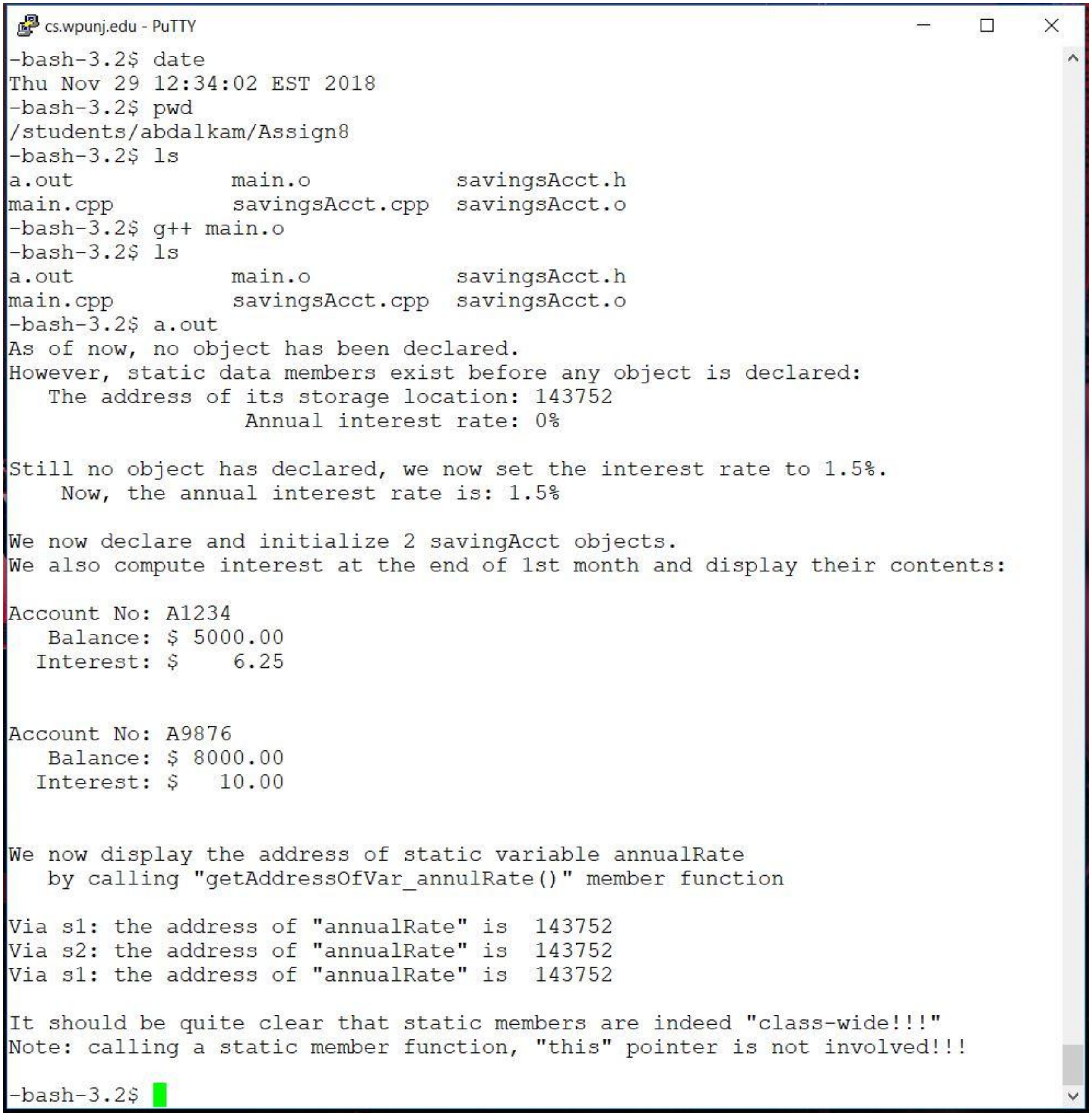
```
//             Course:  CS2400-60 Computer Science 2
//               Name:  Abdalkarim, Marina
//       Assignment:  Programming Assignment P9.1
//    Date assigned: 11/19/18
//           Date due: 12/6/18
//  Date handed in: 12/4/18
//             Remark: The program tests all functions and separate files.
// Code of savingsAcct.h:
#ifndef savingsAcct_H
#define savingsAcct_H
#include <iostream>
using namespace std;
class savingsAcct
{
        friend istream& operator>>(istream&, savingsAcct&);
        // Postcondition: the integer values entered by the user are assigned to the data
        // members
        friend ostream& operator<<(ostream&, const savingsAcct&);
        // Postcondition: displays the contents of the class passed to the function
public:
        savingsAcct();
        savingsAcct(string acctNum, double amt);
        // Postcondition: balance is set to the value passed to parameter amt
        static double * getAddressOfVar_annulRate();
        // Postcondition: return the address of static variable annualRate (should the same for all
        // objects!)
