CL-1004
Object Oriented Programming
Lab No 7

Objectives:

- Operator Overloading
- This pointer
- Constant data members
- Static data members

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 by students; the block should contain brief description about functionality of code.
2. Comment on every function and about its functionality.
3. Mention comments where necessary such as comments with variables, loop, classes etc to increase code understandability.
4. Use understandable name of variables.
5. Proper indentation of code is essential.
6. Write a code in C++ language.
7. Make a Microsoft Word file and paste all of your C++ code with all possible screenshots of every task **output in Microsoft Word and submit word file. Do not submit .cpp file.**
8. First think about statement problems and then write/draw your logic on copy.
9. After copy pencil work, code the problem statement on MS Studio C++ compiler.
10. At the end when you done your tasks, attached C++ created files in MS word file and make your submission on Google Classroom. (Make sure your submission is completed).
11. Please submit your file in this format **20F1234_L1**.
12. Do not submit your assignment after deadline. Late submission is not accepted.
13. Do not copy code from any source otherwise you will be penalized with negative marks.

Problem 1: | (Classes, objects, Constructor, Destructor and Member functions, constant)

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.

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.
count	A static integer that hold the total number of time the program has taken choice

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).

Note: *All **get** methods must be constant. Output must in the same format as given on next page.*

```
 MENU
-----------------------------------------
A) Display the account balance
B) Display the number of transactions
C) Display interest earned for this period
D) Make a deposit
E) Make a withdrawal
F) Add interest for this period
G) Exit the program

Number of times program has teken choice: 0
Enter your choice: d
Enter the amount of the deposit: 6500

 MENU
-----------------------------------------
A) Display the account balance
B) Display the number of transactions
C) Display interest earned for this period
D) Make a deposit
E) Make a withdrawal
F) Add interest for this period
G) Exit the program

Number of times program has teken choice: 1
```

```
Enter your choice: f
Interest added.

 MENU
----------------------------------------
A) Display the account balance
B) Display the number of transactions
C) Display interest earned for this period
D) Make a deposit
E) Make a withdrawal
F) Add interest for this period
G) Exit the program

Number of times program has teken choice: 2
Enter your choice: a
The current balance is $6792.50

 MENU
----------------------------------------
A) Display the account balance
B) Display the number of transactions
C) Display interest earned for this period
D) Make a deposit
E) Make a withdrawal
F) Add interest for this period
G) Exit the program

Number of times program has teken choice: 3
Enter your choice: c
Interest earned for this period: $292.50

 MENU
----------------------------------------
A) Display the account balance
B) Display the number of transactions
C) Display interest earned for this period
D) Make a deposit
E) Make a withdrawal
F) Add interest for this period
G) Exit the program

Number of times program has teken choice: 4
Enter your choice: e
Enter the amount of the withdrawal: 6985
ERROR: Withdrawal amount too large.


 MENU
----------------------------------------
A) Display the account balance
B) Display the number of transactions
C) Display interest earned for this period
D) Make a deposit
E) Make a withdrawal
F) Add interest for this period
G) Exit the program

Number of times program has teken choice: 5
Enter your choice: b
There have been 1 transactions.
```

Use 3 file structure.

Problem 2: Operator Overloading

Write a class Employee having following attributes:

1. String name
2. Integer Age
3. Float Salary

Overload the appropriate operators for the following functionality:

1. Adding two employee objects (concatenating the names of both objects, add the rest of the two data members)
2. Telling which employee is elder (overload – operator for this)
3. Comparing the salary of two employees
4. Input employee object
5. Output employee object

Problem 3: Operator Overloading

Write a class **Factorial** to overload ! operator to find factorial of an integer object. Write a driver program to test your class. Show input and output results on console.

Problem 4: Operator Overloading

Define a class **Matrix** to represent rows × cols matrix. r (row) and c (column) will be passed as parameters to constructor of class Matrix:

1. Overload operators for addition (use "+" operator for addition) and subtraction (use "-" operator for subtraction) of two matrices.

2. Overload operators for pre-increment (use "++" operator. This operator will increment (all elements of matrix by) 1.

3. Overload operators for post-increment (use "++" operator. This operator will increment (all elements of matrix by) 1.

4. Overload operators for pre-decrement (use "--" operator. This operator will decrement (all elements of matrix by) 1.

5. Overload operators for post-decrement (use "--" operator. This operator will decrement

(all elements of matrix by) 1.

6. Overload insertion ">>" to input all elements of matrix.
7. Overload extraction "<<" operator to output all elements on console.
8. Overload less than operator "<" for two matrices. This operator will return true if the sum of all elements of matrix is less than second. i.e. A < B.
9. Overload less than operator ">=" for two matrices. This operator will return true if all elements of matrix A is greater than or equal to second. i.e. A >= B. If any one element is smaller than B at same location, it will return false.
10. Overload unary operators "*" that will return the product of all elements of a matrix.

Note: Size of matrix A will be same as size of B for binary operators.

Write a driver program to test your class.

Problem 5: Operator Overloading

Write a class Complex for complex numbers having the following data members:

1. Float a
2. Float b

Write overloaded and default constructors for your class.

Implement the following functionality for your class.

float mag( );	It will compute and return the magnitude of a complex number. The magnitude of a complex number $a + bi$ is the quantity $\sqrt{a^2 + b^2}$.
Complex add(Complex c);	The method accepts a complex number c, adds it with *this* complex number and returns the answer as another complex number. The addition of two complex numbers, $a + bi$ and $c + di$ is defined as follows: $(a + bi) + (c + di) = (a + c) + (b + d)i$
Complex mul(Complex c);	The method accepts a complex number c, multiplies it with *this* complex number and returns the answer as another complex number. The multiplication of two complex numbers, $a + bi$ and $c + di$ is defined as follows: $(a + bi) * (c + di) = (ac - bd) + (ad + bc)i$
Complex operator++;	Overload pre-increment operator
Complex operator++(int);	Overload post-increment operator
Complex operator--;	Overload pre-decrement operator
Complex operator--(int);	Overload post-decrement operator
ostream& operator<<(ostream& os, const Complex& c);	Overload extraction operator so that it can display *this* complex number, in the format: **a+bi**

Proper code indentation will hold extra marks !

Best of luck ☺

You are done with your exercise, submit on classroom at given time.