

Object Oriented Programming Lab (CL1004)

Conveniently, your best friend

Date:

Course Instructor(s)

Mr. Muhammed Monis, Mr. Talha Shahid

Lab Mid Exam (A)

Total Time: 90 minutes

Total Marks: 40

Total Questions: 03

Semester: SP-2025

Campus: Karachi

Dept: Computer Science

managed efficiently.

Requirements:

- **Account Creation:**
 - Create an account with a unique number and an initial balance of zero.
 - Ensure that each new account is assigned a unique, auto-incremented account number that is immutable.
- **Active Account Tracking:**
 - Properly track the total number of active accounts.
 - Provide a method to retrieve the current count of active accounts.
- **Memory Management:**
 - Dynamically allocate account objects using references.
 - Ensure proper deletion of account objects to avoid memory leaks.
- **Account Details and Operations:**
 - Implement functionality to display account details, including the account number, balance, and bank name.
 - Provide deposit and withdrawal functionalities that update the account balance accordingly (utilizing references where applicable).

Implement your solution in C++ and thoroughly test it by simulating account creation and deletion. Ensure that the total active account count updates correctly after each operation and that all dynamically allocated memory is efficiently managed.

LLO # 2: Understanding and implementing Has-A Relationship concepts between classes with dynamic memory and object management.

Q2. [16 marks | 8 weightage]

You are tasked with designing an Office Management System that keeps track of Employees and their respective Departments. Each employee belongs to a department, establishing a Has-a relationship between the Employee and Department classes.

Your implementation should allow:

Tracking Departments:

Ensure that each department has a unique name.

Provide a mechanism to monitor the total number of departments created.

Managing Employees:

Assign every employee a unique identifier that automatically increments.

Ensure that each employee's name is immutable once set.

Guarantee that an employee is permanently assigned to a specific department upon creation.

Efficient Memory Management:

Dynamically allocate memory for employee and department objects.

Properly deallocate memory when objects are no longer needed, ensuring no memory leaks occur.

LLO # 1: Understand and implement static data members, static member functions, constant members with data immutability

Q1. [10 marks | 5 weightage]

You are tasked with designing a Bank Account Management System where each account is assigned a unique account number that automatically increments when a new account is created. The bank also requires a way to track the total number of active accounts at any given time.

To ensure security and data integrity:

- The bank name should remain constant for all accounts.
- The account number, once assigned, should not be modified.
- The system must efficiently manage account creation and deletion while keeping track of active accounts.
- Objects should be dynamically allocated and

(Contd. At Page 2)

Ensuring Constraints:

The employee's department assignment should be fixed upon creation and not modifiable.

The system should dynamically track the total number of employees and departments.

Considerations:

How will you ensure that each employee has a unique ID that automatically increments?

How can the system track the number of departments and employees dynamically?

What is the best way to establish a Has-a relationship between employees and departments?

How will you prevent modifications to an employee's name or department after they are assigned?

What mechanisms should be in place to properly allocate and free memory when objects are created and deleted?

Implement your solution in C++ and test it by creating multiple employees and departments, verifying that the total count for each is updated correctly, and ensuring that memory is managed efficiently.

CLO # 3: Apply Inheritance (Is-a Relationship) in C++ to build an extensible University Staff Management System, using dynamic memory allocation, pointer references, and pure virtual functions for polymorphism.

Q3 [14 marks | 7 weightage]

You are tasked with designing a University Staff Management System that efficiently manages different types of staff members, such as Professors and Administrative Staff. Since all staff members share common attributes, your implementation should avoid redundancy.

Your system should:

1. Manage staff members, ensuring each has a unique identifier and an unchangeable name.
2. Differentiate between staff roles, such as Professors (who have subject specializations) and Administrative Staff (who belong to a department).
3. Enforce structured behavior, ensuring that every staff member can display their details appropriately.
4. Efficiently allocate and deallocate memory, properly managing dynamically created objects.
5. Demonstrate move semantics by utilizing move constructors where appropriate.

Think about:

- a. How does inheritance help in reducing redundancy when dealing with different staff roles?
- b. How can you ensure that every staff member has a unique and immutable staff ID?
- c. What mechanism will enforce that all staff types implement a function to display their details?
- d. How should memory be handled when objects are dynamically allocated?

Implement your solution in C++ and test it by creating multiple professors and administrative staff members while ensuring proper memory management and correct staff count tracking. Use comments in your code to explain how inheritance helps in reducing redundancy and improving maintainability. Also create an object and move an original object details to another while also deleting the old object..