Output Formats

Larry Caretto
Computer Science 106

Computing in Engineering and Science

February 14, 2006

California State University Northridge

Outline

- Review last week
  - Data types, assignment, expressions, operators and type conversions
- Formatting output
  - Controlling appearance compared with default output of six significant figures
  - Manipulators for width and precision
  - Interaction of manipulators
  - Designing output commands

Review

- All variables have certain data type
  - Use int, double, char, string and bool
  - Integer division truncates and double is approximate (15-16 significant figures)
- Conversion of values among types
- Assignment operator and expressions
- Operators and operator precedence
  - Order: 1. –(unary), 2. * / %, 3. + -(binary)
  - Use parentheses to overcome normal rules

Formatting

- Controlling appearance of output
- Default C++ output gives
  - Six significant figures
  - Does not print trailing zeros
  - Does not print decimal point for floating point numbers with no decimal fraction
  - Chooses to use fixed (123.456) or scientific (1.23456e+123) based on size of number
  - Does not provide any spaces

What We Would Like to Do

- Control appearance of output
  - Select number of significant figures
  - Choose to print trailing zeros
  - Choose to print decimal point for floating point numbers with no decimal fraction
  - Choose to use fixed (123.456) or scientific (1.23456e+123)
  - Space numbers as in output tables
  - Align numbers left or right

How do We Format Output?

- Use manipulators in output statements
- Requires use of #include<iomanip>
- Manipulators we will use
  - **fixed** forces fixed format output
  - **scientific** forces scientific format output
  - **setw(w)** assigns w spaces for output (right justified by default; see **left** and **right**)
  - **setprecision(p)** uses the value of p to set the number of significant figures

Using setw(w) Only

- double x = 123.456; cout << x;
  - prints 123.456 in seven spaces (six digits, decimal point, no blanks)
- cout << setw(10) << x; uses 10 spaces
  - Result is bbb123.456 (b is blank space)
- What is result of cout << setw(20) << x;
  - 13 leading blank spaces before 123.456
- What is result of cout << setw(2) << x;
  - Prints full number 123.456 (no spaces)

California State University Northridge 7

Using fixed and scientific

- double x = 123.4e-12, y = 1.2e12, z = 1;
- Start with default output
- cout << x << " " << y << " " << z;

  1.234e-010 1.2e+012 1
- cout << scientific << x <<" " << y << " " << z;

1.234000e-010 1.200000e+012 1.000000e+000
- cout << fixed << x << " " << y << " " << z;

  0.0000000001234 1200000000000 1

California State University Northridge 8

Persistence of Manipulators

- setw(w) is in effect for one output item only even in same output statement
  - cout << setw(10) << x << setw(10) << y;
- All other manipulators are in effect until changed (even over multiple output statements)
  - cout << fixed << x << " " << y;
  - cout << z; // fixed still in effect!

California State University Northridge 9

Return to Default

- To return to default output where computer selects fixed or scientific
- Use one of the two commands below to turn off the most recently set manipulator
- cout << resetiosflags(iosbase::fixed);
- cout << resetiosflags(iosbase::scientific);

California State University Northridge 10

Use of setprecision(p)

- Effect depends on other options
  - For default output (not using fixed or scientific manipulators) setprecision(p) gives p significant figures
  - If fixed or scientific manipulators are used, setprecision(p) gives p decimal places
    - For scientific output p decimal places is p+1 significant figures
- Once used, setprecision(p) remains in effect until changed
- Setprecision(6) restores default

California State University Northridge 11

Examples of setprecision

- For default output (no use of fixed or scientific manipulators), p is significant figures and computer chooses format
- double a = 1, b = 12345678901.2;
- cout << setprecision(2) << a << " " << b;
  - Output is: 1 1.2e+010
- cout << setprecision(12) << a << " " << b;
  - Output is:1 12345678901.2
- cout << setprecision(11) << a << " " << b;
  - Output is: 1 12345678901