        static double getAnnualRate();
        // Postcondition: returns annual interest rate
        static void setAnnualRate(double rate);        // change the annual interest rate
        // Postcondition: annualRate has been changed to the value passed to parameter rate
        void computeIntrest();
        // Interest is computed at the end of each month using monthly interest rate
        // (i.e.,annualRate/12)
        // Postcondition: interest is computed at the end of the month using the current balance
private:
        string acctNo;                  // e.g., A1234
        double balance;
        static double annualRate;       // classwide; store in a memory location shared by objects
        double interest;                // monthly interest
};
double savingsAcct::annualRate;
#endif
```

```
// Code of savingsAcct.cpp:
#include "savingsAcct.h"
#include <cassert>
#include <iostream>
#include <string>
#include <iomanip>
using namespace std;
savingsAcct::savingsAcct()
{
	acctNo = "Nothing";
	balance = 0;
	annualRate = 0;
	interest = 0;
}
savingsAcct::savingsAcct(string acctNum, double amt)
{
	acctNo = acctNum;
	balance = amt;
}
istream& operator>>(istream& in, savingsAcct &x)
{
	in >> x.acctNo >> x.balance;
	return in;
}
ostream& operator<<(ostream& out, const savingsAcct &x)
{
	cout << fixed << setprecision(2);
	out << "Account No: " << x.acctNo << endl;
	out << "   Balance: $ " << x.balance << endl;
	out << "  Interest: $ " << setw(7) << x.interest << endl;
	return out;
}
double *savingsAcct::getAddressOfVar_annulRate()
{
	double *address;
	address = &(annualRate);
	return address;
}
double savingsAcct::getAnnualRate()
```

```
{
        return annualRate;
}
void savingsAcct::setAnnualRate(double rate)
{
        annualRate = rate;
}
void savingsAcct::computeIntrest()
{
        interest = (balance * (1 + (annualRate * 0.0833))) - balance;
}
```

```
// Code of main.cpp:
#include "savingsAcct.h"
#include "savingsAcct.cpp"
#include <cassert>
#include <iostream>
#include <string>
#include <iomanip>
using namespace std;
int main()
{
        cout << "As of now, no object has been declared.\n";
        cout << "However, static data members exist before any object is declared:" << endl;
        cout << "   The address of its storage location: "
                << int(savingsAcct::getAddressOfVar_annulRate()) << endl;
        cout << "                  Annual interest rate: " << savingsAcct::getAnnualRate() * 100
                << "%" << endl << endl;
        cout << "Still no object has declared, we now set the interest rate to 1.5%." << endl;
        savingsAcct::setAnnualRate(0.015);
        cout << "    Now, the annual interest rate is: " << savingsAcct::getAnnualRate() * 100
                << "%" << endl << endl;
        cout << "We now declare and initialize 2 savingAcct objects." << endl;
        cout << "We also compute interest at the end of 1st month and display their
        contents:\n\n";
        savingsAcct s1("A1234", 5000), s2("A9876", 8000);
        s1.computeIntrest();
        s2.computeIntrest();
        cout << s1 << endl << endl;
        cout << s2 << endl << endl;
        cout << "We now display the address of static variable annualRate" << endl;
        cout << "   by calling \"getAddressOfVar_annulRate()\" member function" << endl <<
        endl;
        cout << "Via s1: the address of \"annualRate\" is  " <<
        int(s1.getAddressOfVar_annulRate())
                << endl;
        cout << "Via s2: the address of \"annualRate\" is  " <<
        int(s2.getAddressOfVar_annulRate())
                << endl;
        cout << "Via s1: the address of \"annualRate\" is  "
                << int(savingsAcct::getAddressOfVar_annulRate()) << endl;
        cout << "\nIt should be quite clear that static members are indeed \"class-wide!!!\"" <<
        endl;
        cout << "Note: calling a static member function, \"this\" pointer is not involved!!!\n\n";
    return 0;
}
```



```
-bash-3.2$ date
Thu Nov 29 12:34:02 EST 2018
-bash-3.2$ pwd
/students/abdalkam/Assign8
-bash-3.2$ ls
a.out            main.o           savingsAcct.h
main.cpp         savingsAcct.cpp  savingsAcct.o
-bash-3.2$ g++ main.o
-bash-3.2$ ls
a.out            main.o           savingsAcct.h
main.cpp         savingsAcct.cpp  savingsAcct.o
-bash-3.2$ a.out
As of now, no object has been declared.
However, static data members exist before any object is declared:
   The address of its storage location: 143752
                  Annual interest rate: 0%

Still no object has declared, we now set the interest rate to 1.5%.
    Now, the annual interest rate is: 1.5%

We now declare and initialize 2 savingAcct objects.
We also compute interest at the end of 1st month and display their contents:

Account No: A1234
   Balance: $ 5000.00
  Interest: $    6.25


Account No: A9876
   Balance: $ 8000.00
  Interest: $   10.00


We now display the address of static variable annualRate
   by calling "getAddressOfVar_annulRate()" member function

Via s1: the address of "annualRate" is  143752
Via s2: the address of "annualRate" is  143752
Via s1: the address of "annualRate" is  143752

It should be quite clear that static members are indeed "class-wide!!!"
Note: calling a static member function, "this" pointer is not involved!!!

-bash-3.2$ 
```