COURSE: (CL-1004) OBJECT ORIENTED PROGRAMMING LAB
LAB TASK # 05

NOTE:
Only submit .cpp file of each question in a folder. Anyone who submits any other format file will get straight **ZERO.** Each question should have a separate .cpp file. Copy Paste or other UFM will also get **ZERO**. Use the following format for naming the folder Roll#_Name (P18-1234_NAME).

Q No.1: Create a class that includes a data member that holds a "serial number" for each object created from the class. That is, the first object created will be numbered as 2022FAST0100P, the second 2022FAST0200P, and so on.

To do this, you'll need another data member that records a count of how many objects have been created so far. (This member should apply to the class as a whole; not to individual objects. What keyword specifies this?) Then, as each object is created, its constructor can examine this count member variable to determine the appropriate serial number for the new object.

Add a member function that permits an object to report its own serial number. Then write a main() program that creates five objects and queries each one about its serial number.

The sample output is as follow:

I'm object with Serial Number: 2022FAST0100P
I'm object with Serial Number: 2022FAST0200P
I'm object with Serial Number: 2022FAST0300P

Q No.2: Create a **SavingsAccount** class. Use a static data member **annualInterestRate** to store the annual interest rate for each of the savers. Each member of the class contains a private data member **savingsBalance** indicating the amount the saver currently has on deposit.

Provide member function **calculateMonthlyInterest** that calculates the monthly interest by multiplying the balance by annualInterestRate dividedby12; this interest should be added to **savingsBalance.**

Provide a member function to **display** the balance of the saver.

Write a program to test class **SavingsAccount.** Instantiate two different objects of class SavingsAccount, **saver1** and **saver2**, with balances of **$2000.00 and $3000.00,** respectively.

Set the **annualInterestRate to 3 percent**. Then calculate the monthly interest and print the new balances for each of the savers.

Variations: Provide a static member function **modifyInterestRate** that sets the **static annualInterestRate** to a new value. Set the **annualInterestRate to 4 percent**, calculate the next month's interest and print the new balances for each of the savers.

Q N0.3: Create a class name **student.** Use a constant variable named **institute** to store the institute name of the student and other variable to store the basic information about student like name, GPA, marks[ ], and email etc. Now create a parameterized constructor to set the values of student class variables. Create **claculateCGPA** function that will calculate the average of the marks and assign the appropriate CGPA to student. Also create a constant display function to show all the student details. In main create an array of 3 students and set their values and CGPA and then display the details. All the data members should be private.

Note: For user understanding purposes you should write comment with each line of code.