National University of Computer & Emerging Sciences, Karachi
Spring-2025 School of Computing
Quiz No. 3
8th May 2025

Course Code: CS-1004	Course Name: Object Oriented Programming
Instructor Name / Names: Ms. Atiya Jokhio	
Section-H	Student-ID:

Time Allowed: 30 minutes. **Total Points**: 10

TYPE B

Question:1 You need to complete the code for counting the Number of Lines in a Text File. **[5 points]**

```
#include <iostream>
#include <string>
//include missing header file here

int main() {
 std::ifstream inputFile("test.txt");  // Opening the file named "test.txt" for reading

 if (inputFile.is_open()) {  // Checking if the file was successfully opened
  std::string line;  // Declaring a string variable to store each line of text
  int lineCount = 0;  // Initializing a variable to count lines

  while (std::getline(inputFile, line)) {  // Loop through each line in the file
   lineCount++;  // Incrementing line count for each line read
  }


  inputFile.close();  // Closing the file after counting lines
```

Question:2 **[5 points]**

Design an **abstract base class User** for a ride-hailing platform like **InDrive**. Which contain a **concrete method** displayUserType() that prints the type of user (like "Guest", "Rider", or "Driver"). This method should be implemented inside the base class using a protected string or enum passed through the constructor and contain a **pure virtual function** requestOrOfferRide() that must be overridden in all derived classes to define how each user type interacts with the app.
Design **derived classes:** Each class must override requestOrOfferRide() with behavior specific to that role.

1. **GuestUser**: Can only browse ride options but **cannot request** or **offer** rides.
2. **RiderUser**: Can **request a ride**, view driver options, and negotiate prices.
3. **DriverUser**: Can **offer rides**, accept passenger requests, and view pickup locations.

Demonstrate runtime polymorphism:

- Create an array of User* pointers.
  (Example: User users[numUsers];)*

- Dynamically allocate different user types and store them in the array.
- Use a loop to invoke both displayUserType() and requestOrOfferRide() methods using only the base class pointers.
- After the loop, **delete all dynamically allocated objects** to prevent memory leaks.