Program Development

Lecture 8
Section 1.6

Robb T. Koether

Hampden-Sydney College

Wed, Sep 11, 2019

Outline

Program Development: Step 1

- Step 1:
  - Decide what the input and the output should be.
  - The input tells you what information you have to work with.
  - The output tells you what the goal is.

Program Development: Step 2

- Step 2:
  - Work an example by hand.
  - Choose simple values.
  - Avoid special cases.

Program Development: Step 3

- Step 3:
  - Sketch the algorithm, based on the example.
  - Use any convenient mixture of English syntax and C++ syntax.
  - Avoid technical issues that can be addressed later (e.g., **`float`** vs. **`int`**).

Program Development: Step 4

- Step 4:
  - Fill in details until it is clear how each step will be written in C++.

Program Development: Step 5

- Step 5:
  - Write the program in C++.
  - Once Step 4 is complete, it should be clear how to write the program statements.

Program Development: Step 6

- Test the program for correctness.

Outline

Example – Savings Account

Example (Example)

- Write a program that will compute the amount of interest earned on a savings plan, given the amount of each monthly deposit, the interest rate, and the number of years.
- Assumptions
  - The deposits are equal and made monthly.
  - The first deposit is made at the *end* of the first month.
  - Interest is applied at the end of each month to the balance *during* that month.
  - The duration of the loan will be a whole number of years.
  - The interest rate is constant over the years.

Step 1 - Input and Output

Example (Input and Output)

- The input will be
  - The monthly deposit.
  - The interest rate as an annual percent.
  - The number of years.

Step 1 - Input and Output

Example (Input and Output)

- The input will be
  - The monthly deposit.
  - The interest rate as an annual percent.
  - The number of years.
- The output will be
  - The total amount of the account.
  - The total amount invested.
  - The total interest earned.

Step 2 - Work an Example

Example (Work an Example)

- Let the monthly deposit be \$500.00, the interest rate be 12% per year, and the duration be 30 years.

Step 2 - Work an Example

Example (Work an Example)

- Let the monthly deposit be \$500.00, the interest rate be 12% per year, and the duration be 30 years.
- The monthly interest rate is

$$r = 0.12/12 = 0.01$$

  and the number of months is

$$n = 30 \times 12 = 360.$$

Step 2 - Work an Example

Example (Work an Example)

- Let the monthly deposit be \$500.00, the interest rate be 12% per year, and the duration be 30 years.
- The monthly interest rate is

$$r = 0.12/12 = 0.01$$

  and the number of months is

$$n = 30 \times 12 = 360.$$

- Doing the calculations month by month could take (us) a long time.

Step 2 Continued

Example (Work an Example)

- The formula for the account balance is

$$F = P\left(\frac{(1+r)^n - 1}{r}\right),$$

  where
  - P is the monthly deposit.
  - F is the future value of the account.
  - r is the *monthly* interest rate.
  - n is the number of *months*.

Step 2 Continued

Example (Work an Example)

- Compute the amount of the account

$$\begin{aligned} F &= 500\left(\frac{(1+0.01)^{360}-1}{0.01}\right) \\ &= \$1{,}747{,}482.07. \end{aligned}$$

Step 2 Continued

Example (Work an Example)

- Compute the amount of the account

$$F = 500\left(\frac{(1+0.01)^{360}-1}{0.01}\right)$$
$$= \$1{,}747{,}482.07.$$

- Total amount invested is

$$360 \times \$500 = \$180{,}000.$$

Step 2 Continued

Example (Work an Example)

- Compute the amount of the account

$$\begin{aligned} F &= 500\left(\frac{(1+0.01)^{360}-1}{0.01}\right) \\ &= \$1{,}747{,}482.07. \end{aligned}$$

- Total amount invested is

$$360 \times \$500 = \$180{,}000.$$

- Total interest earned is

$$\$1{,}747{,}482.07 - \$180{,}000 = \$1{,}567{,}482.07.$$

Step 3 - Write the Algorithm

Example (Write the Algorithm)

- Divide the annual rate by 12 to get the monthly rate r.

Step 3 - Write the Algorithm

Example (Write the Algorithm)

- Divide the annual rate by 12 to get the monthly rate r.
- Multiply the number of years by 12 to get the number of months n.

Step 3 - Write the Algorithm

Example (Write the Algorithm)

- Divide the annual rate by 12 to get the monthly rate r.
- Multiply the number of years by 12 to get the number of months n.
- Substitute P, r, and n into the formula to get F.

