OOP Homework 2

Question No 1

You are a programmer for the Home Software Company. You have been assigned to develop a class that models the basic workings of a bank account. The class should perform the following tasks:

- Save the account balance.
- Save the number of transactions performed on the account.
- Allow deposits to be made to the account.
- Allow withdrawals to be taken from the account.
- Calculate interest for the period.
- Report the current account balance at any time.
- Report the current number of transactions at any time.
- Use this pointer.

Private Member Variables

Variable	Description
balance	A double that holds the current account balance.
interestRate	A double that holds the interest rate for the period.
interest	A double that holds the interest earned for the current period.
transactions	An integer that holds the current number of transactions.

Public Member Functions

Function	Description
Constructor	Takes arguments to be initially stored in the balance and interestRate members. The default value for the balance is zero and the default value for the interest rate is 4.5%.
setInterestRate	Takes a double argument which is stored in the interestRate member.
makeDeposit	Takes a double argument, which is the amount of the deposit. This argument is added to balance.

withdraw	Takes a double argument which is the amount of the withdrawal. This value is subtracted from the balance, unless the withdrawal amount is greater than the balance. If this happens, the function reports an error.
calcInterest	Takes no arguments. This function calculates the amount of interest for the current period, stores this value in the interest member, and then adds it to the balance member.
incCount	increment the value of count
getCount	return the value of count
getInterestRate	Returns the current interest rate (stored in the interestRate member).
getBalance	Returns the current balance (stored in the balance member).
getInterest	Returns the interest earned for the current period (stored in the interest member).
getTransactions	Returns the number of transactions for the current period (stored in the transactions member).

Hint: (To understand *this pointer* read this article: https://www.geeksforgeeks.org/this-pointer-in-c/)

Question No 2

(Account Inheritance Hierarchy) Create an inheritance hierarchy that a bank might use to represent customers' bank accounts. All customers at this bank can deposit (i.e., credit) money into their accounts and withdraw (i.e., debit) money from their accounts. Morespecific types of accounts also exist. Savings accounts, for instance, earn interest on the money they hold. Checking accounts, on the other hand, charge a fee per transaction (i.e., credit or debit).

Create an inheritance hierarchy containing base class Account and derived classes `Savings_Account` and `CheckingAccount` that inherit from class Account. Base class Account should include one data member of type double to represent the account balance. The class should provide a constructor that receives an initial balance and uses it to initialize the data member. The constructor should validate the initial balance to ensure that it's greater than or equal to 0.0. If not, the balance should be set to 0.0 and the constructor should display an error message, indicating that the initial balance was invalid. The class should provide three member functions. Member function credit should add an amount to the current balance. Member function debit should withdraw money from the Account and ensure that the debit amount does not exceed the Account's balance. If it does, the balance should be left unchanged and the function should print the message "Debit amount exceeded account balance." Member function getBalance should return the current balance.

Derived class SavingsAccount should inherit the functionality of an Account, but also include a data member of type double indicating the interest rate (percentage) assigned to the Account. SavingsAccount's constructor should receive the initial balance, as well as an initial value for the SavingsAccount's interest rate. SavingsAccount should provide a public member function calculateInterest that returns a double indicating the amount of interest earned by an account. Member function calculateInterest should determine this amount by multiplying the interest rate by the account balance. [Note: SavingsAccount should inherit member functions credit and debit as is without redefining them.

Derived class CheckingAccount should inherit from base class Account and include an additional data member of type double that represents the fee charged per transaction. Checking_Account's constructor should receive the initial balance, as well as a parameter indicating a fee amount. Class CheckingAccount should redefine member functions credit and debit so that they subtract the fee from the account balance whenever either transaction is performed successfully. CheckingAccount's versions of these functions should invoke the base-class Account version to perform the updates to an account balance. CheckingAccount's debit function should charge a fee only if money is actually withdrawn (i.e., the debit amount does not exceed the account balance).

