National University of Computer & Emerging Sciences, Karachi
Spring-2025 School of Computing
Quiz No. 3
6th May 2025

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

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

TYPE A

Question:1 Complete the missing lines/line of code **[5 points]**

```
class Customer {
private:
    int id;
    string name;
    // Parameterized constructor
    Customer(int cid, string cname, string cemail) {
        id = cid;
        name = cname;
        email = cemail;
    }

    // Method to input details
    void inputDetails() {
        cout << "Enter Customer ID: ";
        cin >> id;
        cin.ignore();  // to handle newline after int input
        cout << "Enter Customer Name: ";
        getline(cin, name);
        cout << "Enter Customer Email: ";
        getline(cin, email);
    }

    void displayDetails() const {
        cout << "ID: " << id << ", Name: " << name << ", Email: " << email << endl;
    }

    // Method to save customer data to file
    void saveToFile() const {
        ofstream outFile("customers.txt", ios::app); // append mode


//method to display all customer records from file
void displayAllCustomers() {
        ifstream inFile("customers.txt");
        if (inFile.is_open()) {
            string line;
            cout << "\nAll Customer Records:\n";
            cout << "----------------------\n";
            while (getline(inFile, line)) {
                    cout << "ID: " << sid << ", Name: " << sname << ", Email: " << semail << endl;
                }
            }
```

Question:2 **[5 points]**

Design an abstract base class User for an online shopping system. Which includes a **concrete method** displayUserType() that prints the type of user (this function is implemented inside the base class) and a **pure virtual method** browseItems() which is to be overridden in all derived classes.

Create three derived classes: Each class must provide its specific version of browseItems().

1. **GuestUser**
2. **RegisteredUser**
3. **PremiumUser**

Demonstrate runtime polymorphism by:

- Creating an array of User*. *(Hint: User* users[numUsers];)*
- Storing different types of users in the container.
- Looping through the users and calling both displayUserType() and browseItems() using only User* pointers.
- Ensure proper memory management by deleting all dynamically allocated objects at the end of the program.