Step 3 - Write the Algorithm

Example (Write the Algorithm)

- Divide the annual rate by 12 to get the monthly rate r.
- Multiply the number of years by 12 to get the number of months n.
- Substitute P, r, and n into the formula to get F.
- Total amount invested is $A = P \times n$.

Step 3 - Write the Algorithm

Example (Write the Algorithm)

- Divide the annual rate by 12 to get the monthly rate r.
- Multiply the number of years by 12 to get the number of months n.
- Substitute P, r, and n into the formula to get F.
- Total amount invested is $A = P \times n$.
- Total interest earned is $I = F - A$.

Step 4 - Fill in the Details

Example (Fill in the Details)

- No further details are necessary in this example.

Step 5 - Write the Program

Example (Write the Program – Input)

```
//  Get the monthly deposit
cout << "Enter the monthly deposit: ";
double amt;
cin >> amt;

//  Get the annual interest rate
cout << "Enter the annual interest rate (as %): ";
double ann_rate;
cin >> ann_rate;

//  Get the number of years
cout << "Enter the number of years: ";
double yrs;
cin >> yrs;
```

Step 5 Continued

Example (The Power Function)

- How do we calculate $(1 + r)^n$?

Step 5 Continued

Example (The Power Function)

- How do we calculate $(1 + r)^n$?
- That is, how do we raise a number to a power in C++?

Step 5 Continued

Example (The Power Function)

- How do we calculate $(1 + r)^n$?
- That is, how do we raise a number to a power in C++?
- There is a "power" function named `pow()`.

$$\texttt{pow(x, y)} \text{ computes } \texttt{x}^{\texttt{y}}.$$

Step 5 Continued

Example (Write the Program – Calculations)

```
//  Compute monthly rate and number of deposits
double mon_rate = ann_rate/12.0;
double num_mons = 12.0 * yrs;

// Compute the future value
double amt = dep*(pow(1.0 + mon_rate, num_mons) - 1.0)
             /mon_rate;

// Compute the total investment and interest
double amt_inv = num_mons * dep;
double tot_int = amt - amt_inv;
```

Step 5 Continued

Example (Write the Program – Output)

```
//  Output the results
cout << "Account balance = $" << amt << endl;
cout << "Total invested = $" << amt_inv << endl;
cout << "Interest earned = $" << tot_int << endl;
```

Example of Program Development

Example (Test the Program)

- Enter the program, run it, and test it.

Outline

Program Debugging

- To debug a program is to correct its errors.
- First we must correct all syntax (compile-time) errors.
- Do not use trial and error. That only makes matters worse.
- Understand each error before attempting to correct it.

Program Debugging

- Then we must correct the logical (run-time) errors.
- Do not use trial and error. That only makes matters worse.
- Understand each error before attempting to correct it.
- Insert output statements to display values of key variables.
- Comment out segments of code.
- Use a debugger.

The Visual Studio Debugger

- The Visual Studio debugger allows the programmer to
  - Execute the program one statement at a time.
  - Check the values of variables during execution.
  - By comparing the actual values to the correct values, the programmer can pinpoint the statement where his program is going wrong.
- To use the debugger, press the F10 key.
- Repeatedly press F10 to step through the program, one statement at a time.

Outline

Program Testing

- To test a program is to demonstrate that it is error-free.
- Use the worked example as test data.
- Check the output for correctness.
- Test special cases.
- Be sure that every possible path in the program is executed.

Outline

Example – Making Change

Example (Example)

- Write a program that will read an amount of money representing change and then output the number of dollar bills, quarters, dimes, nickels, and pennies that represent that amount.
- Assumptions
  - The amount is expressed in dollars and cents (one number).
  - We give as many dollar bills as possible, then as many quarters as possible, and so on.

Step 1 - Input and Output

Example (Input and Output)

- The input will be
  - The amount of change

Step 1 - Input and Output

Example (Input and Output)

- The input will be
  - The amount of change
- The output will be
  - The number of dollar bills.
  - The number of quarters.
  - The number of dimes.
  - The number of nickels.
  - The number of pennies.

Step 2 - Work an Example

Example (Work an Example)

- Let the change be $2.87.

Step 2 - Work an Example

Example (Work an Example)

- Let the change be \$2.87.
- The number of dollar bills is 2, leaving \$0.87, or 87¢.

