricketMatch match) {

System.out.println("MatchCoverageScreen: Dive into the action! Match details have been updated - " + match.getMatchDetails());

}

}

public class Main {

public static void main(String[] args) {

ConcreteSubject subject = new ConcreteSubject();

LiveMatchesScreen liveMatchesScreen = new LiveMatchesScreen();

MatchCoverageScreen matchCoverageScreen = new MatchCoverageScreen();

subject.attach(liveMatchesScreen);

subject.attach(matchCoverageScreen);

CricketMatch newMatch = new CricketMatch("Match1 Details");

subject.setMatch(newMatch);

subject.detach(matchCoverageScreen);

CricketMatch anotherMatch = new CricketMatch("Match2 Details");

subject.setMatch(anotherMatch);

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_