Object Oriented Programming

Lab 5 (Manual)

Instructions:

- You are not allowed to use cin/cout inside setters and getters.
- You are not allowed to use string library.
- Getters and display/prints functions can only be of constant type.
- You must avoid memory leakage

Task 1

Create class SavingsAccount. Use a variable annualInterestRate to store the annual interest rate for all account holders. Each object of the class contains a private instance variable savingsBalance indicating the amount the saver currently has ondeposit. Provide method calculateMonthlyInterest to calculate the monthly interest by multiplying the savingsBalance by annualInterestRate divided by 12 this interest should be added to savingsBalance. Provide a method modifyInterestRate that sets the annualInterestRate to a new value. Write a program to test class SavingsAccount. Instantiate two savingsAccount objects, saver1 and saver2, with balances of $2000.00 and $3000.00, respectively. Set annualInterestRate to 4%, then calculate the monthly
interest and print the new balances for both savers. Then set the annualInterestRate to 5%, calculate the next month’s interest and print the new balances for both savers.

```
int main()
{
SavingsAccount saver1;
SavingsAccount saver2;
saver1.setSavingsBalance(2000.00);
saver2.setSavingsBalance(3000.00);
SavingsAccount::modifyInterestRate(0.04);
saver1.calculateMonthlyInterest();
saver2.calculateMonthlyInterest();
cout << "New Balance for Saver1=%f\n" <<
saver1.getSavingsBalance();
cout << "New Balance for Saver2=%f\n" <<
saver2.getSavingsBalance();
SavingsAccount::modifyInterestRate(0.05);
saver1.calculateMonthlyInterest();
saver2.calculateMonthlyInterest();
cout << "New Balance for Saver1=%f\n" <<
saver1.getSavingsBalance();
cout << "New Balance for Saver2=%f\n" <<
saver2.getSavingsBalance(); return
0;
}
```

Task 2

Create Time class making use of Copy constructor:

In this lab you will write a class called Time. The Time class will have three data members: hours, minutes, and seconds (all ints). Times will be stored using a 24 hour clock, so 10 AM is stored as 10 and 10 PM is stored as 22.

Your Time class should have the following member functions:

default constructor

This function initializes the Time object to contain 12 midnight (hour zero, minutezero, seconds zero).

constructor with three parms

This function has parms for the hour, minutes, and secconds. It initializes the Time object to the values given by the parms.

copy constructor

This function initializes the Time object to contain the same values as the parm.

setTime

This function has parms for the hour, minute, and second, and returns void.setTime sets the time to the values given by the parms.

showTime

This function has no parms and returns void. It prints the time using the normal 12 hour clock and AM/PM designation. For example, if hours is 21, minutes is 32 and seconds is 5, the time should be displayed as 9:32:05 PM. If hours is 5, minutes is 12, and seconds is 45, the time should be displayed as 5:12:45 AM. Don't forget to add a leading zero if the minutes and/or the seconds contains a value less than 10!

tick

This function has no parms and returns void. It increments the time by onesecond, adjusting the minutes and hours if necessary.

Write client code to test your Time class. Your main function should:

1. Create two Time objects and prompt the user to enter values for the Time objects.
2. Print the Time objects.
3. Print a message that tells which Time object contains a shorter distance.
4. Tick the first Time object 10 times, then print the first Time object.
5. Create a third Time object that is initialized by the default constructor.
6. Create a fourth Time object that is initialized to 10:56:05AM.
7. Create a fifth Time object that is initialized to be a copy of the fourth Time object.

Print the third, fourth, and fifth Time objects.

Task 3

Create class for items in a general store. The items will have the following characteristics.

- Item_code (int)
- Item_name (char *)
- Price; (int)
- Expiration_date:
- Current_date:

Calculate the amount of days remaining till the expiration date. If the days remaining are less than 20 then the price of item should be reduced by 40%.
Along with parameterized constructor, the setter and getter functions you have to create a copy constructor(Deep copy).
In main function, create an object1 and initialize it with parameterized constructor. Create object2 and initlize it with object1. Change item name of object2 and print data of both objects.

Task 4

Next Bridge software house Lahore requires an application that calculates employee's increment based on the given criteria. For the purpose software house wants his software engineer to design the application using object oriented approach. The class"NextBridge" is required to create with attributes and functions as mentioned below.

- name:char*
- address:char*
- phoneNumber: char*
- level: char*   //fresh,Junior,sensior
- experience:int   //in years
- currentSalary: double
- increment: double

// Constructors
 	
// Destructor

```
// Get Functions

// Set Function

// Calculate increment

// Display Function (parameterized function)
```

Increment criteria	
Experience	**Increment**
0	0%
01- 02	10 %
02- 06	20%
06 to onward	40%

- When an object of the class is instantiated, it is always done by using an overloaded constructor having 7 parameters. Examples of such instantiations are given below: NextBridge obj("Name","Address","Ph:0300-123","fresh",2,10000,0);
- Create a copy constructor. Copy data of Object1 to object 2 using copy constructor. Write a test program that demonstrates capabilities of the class.
- Use demo main function to illustrate everything

Task 5:

Design and implement your own string library(string.h). This library should include at least 7 most commonly used functions of built in string library.