CL-103

Computer Programming

Lab # 8

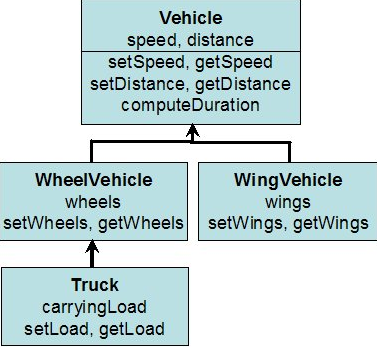
|  |
| --- |
| Objectives:  * Multilevel inheritance * Single level inheritance * Composition * **Accessor** * **Mutator functions** * Practice Problems. |
|  |

**Note: Carefully read the following instructions (***Each instruction contains a weightage***)**

1. There must be a block of comments at start of every question's code; the block should contain brief description about functionality of code.
2. Proper indentation of code is essential
3. Variable name should be meaningful
4. Make a Microsoft Word file and past all of your C++ code with screenshot of outputs in MS word.
5. First think about statement problems and then write/draw your logic on copy.
6. After copy pencil work, code the problem statement on MS Studio C++ compiler.
7. At the end when you done your today lab tasks, attached only MS word file and make your submission on slate.
8. Late and email submission is not accepted. All tasks must be submitted during the lab time.

|  |  |
| --- | --- |
| **Problem 1: (multilevel inheritance, accessor and mutator functions)** |  |

Design the following classes with given variables and methods using inheritance.



Class variables

Class methods

Setters and Getters are used to access private variables of class to set and get their values .

**Note:** In Main, use the object of truck class to access all the data and functions of all inherited classes .And Create the object of wing Vehicle to access all the functions of Vehicle.

Explain your work with proper output.

|  |
| --- |
| **Problem 2: (Single Level Inheritance, accessor and mutator functions)** |

A retail store has a preferred customer plan where customers may earn discounts on all their purchases. The amount of a customer’s discount is determined by the amount of the customer’s cumulative purchases in the store.

**•** When a preferred customer spends $500, he or she gets a 5% discount on all future purchases.

**•** When a preferred customer spends $1,000, he or she gets a 6% discount on all future purchases.

**•** When a preferred customer spends $1,500, he or she gets a 7% discount on all future purchases.

**•** When a preferred customer spends $2,000 or more, he or she gets a 10% discount on all future purchases.

Design a class named PreferredCustomer, which is derived from the CustomerData. The PreferredCustomer class should have the following member variables:

**•** purchasesAmount (a double)

**•** discountLevel (a double)

The purchasesAmount variable holds the total of a customer’s purchases to date. The discountLevel variable should be set to the correct discount percentage, according to the store’s preferred customer plan. Write appropriate member functions for this class and demonstrate it in a simple program.

Demonstrate the class in a program that prompts the user to input the customer’s name, address, phone number, customer number, whether they want to receive mail, and the amount they've spent, and then uses that information to create a PreferredCustomer object and print its information.

***Input Validation: Do not accept negative values for any sales figures.***

|  |
| --- |
| **Problem 3: (Composition, accessor and mutator functions)** |

Write a class instructor with following data members and methods.

**Private Member Variables**

**Variable Description**

FirstName A char pointer that hold the first name of instructor

LastName A char pointer that hold the last name of instructor

OfficeNumber A char pointer that hold an office number of instructor

**Public Member Functions**

**Function Description**

**Default Constructor** Initialize all member variables with null value

**Overloaded Constructor** Accepts three constant char pointers and initialize firstName, lastName and officeNumber;

**Destructor** Show a message that I am a destructor from instructor class

**getData** It takes first, last and office number as input from the user

**setData** It takes three char pointer arguments and initialize instructor attributes

**showData** It display first, last and office number of an instructor

Write another class textBook with following data members and methods;

**Private Member Variables**

**Variable Description**

title A char pointer that hold the first name of instructor

author A char pointer that hold the last name of instructor

publisher A char pointer that hold an office number of instructor

**Public Member Functions**

**Function Description**

**Default Constructor** Initialize all member variables with null value

**Overloaded Constructor** Accepts three constant char pointers and initialize title, author and publisher

**Destructor** Show a message that I am a destructor from textBook class

**getData** It takes title, author and publisher name of book as input from the user

**setData** It takes three char pointer arguments and initialize textBook attributes

**showData** It display title, author and publisher name

Now design an another class course with following data members and methods;

**Private Member Variables**

**Variable Description**

CourseName A char array that hold the name of course

Instructor It hold the attributes of an instructor

TextBook It hold the attributes of textBook

**Public Member Functions**

**Function Description**

**Default Constructor** Initialize all seven members variables with null value

**Overloaded Constructor** Accepts seven constant char pointers and initialize all members with some values. It must call setData methods of instructor and textbook to initialize their own attributes. It only initialize courseName.

**Destructor** Show a message that I am a destructor from course class

**getData** It takes courseName as an input and call getData of instructor and textbook class for the input of their own attribute.

**showData** It display courseName, author and call showData of instructor and textbook class to display their own attribute.

Now in main function create two objects of course class (obj1, obj2) and each call different constructors. Now create obj3, it will call getData and take input from the user for all fields.

Now call the showData function for all three objects. Destructors also be called when objects will go out of scope.

|  |
| --- |
| **Problem 4: (Composition, accessor and mutator functions)** |

Write a class student with following data members and methods;

**Private Member Variables**

**Variable Description**

stdID A string that hold the ID of student

stdName A string that hold the name of student

**Public Member Functions**

**Function Description**

**Default Constructor** Initialize all member variables with null value

**Overloaded Constructor1** Initialize the two member variables with the following data

stdID = 14F-8153

stdName = Muhammad Noman

**getData** Get input from user using **this pointer** for all members

**showData** **Const** method that display data using **this pointer**

Write a class department with following data members and methods;

**Private Member Variables**

**Variable Description**

Dept A string that hold the department name of student

**Public Member Functions**

**Function Description**

**Default Constructor** Initialize with null value

**Overloaded Constructor** Initialize the three member variables with the following data.

dept = Computer Science

**getData** Get input from user using **this pointer**

**showData** **const** method that display data using **this pointer**

Now design an another class course with following data members and methods;

**Private Member Variables**

**Variable Description**

courseId A char pointer that hold the name of course

student It hold the attributes of an student

department It hold the attributes of department

**Public Member Functions**

**Function Description**

**Default Constructor** Initialize all seven member variables with null value

**Overloaded Constructor** Accepts seven constant char pointers and initialize all members with some values. It must call setData methods of instructor and textbook to initialize their own attributes. It only initialize courseName.

**Destructor** Show a message that I am a destructor from course class

**getData** It takes courseId as an input and call getData of student and department class for the input of their own attribute.

**showData** It display courseName, author and call showData of student and department class to display their own attribute.

Now in main function create two objects of course class (obj1, obj2) and each call different constructors. Now create obj3, it will call getData and take input from the user for all fields.

Now call the showData function for all three objects. Destructors also be called when objects will go out of scope.

**Good Luck! ☺**

**You are done with your exercise(s), attached all files and make your submission in slate [MS-word + program files].**