Data Structures and Algorithms
Department of Computer Science
University of Engineering and Technology, Lahore

Open Ended Lab (Final Project) Total Marks:40 Deadline:30-December-2025

Course Code & Title: **CSC-200L: Data Structures & Algorithms Lab**

1. Course Learning Outcomes (CLOs)

CLO	Description	PLO	Domain	Domain Level
CLO 3	Design and develop solutions for complex computing problems using suitable data structures	PLO4	Cognitive	Creation
CLO 4	Comply with ethics and professional practices in computing.	PLO9	Affective	Valuing

2. Schwab/Herron Levels of Laboratory Openness

Level	Methodology	Description	Problem	Ways & Means	Answers
2	Guided Inquiry	Students investigate an instructor-defined real-world problem using self-designed code structures and algorithms.	Given	Open	Open

3. Problem Statement

Design & Implementation of an Intelligent Parcel Sorting, Routing & Tracking System (Courier Logistics Engine)

Problem Description:

SwiftEx Logistics, a nationwide courier company, is facing major challenges. Every day, thousands of parcels arrive at its central hubs, some need to be delivered overnight, some are heavy, some are fragile and all must navigate a complex network of roads, warehouses, and delivery zones.

Delays happen roads get blocked, parcels are misplaced, and sometimes packages take unnecessarily long routes because the system cannot adapt to changing traffic conditions.

To overcome these challenges, SwiftEx has decided to replace their outdated manual system with a smart, dynamic **Intelligent Courier Logistics Engine**. Your team of Four students has been tasked with building this system using **C++ and Data Structures & Algorithms**.

The system should simulate a real-world logistics engine: **sorting parcels intelligently, selecting optimal routes, tracking every movement, and managing courier operations efficiently** across thousands of parcels in real-time.

The system must support:

1. Intelligent Parcel Sorting Module

Students will design mechanisms to organize parcels based on:

- Delivery priority (overnight, 2-day, normal)
- Weight categories
- Destination / zones

The system must also support real-time insertion and withdrawal of parcels.

2. Parcel Routing Module

The system should compute:

- Shortest delivery routes
- Multiple alternative routes
- Handling of blocked/overloaded paths

Dynamic route recalculation when nodes/edges change

3. Parcel Tracking System

Simulate the complete movement lifecycle of a parcel, including:

- Dispatch
- Loading/unloading
- Return to sender
- Delivery attempts
- Final delivery

The system must provide current parcel status and a full timeline/history of events.

4. Courier Operations Engine

Implement realistic courier workflow handling:

- Pickup queues
- Warehouse queues

- Transit queues
- Rider assignment based on priority and load
- Missing parcel detection
- Undo/replay of operations through logs

Students are free to choose their data structures but must justify their selection and explain performance considerations.

4. Instructions

- This is a Final Project; each team must consist of **4 students**.
- Your project will be evaluated according to the rubrics provided.
- Implementation should prioritize custom data structure logic; use STL only when justified.
- Implementation must use multiple data structures in meaningful ways.

5. Deliverables

Students must submit:

- Complete & Well-Structured Source Code
- Documentation, including:
    - UML Diagrams
    - System Architecture Diagram
    - Flowcharts
    - Data Structure & Algorithm Justification
    - Project Report
    - CLI or Menu-Based Interactive System
- Complete Demonstration of All Modules
- Video demonstration (Posted on LinkedIn)

Open Ended Lab Assessment Rubrics

Criteria	Excellent (5)	Good (4)	Satisfactory (3)	Needs Improvement (2)	Poor (1)
Use of Data Structures (CLO1 & CLO3)	Uses DS efficiently (graphs, heaps, hash tables, queues, lists, trees). Custom implementations, no dependency on STL shortcuts.	Uses 4–5 DS with correct logic and partial optimization.	Uses 3 DS, mostly correct but lacks optimization.	Uses only basic DS (arrays/lists) without depth.	Bare minimum DS or incorrect implementations.
Algorithmic Complexity & Performance (CLO2)	Provides full time/space analysis + experimental results + optimized algorithms.	Provides theoretical analysis with minor mistakes.	Basic complexity discussion.	Weak explanation; incorrect analysis.	No analysis.
System Design & Architecture	Highly modular, scalable, clean layers, UML is accurate.	Good structure, minor issues.	Acceptable design but lacks clarity.	Poor structure; hard to follow.	No meaningful design.
Functionality & Module Completion	All 4 modules fully functional with real-time simulation.	3 modules functional.	2 modules functional.	Only partial features implemented.	Project mostly non-functional.
Code Quality & Documentation (CLO4)	Clean, readable, fully documented,	Good code and comments.	Minimal comments; moderate mistakes.	Poorly commented; messy code.	No documentation.

	diagrams included.				
User Interaction	Smooth CLI/menus, all operations intuitive.	Mostly usable.	Works but confusing.	Hard to use.	No functional interface.