

Homework #5

CS5220 – Adv Appl Devlpmnt in Java

Fall 2025

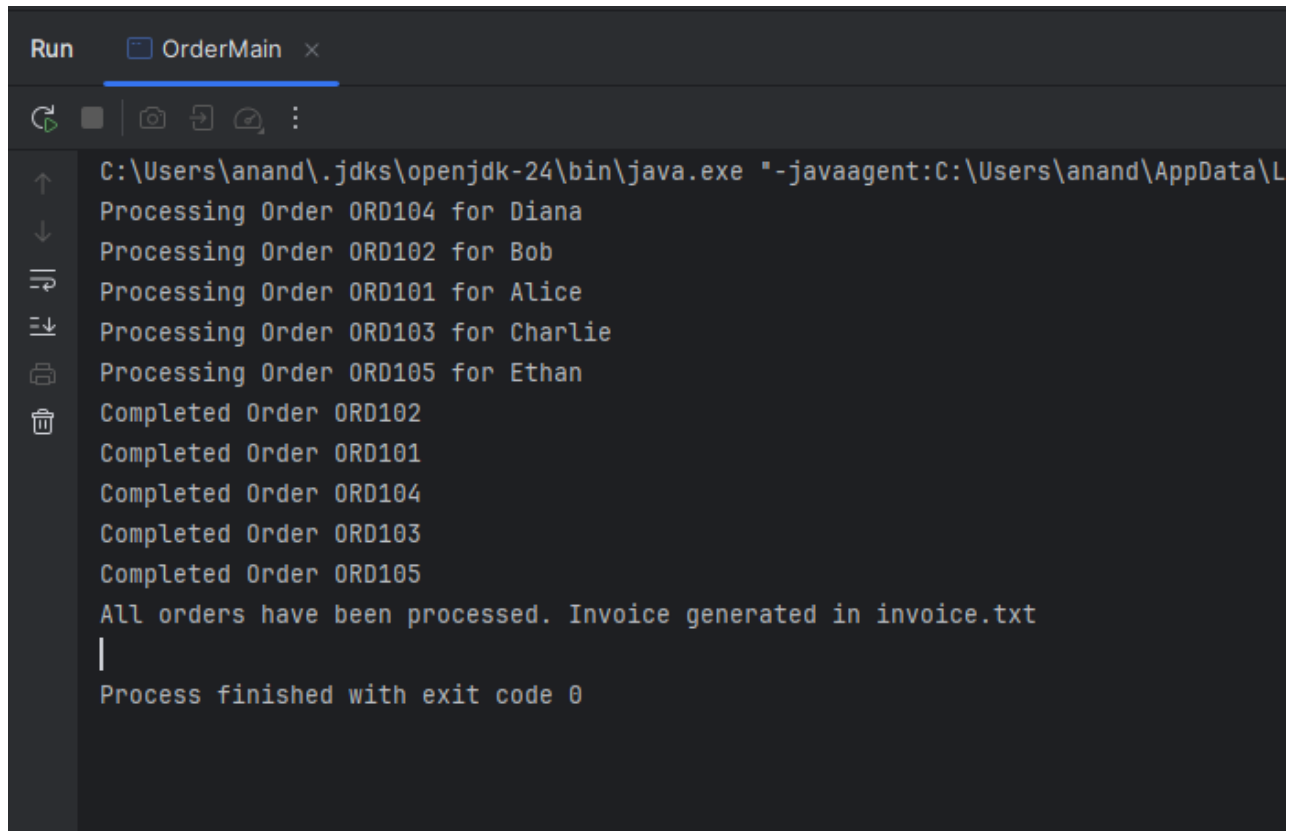
100 Points

Due: 11/07/2025, 11:59 PM

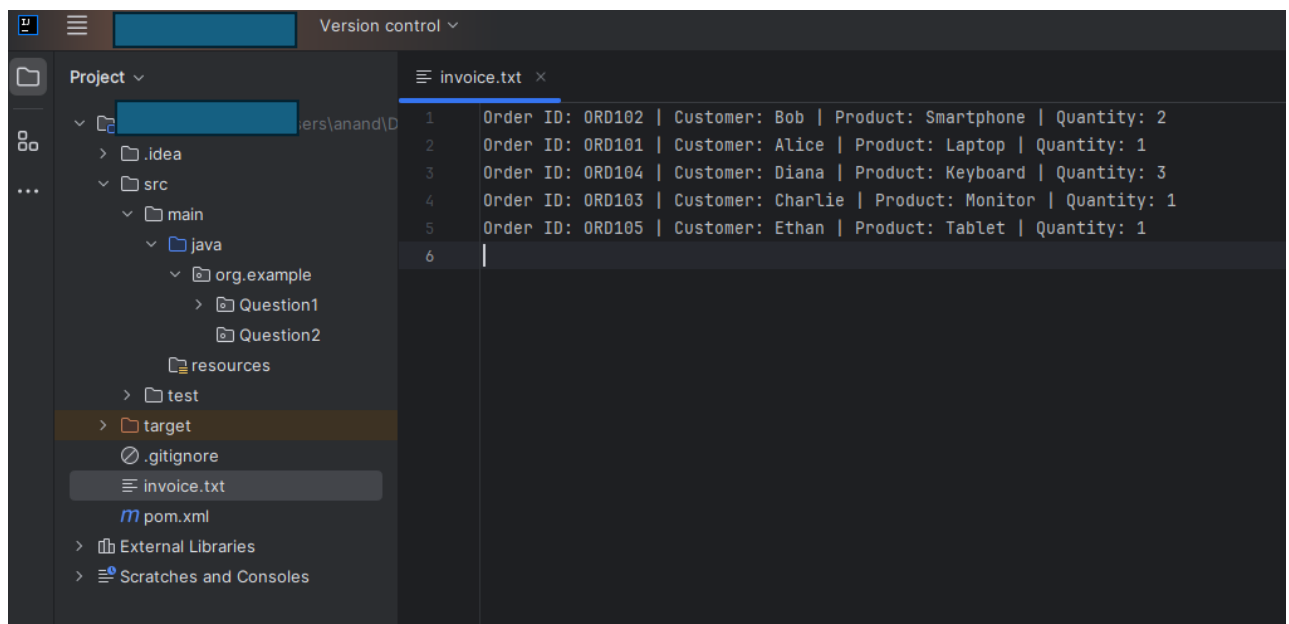
Instructions: Submit all required project files, including source code, as a single compressed archive (ZIP file). Ensure that your project is well-organized, with appropriate folder structures for source files, resources, and any required dependencies. Include a README file if necessary to explain how to run the program. Do not copy and paste code from external sources. Clearly comment on your code to explain important logic. Convert any written explanations, analysis, or required screenshots into a single PDF file and include it in the ZIP archive. Upload the ZIP file to Brightspace before the due date and time.

1. Create a program to simulate an Order Processing System using Multithreading. (100 points)
 - a. Create an Order class with the following attributes:
 - i. orderId (String)
 - ii. customerName (String)
 - iii. productName (String)
 - iv. quantity (int)
 - b. Create an OrderProcessor class that implements Runnable and includes:
 - i. A constructor that accepts an Order object.
 - ii. A run() method that simulates processing the order by printing a message "Processing Order [orderId] for [customerName]"
 - iii. Simulate a 2 second processing delay using Thread.sleep(2000).
 - iv. Write the processed order summary to a shared invoice log file (invoice.txt) using BufferedWriter in append mode.
 - v. Use a synchronized block to ensure thread-safe writing to the shared file resource.
 - c. Implementation in main():
 - i. Create at least 5 sample Order objects.