[**Hint:** Define Account's debit function so that it returns a bool indicating whether money was withdrawn. Then use the return value to determine whether a fee should be charged.]

After defining the classes in this hierarchy, write a program that creates objects of each class and tests their member functions. Add interest to the SavingsAccount object by first invoking its calculateInterest function, then passing the returned interest amount to the object's credit function.

Question No 3

(Package Inheritance Hierarchy) Package-delivery services, such as FedEx ®, DHL® and UPS®, offer a number of different shipping options, each with specific costs associated. Create an inheritance hierarchy to represent various types of packages. Use class Package as the base class of the hierarchy, then include classes TwoDayPackage and OvernightPackage that derive from Package. Base class Package should include data members representing the name, address, city, state and ZIP code for both the sender and the recipient of the package, in addition to data members that store the weight (in ounces) and cost per ounce to ship the package.

Package's constructor should initialize these data members. Ensure that the weight and cost per ounce contain positive values. Package should provide a public member function calculateCost that returns a double indicating the cost associated with shipping the package. Package's

calculateCost function should determine the cost by multiplying the weight by the cost per ounce.

Derived class TwoDayPackage should inherit the functionality of base class Package, but also include a data member that represents a flat fee that the shipping company charges for two-day-delivery service. TwoDayPackage's constructor should receive a value to initialize this data member. TwoDayPackage should redefine member function *calculateCost* so that it computes the shipping cost by adding the flat fee to the weight-based cost calculated by base class Package's calculateCost function.

Class OvernightPackage should inherit directly from class Package and contain an additional data member representing an additional fee per ounce charged for overnight-delivery service. OvernightPackage should redefine member function calcu- Exercises 533 lateCost so that it adds the additional fee per ounce to the standard cost per ounce before calculating the shipping cost. Write a test program that creates objects of each type of Package and tests member function calculateCost.

Question No 4

Write a program that calculates the area and keeps a count that how many times it has calculated the area.

1. Create a class area which has data members of length, width and area (int type).
2. It also has a static int member count, initialize it with zero.
3. Overload constructor of class and calculate area in it, pass the values of length and width
4. from main.(calculate the area three times).
5. Create a static member function in class which tells that how many times area has been
6. calculated. (This function will return the value of count when called)
7. Write function of display and display the area along with the value of count.
8. Use this pointer.

Hint: (To understand *this pointer* read this article: https://www.geeksforgeeks.org/this-pointer-in-c/)

Question No 5

1. Define a class matrix with data members 2d array, row , col
2. Write a default constructor to initialize the data members to the following values:
   a) Row =2
   b) Col=3
   c) Allocate memory

d) Initialize with 1 at each index

3. Write overloaded constructor that will take input row, col and allocate memory accordingly.
4. Write public function to inputData for matrix.
5. Create a dynamic array of matrix of size 4.
6. Assign memory at 0 index with default constructor
7. Assign memory at 1st index with overloaded constructor of same size
8. Write function for addMatrix in class which will add matrix2 with this and return the result matrix. E.g
   Matrix AddMatrix(Matrix matrix2)
9. Save result matrix at 3rd index
10. Call copy constructor which will copy matrix[3] = matrix[2];
11. Free memory by calling destructor.
12. Show proper message for call of constructor, copy constructor and destructor.
13. Use this pointer.

Hint: (To understand *this pointer* read this article: https://www.geeksforgeeks.org/this-pointer-in-c/)

Question No 6

Write a program which can detect the sounds of Animals.

Write a class Animals having the following attributes:

1. String Name
2. Sound()

Inherit the following classes from Animals having the same attributes and behavior of base class

Override the sound function for each class

1. Cat
2. Dog
3. Tiger_Family
4. Deer

From Tiger_family class inherit the following sub classes publically: (override the sound function for each class).

1. Tiger
2. Lion **Note:**

- Make instance of each class and call sound function.
- Each class animal should have its own sound.