California State University Northridge 12

Combine setprecision and fixed

- Fixed output has decimal point and p specifies number of decimal places
- double a = 1, b = 12345678901.2; cout << fixed;
- cout << setprecision(2) << a << " " << b;
  - Output is: 1.00 12345678901.20
- cout << setprecision(12) << a << " " << b;
  - 1.000000000000 12345678901.200000000000

Combine setprecision and scientific

- Scientific manipulator forces e±000 format and p is decimal places
- double a = 1, b = 12345678901.2; cout << scientific
- cout << setprecision(2) << a << " " << b;
  Output is: 1.00e+000 1.23e+010
- cout << setprecision(12) << a << " " << b;
  1.00000000000e+000 1.23456789012e+010

Combine setw and setprecision

- setprecision controls the number of digits to be printed
- setw controls how many spaces are used for the output
- What happens if the setw manipulator does not give enough spaces
  - The entire output item is printed
  - There is no space between the output item and the previous output (if any)

setw and setprecision Examples

- double x = 1.234; y = 23456.789;
- cout << setprecision(2) << setw(10) << x << setw(10) << y; // no fixed/scientific

```
1234567890123456789O<-Counter
bbbbbbb1.2bb2.3e+004
```

- cout << fixed << setprecision(2);
- cout setw(10) << x << setw(10) << y;

```
1234567890123456789O<-Counter
bbbbbb1.23bb23456.79
```


setw, setprecision and scientific

- Need 5 spaces for exponent (e±000) plus p spaces after the decimal
- Must also add one space for decimal point and one for possible minus sign
- Add leading spaces to get final width
- double x = –1.234; y = 23456.789;
- cout << scientific << setprecision(5);
- cout setw(15) << x << setw(15) << y;

```
123456789012345678901234567890
bb-1.23400e+000bbb2.34568e+004
```


Problem: Writing Tables

- Print variables x1, x2, x3, x4, y1, y2, y3, and y4 in table with one column for x, one column for y, and four rows
- Print numbers in fixed format with two decimal places (assume the magnitude of all numbers is less than 10000, but some may be negative)
- Leave at least five spaces between numbers
- How would you do this?

Writing Tables II

- How to write variables x1, x2, x3, x4, y1, y2, y3, and y4 in table with one column for x, one column for y, and four rows
  - This means that we have to output variables as follows (formatting not shown) cout << x1 << y1 << endl << x2 << y2 << endl << x3 << y3 << endl << x4 << y4;
- Next determine how to accommodate format requirements

California State University Northridge 19

Writing Tables III

- How to print numbers in fixed format with two decimal places (how many spaces for numbers with absolute value less than 10000, but may be negative)
  - Use fixed manipulator to give fixed format
  - When using fixed and setprecision(p), p is the number of decimal places
  - Output may need 8 spaces: 1 place each for sign and decimal point and 4 + 2 spaces for numbers left and right of decimal

California State University Northridge 20

Writing Tables IV

- Output statement leaves at least five spaces between numbers; what is setw?
  - Since each number requires at most 8 spaces we need a setw(13)
- What is final output statement?
  - cout << fixed << setprecision(2) << setw(13) << x1 << setw(13) << y1 << endl << setw(13) << x2 << setw(13) << y2 << endl << setw(13) << x3 << setw(13) << y3 << endl << setw(13) << x4 << setw(13) << y4;

California State University Northridge 21

Assignments

- Reading pages in text
  - Today – pp 117 – 137
  - Thursday – pp 143- – 152
  - February 21 – pp 167 – 187 and 200 – 210
- February 21 homework problems
  - Page 98, check-point 3.11; page 106, checkpoint 3.14; page 153, problem 8; page 162, problem 9
- Exercise 3 due Thursday
- Quiz in Laboratory – February 23
  - Exercises 1 – 3; text pages 1 – 98; homework for February 7, 14, and 21(first three problems)

California State University Northridge 22