**Object Oriented Programming**

**Lab Report**

**Lab06**



|  |  |
| --- | --- |
| Group Members Name & Reg #: | **Muhammad Haris Irfan**  **(FA18-BCE-090)** |
|  |  |
| Class | Object Oriented Programming CSC241 (**BCE-4B**) |
| Instructor’s Name | Maam Amber Madeeha Zeb |

**In Lab Tasks**

**5.1 Task 1:**

**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 divided by 12; this interest should be added to savingsBalance. Provide a static member function modifyInterestRate that sets the static annualInterestRate to a new value. Write a driver 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. Then set the annualInterestRate to 4 percent, calculate the next month's interest and print the new balances for each of the savers.**

**Solution:**

The code is given below,

1 #include <iostream>

2

3 **using namespace std**;

4

5 **class** SavingsAccount

6 {

7 **private**:

8 **static float** annualInterestRAte;

9 **float** savingBalance;

10

11 **public**:

12 SavingsAccount()

13 {

14

15 }

16 SavingsAccount(**float** x)

17 {

18 savingBalance=x;

19 }

20 **void** calculateMonthly()

21 {

22 **float** monthlyint;

23 monthlyint=savingBalance\*(annualInterestRAte/12);

24 savingBalance=savingBalance+monthlyint;

25 **cout**<<"The Interest rate is: "<<annualInterestRAte<<**endl**;

26 }

27

28 **void** printbalance()

29 {

30 **cout**<<"The balance is: "<<savingBalance<<**endl**;

31 }

32 **static void** ModifyInterestRate(**float** x)

33 {

34 annualInterestRAte=x;

35 }

36 };

37 **float** SavingsAccount::annualInterestRAte=3;

38 **int** main()

39 {

40 SavingsAccount saver1(2000);

41 SavingsAccount saver2(3000);

42

43 //saver2.ModifyInterestRate(3);

44

45

46

47 saver1.printbalance();

48 saver1.calculateMonthly();

49 saver1.printbalance();

50 **cout**<<**endl**;

51 saver2.printbalance();

52 saver2.calculateMonthly();

53 saver2.printbalance();

54

55 saver2.ModifyInterestRate(4);

56 **cout**<<**endl**;

57 saver1.printbalance();

58 saver1.calculateMonthly();

59 saver1.printbalance();

60 **cout**<<**endl**;

61 saver2.printbalance();

62 saver2.calculateMonthly();

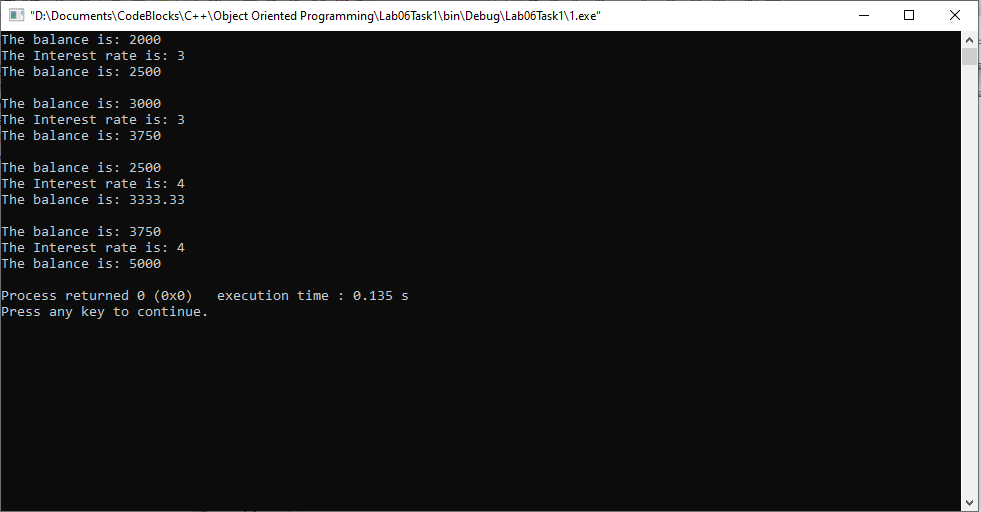
63 saver2.printbalance();

64

65 **return** 0;

66 }

**Console Output is shown below.**

****

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

POST LAB

6.1 Question 1:

Write C++ program to count the number of objects created and destroyed for a class using static data members and static member functions

Solution:

I am attaching my code below,

1 #include <iostream>

2 **using namespace std**;

3 **int** countv=1;

4 **int** countv1=1;

5 **class** mainClass

6 {

7 **private**:

8 **int** value;

9 **int** value1;

10

11 **public**:

12 mainClass()

13 {

14 value=countv++;

15 **cout**<<"Creating Object no: "<<value<<**endl**;

16 }

17 ~mainClass()

18 {

19 value1=countv1++;

20 **cout**<<"Destroying Object no: "<<value1<<**endl**;

21 }

22

23 };

24

25

26 **int** main()

27 {

28 mainClass ob1;

29 mainClass ob2;

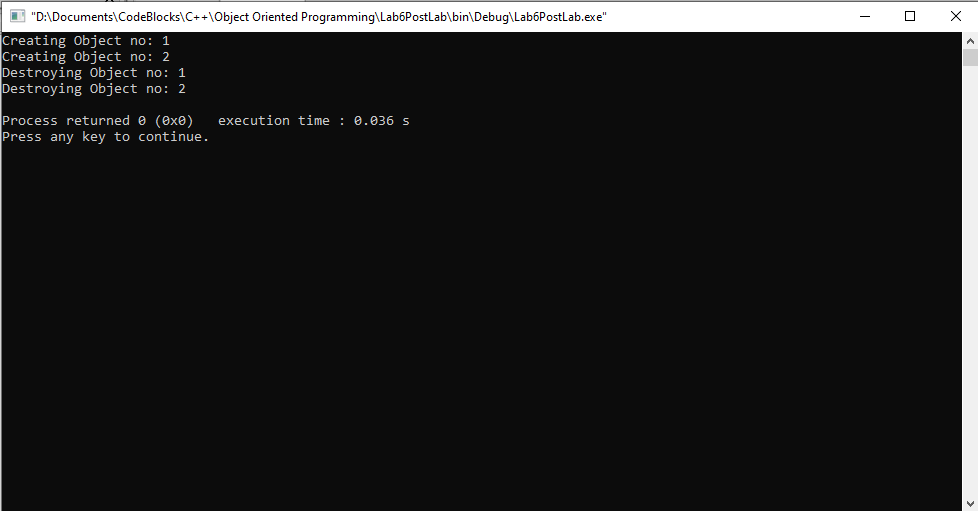
30

31

32 **return** 0;

33 }

The result for this program is shown below,



\_\_\_\_\_\_THE END\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_