**Question 1:**

Code:

import java.util.ArrayList;

import java.util.List;

abstract class Observer {

protected Subject subject;

public abstract void update();

}

class Subject {

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

private int state;

public int getState() {

return state;

}

public void setState(int state) {

this.state = state;

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();

}

}

}

class BinaryObserver extends Observer {

public BinaryObserver(Subject subject) {

this.subject = subject;

this.subject.attach(this);

}

@Override

public void update() {

System.out.println("Binary String: " + Integer.toBinaryString(subject.getState()));

}

}

class OctalObserver extends Observer {

public OctalObserver(Subject subject) {

this.subject = subject;

this.subject.attach(this);

}

@Override

public void update() {

System.out.println("Octal String: " + Integer.toOctalString(subject.getState()));

}

}

class HexaObserver extends Observer {

public HexaObserver(Subject subject) {

this.subject = subject;

this.subject.attach(this);

}

@Override

public void update() {

System.out.println("Hex String: " + Integer.toHexString(subject.getState()).toUpperCase());

}

}

public class ObserverPatternDemo {

public static void main(String[] args) {

Subject subject = new Subject();

HexaObserver hexObserver = new HexaObserver(subject);

OctalObserver octObserver = new OctalObserver(subject);

BinaryObserver binObserver = new BinaryObserver(subject);

System.out.println("First state change: 15");

subject.setState(15);

// Detach the OctalObserver

subject.detach(octObserver);

System.out.println("Detaching OctalObserver.");

System.out.println("Second state change: 10");

subject.setState(10);

// Optionally reattach the observer to demonstrate it can be added back

subject.attach(octObserver);

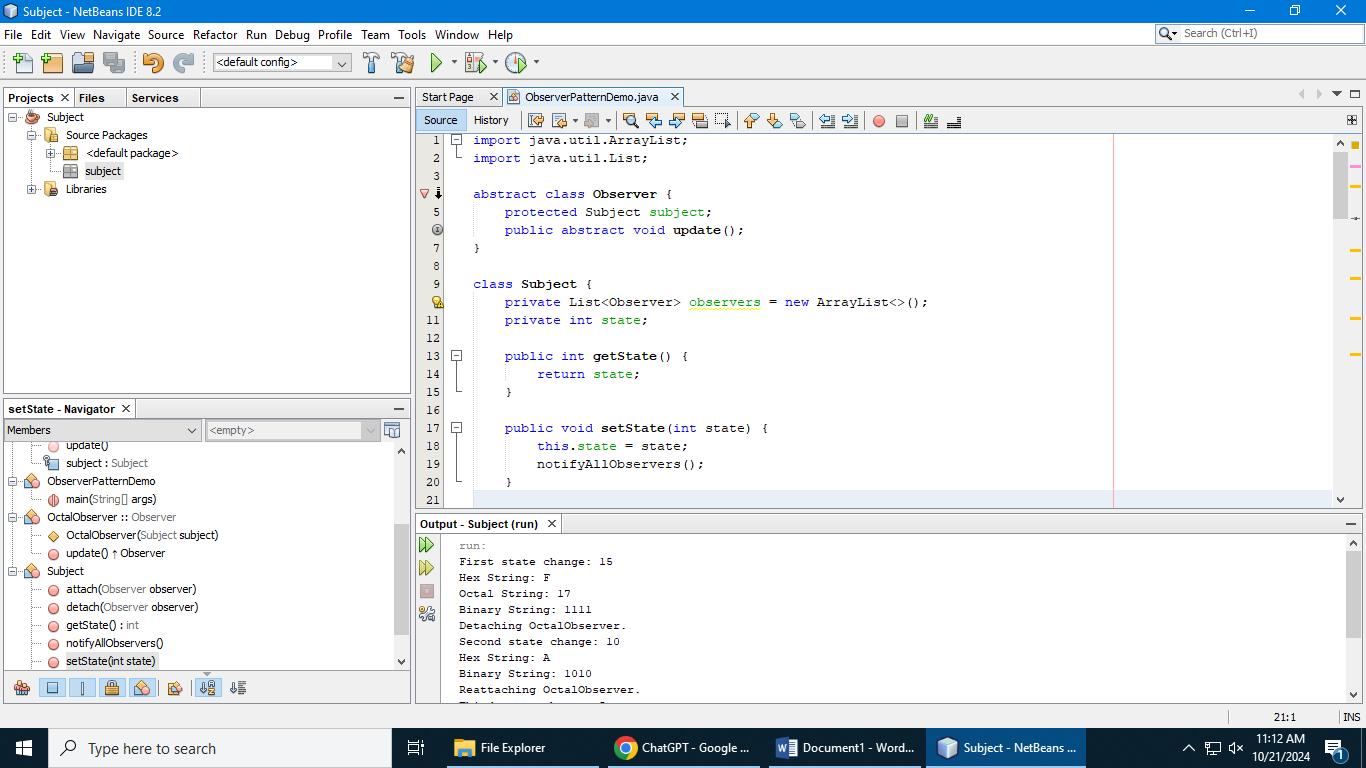
System.out.println("Reattaching OctalObserver.");