Step 2 - Work an Example

Example (Work an Example)

- Let the change be \$2.87.
- The number of dollar bills is 2, leaving \$0.87, or 87¢.
- Remove 3 quarters (75¢) from 87¢, leaving 12¢.

Step 2 - Work an Example

Example (Work an Example)

- Let the change be \$2.87.
- The number of dollar bills is 2, leaving \$0.87, or 87¢.
- Remove 3 quarters (75¢) from 87¢, leaving 12¢.
- Remove 1 dime (10¢) from 12¢, leaving 2¢.

Step 2 - Work an Example

Example (Work an Example)

- Let the change be $2.87.
- The number of dollar bills is 2, leaving $0.87, or 87¢.
- Remove 3 quarters (75¢) from 87¢, leaving 12¢.
- Remove 1 dime (10¢) from 12¢, leaving 2¢.
- Remove 0 nickels.

Step 2 - Work an Example

Example (Work an Example)

- Let the change be $2.87.
- The number of dollar bills is 2, leaving $0.87, or 87¢.
- Remove 3 quarters (75¢) from 87¢, leaving 12¢.
- Remove 1 dime (10¢) from 12¢, leaving 2¢.
- Remove 0 nickels.
- The remaining amount is 2 pennies.

Step 3 - Write the Algorithm

Example (Write the Algorithm)

1. Get the integer part of the input value. That represents the number of dollar bills.

Step 3 - Write the Algorithm

Example (Write the Algorithm)

1. Get the integer part of the input value. That represents the number of dollar bills.
2. Subtract the number of dollars from the amount and then multiply by 100 to get the cents.

Step 3 - Write the Algorithm

Example (Write the Algorithm)

1. Get the integer part of the input value. That represents the number of dollar bills.
2. Subtract the number of dollars from the amount and then multiply by 100 to get the cents.
3. Get the whole number quotient of the amount divided by 25. That represents the number of quarters.

Step 3 - Write the Algorithm

Example (Write the Algorithm)

1. Get the integer part of the input value. That represents the number of dollar bills.
2. Subtract the number of dollars from the amount and then multiply by 100 to get the cents.
3. Get the whole number quotient of the amount divided by 25. That represents the number of quarters.
4. Get the remainder of that same quotient. That is the remaining change.

Step 3 - Write the Algorithm

Example (Write the Algorithm)

1. Get the integer part of the input value. That represents the number of dollar bills.
2. Subtract the number of dollars from the amount and then multiply by 100 to get the cents.
3. Get the whole number quotient of the amount divided by 25. That represents the number of quarters.
4. Get the remainder of that same quotient. That is the remaining change.
5. Repeat steps 3 and 4, using divisors 10 and then 5.

Step 3 - Write the Algorithm

Example (Write the Algorithm)

1. Get the integer part of the input value. That represents the number of dollar bills.
2. Subtract the number of dollars from the amount and then multiply by 100 to get the cents.
3. Get the whole number quotient of the amount divided by 25. That represents the number of quarters.
4. Get the remainder of that same quotient. That is the remaining change.
5. Repeat steps 3 and 4, using divisors 10 and then 5.
6. After getting the nickels, whatever is left is the pennies.

Step 4 - Fill in the Details

Example (Fill in the Details)

- No further details are necessary in this example.

Step 5 - Write the Program

Example (Write the Program – Input)

```
cout << "Enter the change, in dollars: ";
float amt;                  // The amount of change
cin >> amt;                 // Read the amount
```

Step 5 Continued

Example (Write the Program – Calculations)

```
int dollars = amt;              // Truncate to get dollars
int change = 100*(amt - dollars); // Get cents
int quarters = change/25; // No. of quarters
change = change % 25;       // Remaining change
```

Step 5 Continued

Example (Write the Program – Calculations)

```
int dimes = change/10;    // No. of dimes
change = change % 10;     // Remaining change
int nickels = change/5;  // No. of nickels
change = change % 5;      // Remaining change
int pennies = change;     // No. of pennies
```

Step 5 Continued

Example (Write the Program – Output)

```
cout << "Number of dollar bills = " << dollars << endl;
cout << "Number of quarters = " << quarters << endl;
cout << "Number of dimes = " << dimes << endl;
cout << "Number of nickels = " << nickels << endl;
cout << "Number of pennies = " << pennies << endl;
```

Example of Program Development

Example (Test the Program)

- Enter the program, run it, and test it.

Outline

Assignment

- Read Section 1.6.