

CSE 105, Lab 2, Summer 2006

Instructor: Cory Strobe

July 16, 2006

1. Review of items added to last weeks lab

- To move back to the root directory: `cd`
- To move back one directory level: `cd ..`
- To change permissions: `chmod 755 filename`
- To remove (delete) a file: `rm filename`

2. *Demo*: Accessing cse files using windows (computer science labs only).

3. Handing in assignments through email:

- **pine**: In pine (cse's email system), type `c` (compose). In the address bar, type `cstrobe@cse.unl.edu`, and place cursor in the Attchmnt field. Type `ctrl-t`, and navigate to the correct file. Once you find the file, hit enter, and it will be attached to the message. You may attach more files by simply typing `ctrl-t` again in the attachment field.
- Using a web based email program (MSN, hotmail, gmail, yahoo, ...): Go to the attachment field in your email program, and click the browse button. Access cse files as was done previously (point 2).

4. More controlled formatting for `printf()`

- Integers
 - In its simplest form, `%d`, will print all digits with no leading zeros.
 - To specify a minimum field size, use `%xd`, where `x = 1, 2, 3, . . .`. This will have a minimum field of `x` digits, padding the number with leading spaces if the number has less the `x` digits.

- If an integer has more than x digits, the field is increased to allow for all digits to be displayed.
- Real numbers
 - In its simplest form, `%f`, will print all digits to the left of the decimal point with no leading zeros and 6 decimal places, padding with extras zeroes if needed.
 - Minimum field size and limited decimal places, `%x.yf`, where $x = 1, 2, 3, \dots$, and $y = 1, 2, 3, \dots$. This will provide a minimum field of x total digits, padding the number with leading spaces if the number has less the x total digits. There will also be only y decimal places displayed. If the number has more decimal places than specified, the number is rounded off (not truncated). If the number has fewer decimal places than specified, trailing zeroes are added to make up the difference.
 - If a real number has more than x total digits, the field is increased to allow for all digits to the left of the decimal point plus the number of digits specified to the right of the decimal point.
- Exercise in understanding
 - The following pairs of numbers (one integer, followed by one real number) are to be printed, one pair after the other using each of the following formats. Indicate the appearance of the result in the grids below:

<u>int1</u>	<u>doubl</u>
213	2.9876
3	3123.4
74878	78.3
11	11

- `printf("%d %f", int1, doubl);` (used for all four pairs)

- `printf("%6d%10.3f", int1, doubl);` (used for all four pairs)

- Check your answer by downloading the source code "formatExercise.c" from the labC webpage, compiling and running. The first two displays are for the exercise given above.
- The third display is different. Below, give the `printf()` statement that would generate the format you see.
- The fourth display is a bit irregular. It was formatted using similar format statements. Why do you think it appears a bit "irregular".

5. Operators

Basic Math: $x + y$ $x - y$ $x * y$ x / y $x \% y$

Relational: $x < y$ $x > y$ $x == y$ $x <= y$ $x >= y$ $x != y$

Assignment: $x = y$

6. Mixing integers and real numbers in expressions:

- Integer with integer produces integer
- Real number with real number produces a real number
- Integer with a real number produces a real number
- When expressions are nested, type is determined after each operation

7. Casting Type

- When division by integers is needed to produce a double, type casting is used.
- The cast is performed by the type desired being enclosed in parenthesis immediately preceding the variable to be cast.
- Given that `var1` is an integer, to cast it as a double is done with `(double)var1`.
- Download the code "mixedMath.c" from the lab webpage. Compile and run.
- Notice when type causes unusual mathematical results (i.e. integer divided by integer)

- The last two examples show two attempts at casting. Why is the first attempt not successful but the second one is?

8. Precedence of Operators introduced so far,

- Expressions enclosed inside parenthesis are evaluated before the operators outside the same parenthesis.
- Highest precedence to lowest precedence (in the absence of parenthesis)

$x * y$ x / y $x \% y$ (left to right)

$x + y$ $x - y$ (left to right)

$x < y$ $x <= y$ $x == y$ $x >= y$ $x > y$ $x != y$ (results are 1 or 0)

$x = y$

- Operators at the same level are evaluated left to right.
- Exercise: Given the following variable and values,

$int1 = 2$, $int2 = 3$, $int3 = 7$, $int4 = 11$ (all integers).

$doub1 = 2$, $doub2 = 3$, $doub3 = 7$, $doub4 = 11$ (all doubles).

- EVALUATE the following expressions:

(a) $int3 + int2 * int4 / int1$

(b) $int3 * int2 - int4 / int1$

(c) $doub3 + doub2 * doub4 / doub1$

(d) $doub3 * doub2 + int4 / int1$

9. Write a program that takes as input the numerators and denominators of two fractions. Your program should display the numerator and denominator of the fraction that represents the product of the two fractions. Also, display the product in decimal notation.
10. Repeat the previous program, but computing the sum of the two fractions instead of the product.
11. Give suggestions on how I can improve our time in lab!