```cpp
//
// Shaun Chemplavil U08713628
// shaun.chemplavil@gmail.com
// C/C++ Programming III : Intermediate Programming with Objects
// 151116  Raymond L.Mitchell III
// SavingsAccount.h
// Win10
// Visual C++ 19.0
//
// File contains the SavingsAccount class definition
//

#ifndef SHAUNCHEMPLAVIL_SAVINGSACCOUNT_H
#define SHAUNCHEMPLAVIL_SAVINGSACCOUNT_H

#include <iostream>
using std::cerr;

namespace ShaunChemplavil
{
   class SavingsAccount
   {
   public:

      SavingsAccount(double initialBalance);

      double getSavingsBalance() const;
      // static member function to apply to ALL SavingsAccount objects
      static void  setAnnualInterestRate(double newInterestRate);
      void  applyMonthlyInterest();

   private:
      double savingsBalance;
      static double annualInterestRate;
   };
}

#endif

```

```
//
// Shaun Chemplavil U08713628
// shaun.chemplavil@gmail.com
// C/C++ Programming III : Intermediate Programming with Objects
// 151116  Raymond L.Mitchell III
// SavingsAccount.cpp
// Win10
// Visual C++ 19.0
//
// File contains the display member function for the SavingsAccount class
//

#include <iostream>
#include "SavingsAccount.h"
using std::cerr;

// Static class variable
double ShaunChemplavil::SavingsAccount::annualInterestRate;

// Constructor
ShaunChemplavil::SavingsAccount::SavingsAccount(double initialBalance)
{
   if (initialBalance < 0)
   {
      cerr << "\nERROR: INVALID INITIAL BALANCE!\n";
      this->savingsBalance = 0.0;
   }
   else
      this->savingsBalance = initialBalance;
}
// Member function returning the Account Balance
double ShaunChemplavil::SavingsAccount::getSavingsBalance() const
{
   return this->savingsBalance;
}

// Member function to apply the monthly interest rate
void ShaunChemplavil::SavingsAccount::applyMonthlyInterest()
{
   const double ANNUAL_TO_MONTHLY = 1.0 / 12.0;
   this->savingsBalance *= (1 + this->annualInterestRate * ANNUAL_TO_MONTHLY);
}

// Member function to set the annual interest rate for ALL SavingsAccount objs
void ShaunChemplavil::SavingsAccount::setAnnualInterestRate(double newInterestRate)
{
   const double INTEREST_RATE_TO_DEC = 0.01;
   if (newInterestRate < 0)
   {
      cerr << "\nERROR: INVALID INTEREST RATE!\n";
      ShaunChemplavil::SavingsAccount::annualInterestRate = 0.0;
```

```
    }
    else
        ShaunChemplavil::SavingsAccount::annualInterestRate = newInterestRate * INTEREST_RATE_TO_DEC;
}

```

```
//
// Shaun Chemplavil U08713628
// shaun.chemplavil@gmail.com
// C/C++ Programming III : Intermediate Programming with Objects
// 151116  Raymond L.Mitchell III
// hw3.cpp
// Win10
// Visual C++ 19.0
//
// Test Program for the SavingsAccount class
//

#include <iostream>
#include "SavingsAccount.h"
using std::cout;
using std::cerr;

int main()
{
   // declare original date variables (arbitrary values)
   double initialBalance0 = 1234.56, initialBalance1 = 1000.00,
      interestRate = 5.0;

   // exercise all of SavingsAccount's public functions
   ShaunChemplavil::SavingsAccount validAccount0(initialBalance0),
      validAccount1(initialBalance1), invalidAccount(-initialBalance0);

   // exercising all of the get functions
   cout << "Valid Account Balance 0 is "
      << validAccount0.getSavingsBalance() << "\n"
      << "Valid Account Balance 1 is "
      << validAccount1.getSavingsBalance() << "\n"
      << "Invalid Account Balance is "
      << invalidAccount.getSavingsBalance() << "\n\n";

   // set Invalid Annual Interest Rate
   cout << "Setting Savings Account interest rate to " << -interestRate << "%\n";
   ShaunChemplavil::SavingsAccount::setAnnualInterestRate(-interestRate);

   for (int month = 1; month <= 3; month++)
   {
      validAccount0.applyMonthlyInterest();
      validAccount1.applyMonthlyInterest();
      invalidAccount.applyMonthlyInterest();

      // exercising all of the get functions
      cout << "Valid Account Balance 0 " << "after " << month << " months is "
         << validAccount0.getSavingsBalance() << "\n"
         << "Valid Account Balance 1 " << "after " << month << " months is "
         << validAccount1.getSavingsBalance() << "\n"
         << "Invalid Account Balance " << "after " << month << " months is "
         << invalidAccount.getSavingsBalance() << "\n\n";
```

