

## Programming Assignment #6

### Build Your Nest Egg

CS 1428.254, Spring 2012

Instructor: Jill Seaman

**Due: in class Tuesday, 4/10/2012** (upload electronic copy by 9:00am)

---

#### Problem:

You've received a windfall and you want to know how much it will grow if you invest a certain amount for a certain number of years, at various interest rates.

Write a program that asks the user for an initial investment amount, the interest rate (compounded annually), and the number of years to let the investment grow.

Then calculate and output the final accumulated value of the investment, based on the input values.

The formula to compute the investment total is the future value formula:

$$F = P * (1 + r)^n$$

where F is the future value (the amount that will have accumulated at the end of the investment period), P is the present value (the amount deposited up front), r is the annual interest rate, and n is the number of years the money will be invested.

After displaying the future value, then output a table showing the balance of the investment at the end of each year that the money is invested. To output the table, you will calculate F using the input initial value and interest rate, with n taking on 1 through the total number of years. In other words, the first year will use P, r, and n=1, the second year will use the same P and r, but n=2, and so on up to the final number of years of the investment.

Finally, you will notify the user in which year the balance will reach one million dollars (\$1,000,000). The formula to calculate the number of years to reach one million is:

$$n = (\log(1000000) - \log(P)) / \log(1 + r)$$

where log is a function in the cmath library. Note: if the value n has a fractional part (i.e. 12.34 years) it should be rounded up to the next whole number (13 years). The function ceil (for ceiling) in the cmath library will do this. Then add this to the current year (2012).

After calculating these values, ask the user if they want to run the calculator again, and repeat as requested.

### **Input:**

Ask the user for the initial investment amount (the present value), the interest rate, and the number of years of the investment. The initial investment value and interest rate may have fractional values. You should validate the input.

- The initial investment should be at least \$100 but not more than \$1,000,000
- The interest rate should be greater than 0 and less than or equal to 50.
- The years should not be over 60 (and greater than 0).

If the input is out of range, use an informative prompt to ask the user to re-enter valid data until the input is valid.

For the calculations to work properly, you must convert the interest rate which is input as a percent between 0 and 100 (i.e. 3.5%) to a fractional value between 0 and 1 (i.e. 0.035).

**Processing:** You should use the following 4 functions in your program.

`getInput`: has three reference parameters to input initial investment value, interest rate, and number of years from the user.

`calculateFV`: has 3 parameters: present value, interest rate, and years of investment. Returns the calculated future value. Do not do any output from this function.

`yearsToOneMillion`: has 2 parameters, the initial value and the interest rate. Returns the calculated number of years it takes for the balance to reach one million. Do not do any output from this function.

`outputTable`: has 3 parameters: initial investment (present value), interest rate, and total years. It outputs the table with column headers, and one row for each year from 0 to total years, showing the balance at the end of that year.

**Output:** Display the dollar amounts and formatted to 2 decimal places. It is not necessary to show dollar signs in the table. The values in the table should line up (right-justified). The year values should be integers. You may assume that the balance will fit within 20 characters.

### Sample output:

```
Enter your initial investment amount: 150000
Enter the annual interest rate: 14.5
Enter the number of years of the investment: 20
```

```
Future value of investment: $2250095.71
```

Year	Balance
0	150000.00
1	171750.00
2	196653.75
3	225168.54
4	257817.98
5	295201.59
6	338005.82
7	387016.66
8	443134.08
9	507388.52
10	580959.86
11	665199.04
12	761652.90
13	872092.57
14	998545.99
15	1143335.16
16	1309118.76
17	1498940.98
18	1716287.42
19	1965149.10
20	2250095.71

```
One million dollar year: 2027
```

```
Would you like to run the calculator again(Y/N)? : N
```

---

### Style:

See the Style Guidelines document on the course website. **Especially pay attention to the comments required for functions.** The grader will deduct points if your program violates the style guidelines.

### Note:

Some common interest rates are: 1.15% (Certificates of Deposit), 3.35% (30 year US Treasury Bond), 10% (average for stock mutual funds over 30 year period), 20% (average of stock mutual funds from 1990 to 1999).

You could use an on-line investment calculator to check your results, but keep in mind that these may compound the interest monthly, whereas our calculator compounds the interest yearly.

**[An online investing calculator \(click here\)](#)**

Click on Launch Calculator, then click on Use Advanced Version, and set Compounding Freq. to yearly. Then fill in starting Balance, the number of years, and the Rate of Return.

**Logistics:**

Your program must be free of compiler errors.

Do you not use any features of C++ that we have not yet covered in class (use features from Chapters 1-6 only (arrays are not needed for this program)).

Name your file **assign6\_XXXXX.cpp** where XXXXX is your TX State NetID (your txstate.edu email id). The file name should look something like this: assign6\_js236.cpp

There are two steps to the turn-in process:

1. Submit an electronic copy using the upload link on the course webpage (or go directly to <https://hwupload.cs.txstate.edu>)  
**by 9:00am, Tuesday 4/10/2012**
2. Submit a printout of the file at the **beginning of class on Tues 4/10/2012**.  
Please print your name on the front page, staple if there is more than one page.

See the following for details about turning in homework late:

<http://www.cs.txstate.edu/~js236/cs1428/assignmentpolicy.txt>