National University of Computer & Emerging Sciences, Karachi
Spring-2025, School of Computing (BSCS, BSSE, BSAI, BSCY)

Assignment # 3

Subject: Object Oriented Programming **Total Marks: 40**	**Post Date: 21st April 2025** **Due Date: 2nd May 2025**
Course Instructors: : Ms. Bakhtawar, Ms. Sobia Iftikhar, Ms. Abeer Gauher, Ms. Nida, Mr. Basit Ali, Ms. Atiya Johkio, Ms. Sumaiyah Zahid, Mr. Minhal Raza, Ms. Abeeha Sattar, Ms. Rafia.	

Instructions to be strictly followed.

1. **Each student should submit these files:**
    a. **A zip of all source files named as "A3-Q#[StudentID]" where # is the question number and Student ID is your ID.**
    b. **A DOC file where they copy code for each question and screen shot of the output. This document contains all the questions, answer codes and output in sequence. Name this document as “A3-[StudentID].docx”.**
    c. **All the submissions will be made on Google Classroom.**
2. **Each output should have STUDENT ID and NAME of the student at the top.**
3. **It should be clear that your assignment would not get any credit if the assignment is submitted after the due date.**
4. **Zero grade for plagiarism (copy/ cheating) and late submissions.**

Scenario:

FAST-NUCES has been running a dedicated Point Service for faculty and students to provide daily pick and drop service from various parts of the city. Earlier, the entire system was managed manually by a transport manager. However, the transport manager recently resigned. Instead of hiring a new resource, the university administration has decided to digitize the transport operations by developing an automated Transport Management System.

As a junior intern currently studying Object-Oriented Programming, you have been assigned the task of designing and implementing this system using OOP concepts. The goal is to build a software system that replicates the key responsibilities of the transport manager: assigning drivers, managing vehicle routes, allocating seats, and handling bookings all through code.

FAST-NUCES currently outsources its transport service to two third-party providers: Nadeem Transporter and Zulfiqar Transporter. Each transporter maintains a fleet of vehicles (called Points), a pool of drivers, and a set of assigned routes. Vehicles can either be buses (with 52 seats) or coasters (with 32 seats), and some of them are air-conditioned. Each vehicle follows a fixed route, has its own driver, and carries both faculty and students. Seats in these vehicles are designated either for students or faculty to prevent overlap.

Each vehicle (Point) has a list of seats which can be booked by registered users only after they've made a payment. Faculty and student users have different fare structures with AC vehicles charging an additional Rs. 2000. Bookings are limited to one seat per user per month and users must have completed their payment before the seat is reserved. Each booking keeps track of the selected vehicle, route, user, and seat.

Drivers have personal details and valid licenses, and can only be assigned to one vehicle at a time. Routes have information about the start and end locations and the distance covered. If a route covers a long distance (based on a threshold), it is marked accordingly.

To make the system maintainable and reusable, the university also wants the system to manage lists of various entities (such as users, vehicles, and bookings) using a generic data structure. If a requested entity is not found, the system should raise an appropriate exception. Moreover, all important operations, such as saving user data, bookings, and vehicle states must be stored and loaded from files using file handling, allowing the system to persist across sessions.

You are expected to design classes that reflect real-world responsibilities and relationships. For instance, a vehicle cannot exist without a driver (composition), but can share routes and transporter affiliations (aggregation). The system must also include exception handling to deal with common errors like booking without payment, seat unavailability, or role-based seat violations.

The final system should be able to register faculty and student users, assign drivers to vehicles, define routes, handle seat bookings, calculate fares, and display all relevant information through meaningful output.