Name : Mali Anjali Prakash

Roll no: 27

Assigment no 5

Assignment title: Write a program to create multiple threads and demonstrate how two threads communicate with each other.

// Shared resource class to hold data for communication

class SharedData {

    private int data = 0;

    // Method for producer to generate data

    public synchronized void produceData() throws InterruptedException {

        while (data != 0) {

            wait(); // Wait if data is already produced

        }

        data = (int) (Math.random() \* 100); // Generate some random data

        System.out.println("Produced: " + data);

        notify(); // Notify the consumer that data is ready

    }

    // Method for consumer to consume data

    public synchronized void consumeData() throws InterruptedException {

        while (data == 0) {

            wait(); // Wait if no data is produced yet

        }

        System.out.println("Consumed: " + data);

        data = 0; // Reset data after consumption

        notify(); // Notify the producer that data is consumed

    }

}

// Producer thread that produces data

class ProducerThread extends Thread {

    private SharedData sharedData;

    public ProducerThread(SharedData sharedData) {

        this.sharedData = sharedData;

    }

    public void run() {

        try {

            for (int i = 0; i < 5; i++) {

                sharedData.produceData();

                Thread.sleep(1000); // Sleep for a while before producing next data

            }

        } catch (InterruptedException e) {

            e.printStackTrace();

        }

    }

}

// Consumer thread that consumes data

class ConsumerThread extends Thread {

    private SharedData sharedData;

    public ConsumerThread(SharedData sharedData) {

        this.sharedData = sharedData;

    }

    public void run() {

        try {

            for (int i = 0; i < 5; i++) {

                sharedData.consumeData();

                Thread.sleep(1500); // Sleep for a while before consuming next data

            }

        } catch (InterruptedException e) {

            e.printStackTrace();

        }

    }

}

public class ThreadCommunicationExample {

    public static void main(String[] args) {

        SharedData sharedData = new SharedData();

        // Create Producer and Consumer threads

        ProducerThread producer = new ProducerThread(sharedData);

        ConsumerThread consumer = new ConsumerThread(sharedData);

        // Start the threads

        producer.start();

        consumer.start();

    }

}

