Create class SavingsAccount. Use a static variable annualInterestRate to store the annual interest rate for all account holders. Each object of the class contains a private instance variable savingsBalance indicating the amount the saver currently has on deposit. Provide method calculateMonthlyInterest() to calculate the monthly interest by multiplying the savingsBalance by annualInterestRate divided by 12.This interest should be added tosavingsBalance. Provide a static method modifyInterestRate() that sets the annualInterestRate to a new value. Write a program to test class SavingsAccount. Instantiate two savingsAccount objects, saver1 and saver2, with balances of Rs2000.00 and Rs3000.00, respectively. Set annualInterestRate to 4%, then calculate the monthly interest and print the new balances for both savers. Then set the annualInterestRate to 5%, calculate the next month’s interest and print the new balances for both savers

SOURCE CODE:

/\*Create class SavingsAccount. Use a static variable annualInterestRate to store the

annual interest rate for all account holders. Each object of the class contains a private

instance variable savingsBalance indicating the amount the saver currently has on

deposit. Provide method calculateMonthlyInterest() to calculate the monthly

interest by multiplying the savingsBalance by annualInterestRate divided by

12.This interest should be added tosavingsBalance. Provide a static method

modifyInterestRate() that sets the annualInterestRate to a new value. Write a

program to test class SavingsAccount. Instantiate two savingsAccount objects,

saver1 and saver2, with balances of Rs2000.00 and Rs3000.00, respectively. Set

annualInterestRate to 4%, then calculate the monthly interest and print the new

balances for both savers. Then set the annualInterestRate to 5%, calculate the next

month’s interest and print the new balances for both savers\*/

#include<iostream>

using namespace std;

class savingaccounts

{

    private:

    float savingbalance;

    public:

    static int ar;

    void setsavingbalance(float savingbalance)

    {

        this->savingbalance=savingbalance;

    }

    static void modifyrate(int ar)

    {

        ar=ar;

    }

    int calc()

    {

        savingbalance=savingbalance+(savingbalance\*ar)/1200;

    }

    void display()

    {

        cout<<savingbalance<<endl;

    }

};

int savingaccounts::ar=4;

int main()

{

    savingaccounts d1,d2;

    d1.setsavingbalance(2000);

    d2.setsavingbalance(3000);

    d1.calc();

    d2.calc();

    d1.display();

    d2.display();

    savingaccounts::modifyrate(5);

    d1.calc();

    d2.calc();

    d1.display();

    d2.display();

}

OUTPUT: