National University of Computer & Emerging Sciences, Karachi
Spring-2025 School of Computing
Quiz No. 1
13th February 2025

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

Time Allowed: 30 minutes. **Total Points**: 12

TYPE B

Question:1 Suppose your program contains the following class definition (along with definitions of the member functions): **[6 points]**

```
class LibraryBook {
public:
   LibraryBook(int bookID, char shelfCode);
   LibraryBook();
private:
   int bookID;
   char shelfCode;
};
// Constructor Definitions
LibraryBook::LibraryBook(int bookID1, char shelfCode1) {
   bookID1 = bookID;
   shelfCode1 = shelfCode; }
LibraryBook::LibraryBook() {
   bookID = 0;
   shelfCode = 'X'; }
```

Which of the following are legal? If any statement is illegal then write the corrected (legal) statement.

1. LibraryBook book1(101, 'A'); // legal
2. LibraryBook book2; // legal
3. LibraryBook book3(); **// illegal LibraryBook book3;**
4. book1 = LibraryBook(202, 'B'); // legal
5. book1 = LibraryBook(); // legal
6. book1 = LibraryBook; **// illegal Book1 is a type, not an instance or function that returns an instance.**

Question:2 **[6 points]**

Create a class called Employee that includes three pieces of information as data members—a first name (type string), a last name (type string) and a monthly salary (type int).

Your class should have a constructor that initializes the three data members. Provide a *set* and a *get* function for each data member. If the monthly salary is not positive, set it to 0. Design a method "Increment" which give each Employee a 10 percent raise. Write a test program that demonstrates class Employee's capabilities. Create two Employee objects and display each object's *yearly* salary. Display each Employee's yearly salary using member function. Your code needs to destroy the objects and avoid memory leak.

```
#include <iostream>
#include <string>
using namespace std;
```

```
class Employee {
private:
    string firstName;
    string lastName;
    int monthlySalary;

public:
    // Constructor
    Employee(string fName, string lName, int salary) {
        firstName = fName;
        lastName = lName;
        setMonthlySalary(salary);
    }

    // Setters
    void setFirstName(string fName) {
        firstName = fName;
    }

    void setLastName(string lName) {
        lastName = lName;
    }

    void setMonthlySalary(int salary) {
        if (salary > 0)
            monthlySalary = salary;
        else
            monthlySalary = 0;
    }

    // Getters
    string getFirstName() const {
        return firstName;
    }

    string getLastName() const {
        return lastName;
    }

    int getMonthlySalary() const {
        return monthlySalary;
    }

    // Increment salary by 10%
    void Increment() {
        monthlySalary = static_cast<int>(monthlySalary * 1.10);
    }

    // Display yearly salary
    void displayYearlySalary() const {
        cout << firstName << " " << lastName << "'s yearly salary: " <<
(monthlySalary * 12) << endl;
    }

    // Destructor
    ~Employee() {
        cout << "Destructor called for " << firstName << " " << lastName <<
endl;
    }
};

// Test program
int main() {
    // Create two employees
    Employee* emp1 = new Employee("Alice", "Johnson", 3000);
    Employee* emp2 = new Employee("Bob", "Smith", -4000); // Should set
salary to 0

    cout << "Before raise:" << endl;
    emp1->displayYearlySalary();
    emp2->displayYearlySalary();

    // Give 10% raise
    emp1->Increment();
    emp2->Increment();

    cout << "\nAfter 10% raise:" << endl;
    emp1->displayYearlySalary();
    emp2->displayYearlySalary();

    // Delete objects to prevent memory leak
    delete emp1;
    delete emp2;

    return 0;
}
```