

Name – Sachita Seth

UID – 22BCS12150

Section and Group – IoT_635 (B)

Q.3. Develop a ticket booking system with synchronized threads to ensure no double booking of seats. Use thread priorities to simulate VIP bookings being processed first.

```
import java.util.*;
```

```
class TicketBookingSystem {    private static final int
```

```
TOTAL_SEATS = 10;    private static final boolean[] seats = new
```

```
boolean[TOTAL_SEATS];
```

```
    public synchronized boolean bookSeat(int seatNumber, String customer) {
```

```
    if (seatNumber < 0 || seatNumber >= TOTAL_SEATS) {
```

```
        System.out.println(customer + " tried to book an invalid seat.");
```

```
        return false;
```

```
    }
```

```
    if (!seats[seatNumber]) {
```

```
    seats[seatNumber] = true;
```

```
        System.out.println(customer + " successfully booked seat " +  
seatNumber);
```

```
        return true;
```

```
    } else {
```

```
        System.out.println(customer + " tried to book an already occupied seat "  
+ seatNumber);
```

```

        return false;
    }
}

```

```

class BookingThread extends Thread {
    private final TicketBookingSystem system;
    private final int seatNumber;    private final
    String customer;

```

```

    public BookingThread(TicketBookingSystem system, int seatNumber, String
    customer, int priority) {        this.system = system;        this.seatNumber =
    seatNumber;        this.customer = customer;        setPriority(priority);
    }

    @Override    public void run() {
    system.bookSeat(seatNumber, customer);
    }
}

```

```

public class TicketBookingMain {
    public static void main(String[] args) {
        TicketBookingSystem system = new TicketBookingSystem();
        List<Thread> threads = new ArrayList<>();
    }
}

```

```

        threads.add(new BookingThread(system, 3, "VIP_1",
Thread.MAX_PRIORITY));
        threads.add(new BookingThread(system, 3, "User_1",
Thread.NORM_PRIORITY));
        threads.add(new BookingThread(system, 5, "VIP_2",
Thread.MAX_PRIORITY));
        threads.add(new BookingThread(system, 5, "User_2",
Thread.NORM_PRIORITY));
        threads.add(new BookingThread(system, 7, "User_3",
Thread.MIN_PRIORITY));

```

```

    for (Thread thread : threads) {
        thread.start();
    }

    for (Thread thread : threads) {
        try {
thread.join();
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
}

```

Output:

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

```
● VIP_1 successfully booked seat 3
  User_3 successfully booked seat 7
  User_2 successfully booked seat 5
  User_1 tried to book an already occupied seat 3
  VIP_2 tried to book an already occupied seat 5
○ PS C:\Users\dwive\OneDrive\Desktop\sunidhi java> |
```