```cpp
//
// 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";
```

```
53     }
54
55     // set Annual Interest Rate
56     cout << "\n--------------------------------------------------------------\n" <<
57        "Setting Savings Account interest rate to " << interestRate << "%\n";
58     ShaunChemplavil::SavingsAccount::setAnnualInterestRate(interestRate);
59
60     for (int month = 1; month <= 6; month++)
61     {
62        validAccount0.applyMonthlyInterest();
63        validAccount1.applyMonthlyInterest();
64        invalidAccount.applyMonthlyInterest();
65
66        // exercising all of the get functions
67        cout << "Valid Account Balance 0 " << "after " << month << " months is "
68           << validAccount0.getSavingsBalance() << "\n"
69           << "Valid Account Balance 1 " << "after " << month << " months is "
70           << validAccount1.getSavingsBalance() << "\n"
71           << "Invalid Account Balance " << "after " << month << " months is "
72           << invalidAccount.getSavingsBalance() << "\n\n";
73     }
74  }
75
```