Objective:

This assignment has been designed for students to apply appropriate concurrent program methods in **implementing** a concurrent program from a program specification.

Learning Outcomes

ON COMPLETION OF THIS ASSIGNMENT, YOU SHOULD BE ABLE TO DEMONSTRATE THE FOLLOWING LEARNING OUTCOME(S):

No.	Learning Outcome	Assessment
1	Explain the fundamental concepts of concurrency and parallelism in the design of a concurrent system (C2, PLO1)	Exam
2	**Apply the concepts of concurrency and parallelism in the construction of a system using a suitable programming language. (C3, PLO2)**	**Individual Assignment (System)**
3	**Explain the safety aspects of multi-threaded and parallel systems (A3, PLO6)**	**Individual Assignment (Report)**

Programme Outcomes (PO):

PLO2 Cognitive Skills - This relates to thinking or intellectual capabilities and the ability to apply knowledge and skills. The capacity to develop levels of intellectual skills progressively begins from understanding, critical/creative thinking, assessment, and applying, analysing, problem solving as well as synthesizing to create new ideas, solutions, strategies, or new practices. Such intellectual skills enable the learner to search and comprehend new information from different fields of knowledge and practices.

Individual Assignment - Report (20%):

Question No.	Topic	Question Vs Taxonomy					PLO
		Affective Level					
		1	2	3	4	5	
		SQ	SQ	SQ	SQ	SQ	
1	Introduction and background			**20%**			6
2	Explanation of the safety aspects of multi-threaded system implemented			**30%**			6
3	Justification of coding techniques implemented.			**30%**			6
4	Depth of discussion of concurrency concepts.			**20%**			6
	Total			**100%**			

Individual Assignment - System (25%):

Question No.	Topic	Question Vs Taxonomy						PLO
		Cognitive Level						
		1	2	3	4	5	6	
		SQ	SQ	SQ	SQ	SQ	SQ	
1	Appropriateness of coding techniques used to implement design with appropriate comment lines in source codes			**20%**				2
2	Appropriateness of the Java concurrent programming facilities used.			**20%**				2
3	Program runs appropriately with basic requirements			**20%**				2
4	Additional requirements met			**20%**				2
5	Explanations of concurrency concepts implemented with relevant code samples			**20%**				2
	Total			**100%**				

Submission Requirements:

Assignment Handout Date : 6th May 2025

Assignment Due Date : 2nd August 2025

Case Study

SwiftCart E-commerce Centre

You have been assigned to simulate the operations of SwiftCart, a highly automated e-commerce centre that handles online orders for a major shopping platform.

**********Basic Requirements**********

SwiftCart is in Selangor and is built to operate with minimal human intervention. It consists of the following **six key sections**:

1. **Order Intake System**
    a. Orders arrive from the online platform at a rate of 1 order every **500ms**.
    b. Each order is verified for payment, inventory availability, and shipping address.
2. **Picking Station**
    a. Robotic arms pick items from shelves and place them into order bins.
    b. Up to 4 orders can be picked at a time.
    c. Orders are verified for missing items.
3. **Packing Station**

a. Completed bins are packed into shipping boxes (1 order at a time).

b. A scanner checks each box to ensure contents match the order.

4. **Labelling Station**

    a. Each box is assigned a shipping label with destination and tracking.

    b. Boxes pass through a quality scanner (1 at a time).

5. **Sorting Area**

    a. Boxes are sorted into batches of 6 boxes based on regional zones.

    b. Batches are loaded into transport containers (30 boxes per container).

6. **Loading Bay & Transport**

    a. 3 autonomous loaders (AGVs) transfer containers to 2 outbound loading bays.

    b. Trucks take up to **18 containers** and leave for delivery hubs.

    c. If both bays are occupied, incoming trucks must wait.

**********Additional Requirements**********

Defective Orders

- Orders may be rejected at any stage (e.g., out-of-stock items, packing errors, mislabelling).
- Defective orders are removed by a reject handler and logged.