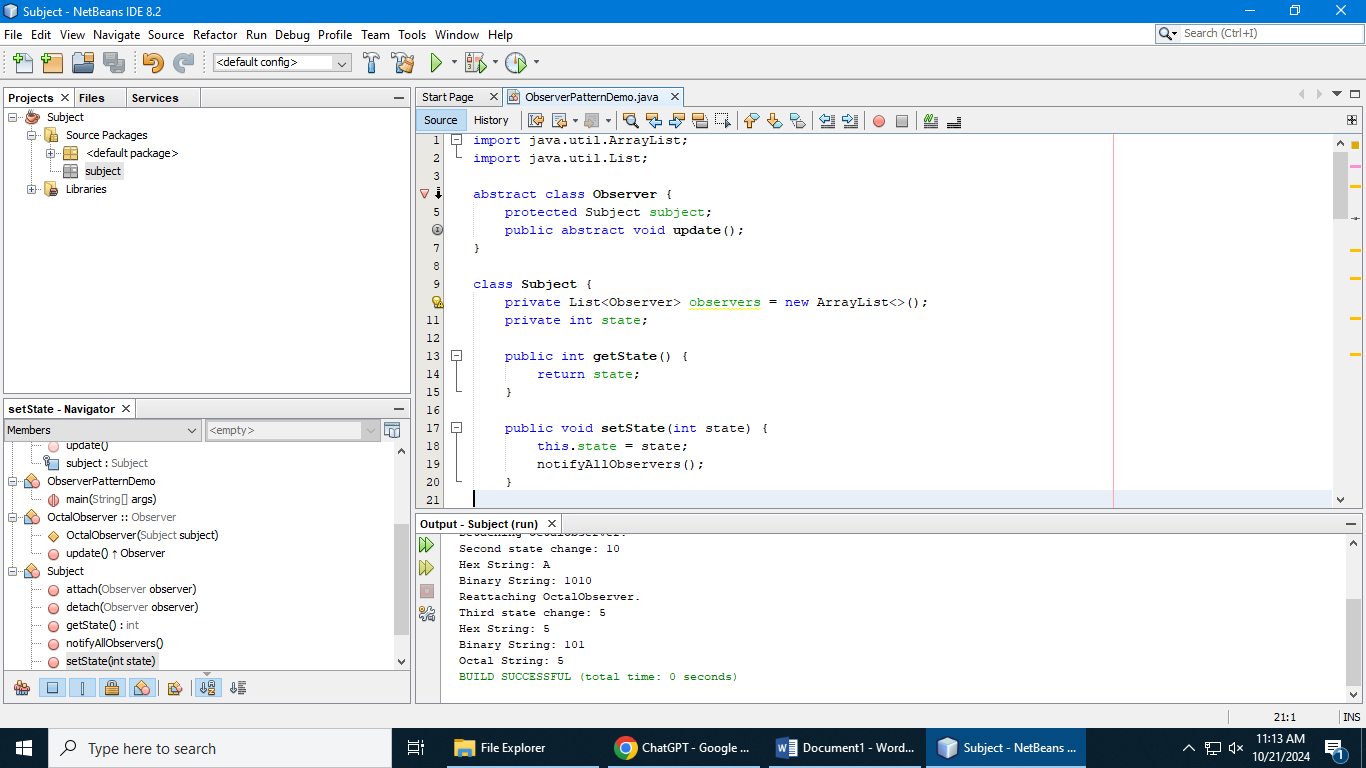
System.out.println("Third state change: 5");

subject.setState(5);

}

}





**Question 2:**

**Code:**

import java.util.ArrayList;

import java.util.List;

// Observer pattern classes

abstract class Observer {

protected Subject subject;

public abstract void update();

}

class Subject {

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

private int state;

public int getState() {

return state;

}

public void setState(int state) {

this.state = state;

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();

}

}

}

class BinaryObserver extends Observer {

public BinaryObserver(Subject subject) {

this.subject = subject;

this.subject.attach(this);

}

@Override

public void update() {

System.out.println("Binary String: " + Integer.toBinaryString(subject.getState()));

}

}

class OctalObserver extends Observer {

public OctalObserver(Subject subject) {

this.subject = subject;

this.subject.attach(this);

}

@Override

public void update() {

System.out.println("Octal String: " + Integer.toOctalString(subject.getState()));

}

}

class HexaObserver extends Observer {

public HexaObserver(Subject subject) {

this.subject = subject;

this.subject.attach(this);

}

@Override

public void update() {

System.out.println("Hex String: " + Integer.toHexString(subject.getState()).toUpperCase());

}

}

// New package: observer.eventsmanagement

class SmsSupportListener extends Observer {

private String phoneNumber;

private String defaultSms;

public SmsSupportListener(Subject subject, String phoneNumber, String defaultSms) {

this.subject = subject;

this.phoneNumber = phoneNumber;

this.defaultSms = defaultSms;

this.subject.attach(this);

}

public void setDefaultSms(String defaultSms) {

this.defaultSms = defaultSms;

}

@Override

public void update() {

if (defaultSms.length() > 160) {

System.out.println("Warning: Define a valid default SMS (length > 160 characters).");

} else {

sendSms();

}

}

private void sendSms() {

System.out.println("Sending SMS to " + phoneNumber + ": " + defaultSms);

}

}

// Main class

public class ObserverPatternDemo {

public static void main(String[] args) {

Subject subject = new Subject();

HexaObserver hexObserver = new HexaObserver(subject);

OctalObserver octObserver = new OctalObserver(subject);

BinaryObserver binObserver = new BinaryObserver(subject);

// Create SMS support listener

SmsSupportListener smsListener = new SmsSupportListener(subject, "123-456-7890", "Hello, this is a test SMS!");

System.out.println("First state change: 15");

subject.setState(15);

// Change default SMS length

smsListener.setDefaultSms("This SMS is way too long to be sent as a single message, and should trigger a warning about the length being over the limit.");

// Detach the OctalObserver

subject.detach(octObserver);

System.out.println("Detaching OctalObserver.");

System.out.println("Second state change: 10");

subject.setState(10);

}

}

