
📄 Assignment Submission Template

🔹 Student Details

Name. N THIYANESWARAN
Registration No. 2117240080160
Department . CSBS
Section . C
Subject Code CS23322
Subject Name OOS

Assignment No. Assignment 3

---

1. Aim

To implement a bounded buffer (Producer-Consumer problem) using synchronization in Java, where the producer adds data to the buffer and the consumer removes data from the buffer, ensuring proper coordination. _____
2. Algorithm

Step 1: Initialize Buffer

Create a buffer variable to store data.

Use a boolean flag to indicate if data is available in the buffer.

Step 2: Producer Thread

Loop to produce data.

If the buffer is full (data available), wait.

Else, put data in buffer, set flag to true, and notify the consumer.

Step 3: Consumer Thread

Loop to consume data.

If the buffer is empty (no data), wait.

Else, get data from buffer, set flag to false, and notify the producer.

Step 4: Start Threads

Start producer and consumer threads.

Threads synchronize using synchronized, wait(), and notify().

---

3. Java Code

```
class BoundedBuffer {
    private int data;
    private boolean available = false;

    // Producer puts data
    public synchronized void put(int value) {
        while (available) {
            try {
                wait(); // wait if buffer is full
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
        data = value;
        available = true;
        System.out.println("Produced: " + value);
        notify(); // notify consumer
    }

    // Consumer gets data
    public synchronized int get() {
        while (!available) {
            try {
                wait(); // wait if buffer is empty
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
```

```
        System.out.println("Consumed: " + data);
        available = false;
        notify(); // notify producer
        return data;
    }
}

class Producer extends Thread {
    BoundedBuffer buffer;

    Producer(BoundedBuffer b) {
        buffer = b;
    }

    public void run() {
        for (int i = 1; i <= 5; i++) {
            buffer.put(i);
            try {
                Thread.sleep(400);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}

class Consumer extends Thread {
    BoundedBuffer buffer;

    Consumer(BoundedBuffer b) {
        buffer = b;
    }

    public void run() {
        for (int i = 1; i <= 5; i++) {
            buffer.get();
            try {
                Thread.sleep(800);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}
```

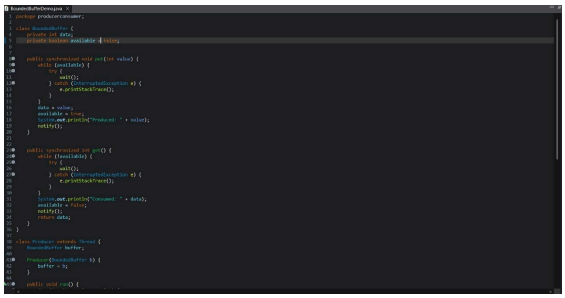
```
public class BoundedBufferDemo {
    public static void main(String[] args) {
        BoundedBuffer buffer = new BoundedBuffer();
        Producer p = new Producer(buffer);
        Consumer c = new Consumer(buffer);

        p.start();
        c.start();
    }
}
```

---

4. Output

---

5. Screenshots



---

6. GitHub Repository Link

Paste your GitHub project URL here:
🔗