Capacity Constraints

- The loading bay has space for only 10 containers. If full, packing must pause.

Autonomous Loaders

- Only 3 loaders are available and can break down randomly (simulate with thread stalls).
- Trucks can only be loaded when a loader and bay are free.

Concurrent Activities

- Loaders and outbound trucks operate concurrently.
- Simulate congestion: e.g., 1 truck is waiting while both loading bays are in use.

The Statistics

When all trucks have departed for the day, the system should print a detailed report:

Confirm that all orders, boxes, and containers have been cleared.

Maximum, minimum and average loading and wait time per truck.

Total number of orders processed, rejected, boxes packed, containers shipped and trucks dispatched

Deliverables:

Simulate 600 orders.

Orders arrive every 500ms.

Thread classes for order intake, picking, packing, labelling, sorting, loader, and truck.

Random faults in loaders (simulate breakdowns).

Rejection logic at multiple stages.

Blocking/waiting when container capacity or loading bays are full.

Logging for each major activity per thread.

Sample Output

To see what is happening dynamically you must have output from the sections, the respective entities reporting all their major events.

Add information about which process/thread is doing the output. This way you can see if a process/thread acts for another, which is strictly forbidden, but is a common error for Java solutions (objects are not processes!).

OrderIntake: Order #155 received (Thread: OrderThread-1)

PickingStation: Picking Order #155 (Thread: Picker-3)

PackingStation: Packed Order #155 (Thread: Packer-1)

LabellingStation: Labelled Order #155 with Tracking ID #A415 (Thread: Labeller-2)

Sorter: Added Order #155 to Batch #22 (Thread: Sorter-1)

Loader-2: Moving Container #7 to Loading Bay-1

Truck-1: Fully loaded with 18 containers. Departing to Distribution Centre.

Truck-2: Waiting for loading bay to be free.

Supervisor: Dispatch paused. 20 containers at bay – waiting for truck.

Implementation

You should implement your simulation in Java.

The simulation run should take about **5 minutes** to simulate.

Documentation for System (Report)

The documentation should detail the system implementation and testing.

1. Basic requirements met:
   - List of requirements met.
     - Short explanation of concurrency concepts (atomic statements, synchronization, etc) implemented.
     - Code snippet of the Java concurrent programming facilities implemented.

2. Additional requirements met:
   - List of requirements met.
     - Short explanation of concurrency concepts (atomic statements, synchronization, etc) implemented.
     - Code snippet of the Java concurrent programming facilities implemented.

3. Requirements which were **NOT** met:
   - List of basic requirements.
   - List of additional requirements

Should not exceed 1500 words excluding references/appendix/coding.

Submission for System

- *Java files required to run the simulation.*
- *Video of the simulation running. Maximum 5 minutes per person.*
  - *Simulate scenarios as stated above.*
  - *Show which requirements are met in the output and corresponding code.*

Both code and presentation Zipped into a single zip file named TP0XXXXX CCP.zip

Marking Scheme (NOT for student's documentation guidance)

Report (20%)

Criteria	**Total marks**	**Marks awarded**
Assumptions [LO3-PO6]	20	
Explanation of the safety aspects of multi-threaded system implemented [LO3-PO6]	30	
Justification of coding techniques implemented [LO3-PO6]	30	
Depth of discussion of concurrency concepts [LO3-PO6]	20	
TOTAL MARKS	100	

System (25%)

Criteria	**Total marks**	**Marks awarded**
Appropriateness of coding techniques used to implement design with appropriate comment lines in source codes *	20	
Appropriateness of the Java concurrent programming facilities used.	20	

Program runs appropriately with basic requirements *	20	
Additional requirements met *	20	
Explanations of concurrency concepts implemented with relevant code samples *	20	
TOTAL MARKS	100	

*Based on video presentation of the simulation.