

# CL-1004

## Object Oriented Programming

### Lab No 7

#### Objectives:

- Classes and objects
- Data members
- Member function
- Data encapsulation
- Member access specifier (private , public)
- Constructor, Destructor

**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 about its functionality.
3. Use understandable name of variables.
4. Proper indentation of code is essential
5. Write a C++ statement(s) for each of the following task one after the other, in the same order.
6. Make a Microsoft Word file and paste all of your C++ code with all possible screenshots of every **task output in MS word and submit .cpp file with word file.**
7. Make separate .cpp files for all tasks and use this format **22F-1234\_Task1.cpp.**
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 **22F-1234\_L1.**
12. Do not submit your assignment **after deadline.**
- 13. Do not copy code from any source otherwise you will be penalized with negative marks.**

### Problem : | (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
<b>Constructor</b>	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%.
<b>setInterestRate</b>	Takes a double argument which is stored in the interestRate member.
<b>makeDeposit</b>	Takes a double argument, which is the amount of the deposit. This argument is added to balance.
<b>withdraw</b>	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.
<b>calcInterest</b>	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.
<b>incCount</b>	increment the value of count
<b>getCount</b>	return the value of count
<b>getInterestRate</b>	Returns the current interest rate (stored in the interestRate member).
<b>getBalance</b>	Returns the current balance (stored in the balance member).
<b>getInterest</b>	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.