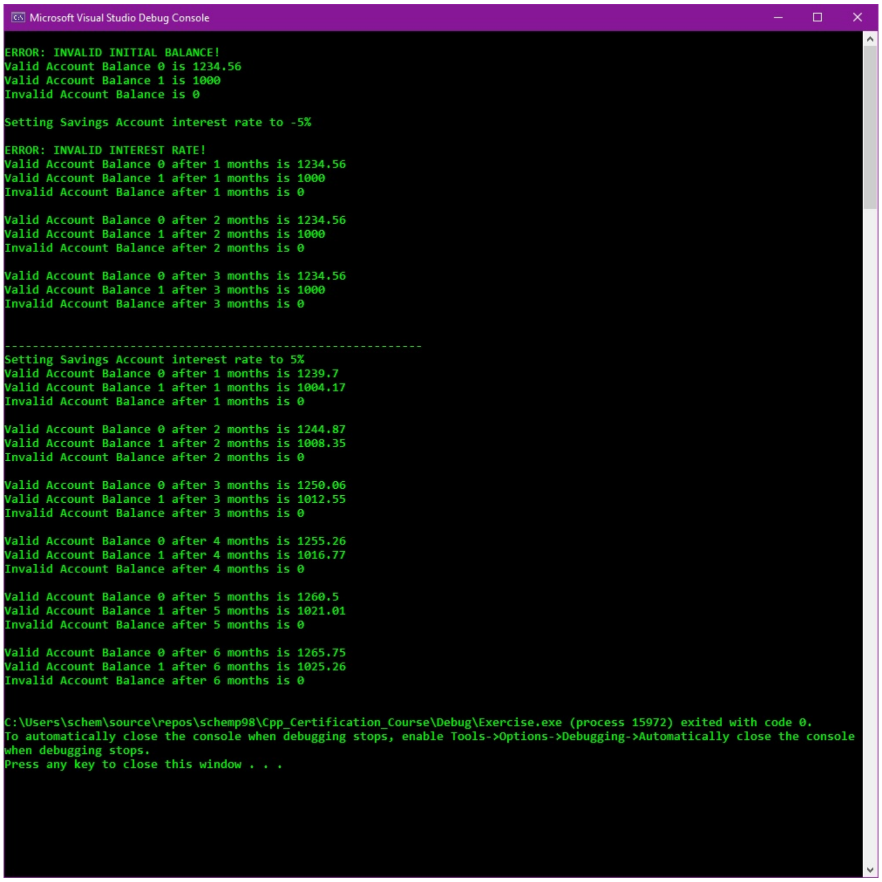
```cpp
   }

   // set Annual Interest Rate
   cout << "\n------------------------------------------------------------\n" <<
      "Setting Savings Account interest rate to " << interestRate << "%\n";
   ShaunChemplavil::SavingsAccount::setAnnualInterestRate(interestRate);

   for (int month = 1; month <= 6; month++)
   {
      validAccount0.applyMonthlyInterest();
      validAccount1.applyMonthlyInterest();
      invalidAccount.applyMonthlyInterest();

      // exercising all of the get functions
      cout << "Valid Account Balance 0 " << "after " << month << " months is "
         << validAccount0.getSavingsBalance() << "\n"
         << "Valid Account Balance 1 " << "after " << month << " months is "
         << validAccount1.getSavingsBalance() << "\n"
         << "Invalid Account Balance " << "after " << month << " months is "
         << invalidAccount.getSavingsBalance() << "\n\n";
   }
}

```



```
ERROR: INVALID INITIAL BALANCE!
Valid Account Balance 0 is 1234.56
Valid Account Balance 1 is 1000
Invalid Account Balance is 0

Setting Savings Account interest rate to -5%

ERROR: INVALID INTEREST RATE!
Valid Account Balance 0 after 1 months is 1234.56
Valid Account Balance 1 after 1 months is 1000
Invalid Account Balance after 1 months is 0

Valid Account Balance 0 after 2 months is 1234.56
Valid Account Balance 1 after 2 months is 1000
Invalid Account Balance after 2 months is 0

Valid Account Balance 0 after 3 months is 1234.56
Valid Account Balance 1 after 3 months is 1000
Invalid Account Balance after 3 months is 0


------------------------------------------------------------
Setting Savings Account interest rate to 5%
Valid Account Balance 0 after 1 months is 1239.7
Valid Account Balance 1 after 1 months is 1004.17
Invalid Account Balance after 1 months is 0

Valid Account Balance 0 after 2 months is 1244.87
Valid Account Balance 1 after 2 months is 1008.35
Invalid Account Balance after 2 months is 0

Valid Account Balance 0 after 3 months is 1250.06
Valid Account Balance 1 after 3 months is 1012.55
Invalid Account Balance after 3 months is 0

Valid Account Balance 0 after 4 months is 1255.26
Valid Account Balance 1 after 4 months is 1016.77
Invalid Account Balance after 4 months is 0

Valid Account Balance 0 after 5 months is 1260.5
Valid Account Balance 1 after 5 months is 1021.01
Invalid Account Balance after 5 months is 0

Valid Account Balance 0 after 6 months is 1265.75
Valid Account Balance 1 after 6 months is 1025.26
Invalid Account Balance after 6 months is 0


C:\Users\schem\source\repos\schemp98\Cpp_Certification_Course\Debug\Exercise.exe (process 15972) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console
when debugging stops.
Press any key to close this window . . .
```