 - ii. Create and start a separate thread (OrderProcessor) for each order.
 - iii. After all threads finish, print "All orders have been processed. Invoice generated in invoice.txt".
 - d. Points to consider:

- i. The shared resource in this simulation is the file invoice.txt, where each thread appends a processed order.
- ii. Use Thread.join() to wait all threads to complete before printing “All orders have been processed. Invoice generated in invoice.txt”.
- iii. Use FileWriter + BufferedWriter in append mode for writing order summaries.



```
Run OrderMain x
C:\Users\anand\.jdk\openjdk-24\bin\java.exe "-javaagent:C:\Users\anand\AppData\Local\
Processing Order ORD104 for Diana
Processing Order ORD102 for Bob
Processing Order ORD101 for Alice
Processing Order ORD103 for Charlie
Processing Order ORD105 for Ethan
Completed Order ORD102
Completed Order ORD101
Completed Order ORD104
Completed Order ORD103
Completed Order ORD105
All orders have been processed. Invoice generated in invoice.txt
|
Process finished with exit code 0
```



```
Project
  .idea
  src
    main
      java
        org.example
          Question1
          Question2
          resources
        test
        target
          invoice.txt
          pom.xml
      External Libraries
      Scratches and Consoles

Version control
invoice.txt
1 Order ID: ORD102 | Customer: Bob | Product: Smartphone | Quantity: 2
2 Order ID: ORD101 | Customer: Alice | Product: Laptop | Quantity: 1
3 Order ID: ORD104 | Customer: Diana | Product: Keyboard | Quantity: 3
4 Order ID: ORD103 | Customer: Charlie | Product: Monitor | Quantity: 1
5 Order ID: ORD105 | Customer: Ethan | Product: Tablet | Quantity: 1
6
```