**Object Oriented Programming**

**Lab 6**

**Submitted To:**

Ma’am Amber Madeeha Zeb

**Submitted By:**

Manaal Waseem

FA18-BCE-074

**In Lab:**

**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.**

**Code:**

1 #include <iostream>

2

3 **using namespace std**;

4

5 **class** SavingAccount

6 {

7 **private**:

8 **static double** annualInterestRate;

9 **double** savingsBalance;

10

11 **public**:

12

13 SavingAccount()

14 {

15 savingsBalance=0;

16 }

17

18 SavingAccount(**double** u)

19 {

20 savingsBalance=u;

21 }

22

23 **void** calculateMonthlyInterest()

24 {

25 **cout** << "\nCalculating MonthlyInterest ...";

26 savingsBalance +=(annualInterestRate\*savingsBalance)/12;

27 **cout** << "\nMonthly Interest is calculated successsfully" << **endl**;

28 }

29

30 **static void** modifyInterestRate(**double** p)

31 {

32 annualInterestRate=p;

33 }

34 **void** display()

35 {

36 **cout** << savingsBalance<< **endl** << **endl**;

37 }

38 **const double** getAnnualInterestRate()

39 {

40 **return** annualInterestRate;

41 }

42 };

43 **double** SavingAccount:: annualInterestRate=0;

44

45 **int** main()

46 {

47 SavingAccount saver1(2000.00), saver2(3000.00);

48

49 **cout** << "\nSaving Balance of saver1 is: $ ";

50 saver1.display();

51

52 **cout** << "\nSaving Balance of saver2 is: $ ";

53 saver2.display();

54

55 saver1.modifyInterestRate(3);

56 **cout** << "\nAnnual Interest Rate has been modified. The new rate is: " << saver1.getAnnualInterestRate() << **endl**;

57

58 saver1.calculateMonthlyInterest();

59 **cout** << "\nThe new Balance of saver1 is: $ ";

60 saver1.display();

61

62 saver2.calculateMonthlyInterest();

63 **cout** << "\nThe new Balance of saver2 is: $ ";

64 saver2.display();

65

66 saver2.modifyInterestRate(4);

67 **cout** << "\nAnnual Interest Rate has been modified. The new rate is: " << saver1.getAnnualInterestRate() << **endl**;

68 saver1.calculateMonthlyInterest();

69 **cout** << "\nThe new Balance of saver1 is: $ ";

70 saver1.display();

71

72 saver2.calculateMonthlyInterest();

73 **cout** << "\nThe new Balance of saver2 is: $ ";

74 saver2.display();

75

76 **return** 0;

77 }

**Output:**



**Post Lab:**

**Task 1:**

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

**Code:**

1 #include <iostream>

2

3 **using namespace std**;

4

5 **class** counter

6 {

7 **private**:

8 **static int** count\_created, count\_destroyed;

9 **public**:

10

11 counter()

12 {

13 count\_created++;

14 **cout** << "Number of objects created= " << count\_created << **endl**;

15 };

16

17 ~counter()

18 {

19 count\_destroyed++;

20 **cout** << "Number of objects destroyed= " << count\_destroyed << **endl**;

21 };

22 };

23

24 **int** counter::count\_created=0, counter::count\_destroyed=0;

25

26 **int** main()

27 {

28 counter obj1;

29

30 **for**(**int** i=2; i<=5; i++)

31 {

32 counter obj;

33 **cout** << "Iteration no. " << i << **endl**;

34 }

35

36 **return** 0;

37 }

**Output:**

****

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**THE END**