Experiment - 4

Student Name: Manjot Singh **UID:** 23BCS12549
Branch: BE-CSE **Section/Group:** KRG-2B
Semester: 5th **Date of Performance:**23/9/25
Subject Name: Project Based Learning in Java
Subject Code: 23CSH-304

Aim: To 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.

Objective: To understand multithreading,thread synchronization, and thread priorities in Java.

Procedure:

1. Create a TicketBooking class with synchronized bookTicket() method.
2. Use a Thread class to simulate customers (normal and VIP).
3. Create threads with different priorities.
4. Start threads and observe how VIPs are handled first due to higher priority.
5. Ensure no 2 threads can book the same seat using synchronization.

Sample Input -
Thread 1: Normal User - Booking Seat 1
Thread 2: VIP User - Booking Seat 1

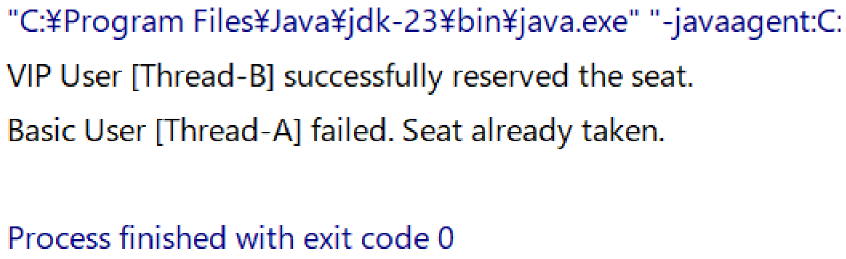
Sample Output VIP
Thread booked Seat 1
Normal Thread could not book. Seat already booked.

Code -

```
class SeatReservation {
   private boolean reserved = false;

   public synchronized void
reserveSeat(String userRole, String
threadName) {
       if (!reserved) {
          System.out.println(userRole + " ["
+ threadName + "] successfully reserved
the seat.");
      reserved = true;
    } else {
      System.out.println(userRole + " ["
+ threadName + "] failed. Seat already
taken.");
    }
  }
}

class User implements Runnable {
  private SeatReservation
reservationSystem;
  private String role;

  public User(SeatReservation
reservationSystem, String role) {
    this.reservationSystem =
reservationSystem;
    this.role = role;
  }

  @Override
  public void run() {
    reservationSystem.reserveSeat(role,
Thread.currentThread().getName());
  }
}

public class Main {
  public static void main(String[] args) {
    SeatReservation reservation = new
SeatReservation();

    Thread basicUser = new
Thread(new User(reservation, "Basic
User"), "Thread-A");
    Thread vipUser = new Thread(new
User(reservation, "VIP User"), "Thread-
B");
```

```
basicUser.setPriority(Thread.MIN_PRIORITY);

vipUser.setPriority(Thread.MAX_PRIORITY);

        basicUser.start();
        vipUser.start();
    }
}
```

Output -



```
"C:¥Program Files¥Java¥jdk-23¥bin¥java.exe" "-javaagent:C:
VIP User [Thread-B] successfully reserved the seat.
Basic User [Thread-A] failed. Seat already taken.

Process finished with exit code 0
```