

EECS 10: Computational Methods in Electrical and Computer Engineering

Lecture 9

Rainer Dömer

doemer@uci.edu

The Henry Samueli School of Engineering
Electrical Engineering and Computer Science
University of California, Irvine

Lecture 9: Overview

- Formatted output
 - Formatting of integral values
 - Formatting of floating-point values
 - Example `Formatting.c`
- Programming Principles
 - Algorithm
 - Control flow

Formatted Output

- Formatted output using `printf()`
 - standard format specifiers for integral values
 - `unsigned long long` `%llu`
 - `long long` `%lld`
 - `unsigned long` `%lu`
 - `long` `%ld`
 - `unsigned int` `%u`
 - `int` `%d`
 - `short` `%hd`
 - standard format specifiers for floating point values
 - `long double` `%Lf`
 - `double` `%f`
 - `float` `%f`

EECS10: Computational Methods in ECE, Lecture 9

(c) 2006 R. Doemer

3

Formatted Output

- Detailed formatting sequence for integral values
 - `% flags width length conversion`
 - *flags*
 - (none) standard formatting (right-justified)
 - `-` left-justified output
 - `+` leading plus-sign for positive values
 - `0` leading zeros
 - field *width*
 - (none) minimum number of characters needed
 - integer width of field to be filled with output
 - *length* modifier
 - (none) `int` type
 - `h` `short int` type
 - `l` `long int` type
 - `ll` `long long int` type
 - *conversion* specifier
 - `d` signed decimal value
 - `u` unsigned decimal value
 - `o` (unsigned) octal value
 - `x` (unsigned) hexadecimal value using characters `0-9, a-f`
 - `X` (unsigned) hexadecimal value using characters `0-9, A-F`

EECS10: Computational Methods in ECE, Lecture 9

(c) 2006 R. Doemer

4

Formatted Output

- Detailed formatting sequence for floating-point values
 - *% flags width precision length conversion*
 - **flags**
 - (none) standard formatting (right-justified)
 - - left-justified output
 - + leading plus-sign for positive values
 - 0 leading zeros
 - **field width**
 - (none) minimum number of characters needed
 - integer width of field to be filled with output
 - **precision**
 - (none) default precision (e.g. 6)
 - .int number of digits after decimal point (for **f**, **e**, or **E**), maximum number of significant digits (for **g**, or **G**)
 - **length** modifier
 - (none) float or double type
 - L long double type
 - **conversion** specifier
 - **f** standard floating-point notation (fixed-point)
 - **e** or **E** exponential notation using (**e** or **E**)
 - **g** or **G** standard or exponential notation (using **e** or **E**)

EECS10: Computational Methods in ECE, Lecture 9

(c) 2006 R. Doemer

5

Formatted Output

- Program example: `Formatting.c` (part 1/2)

```

/* Formatting.c: formatted output demo          */
/* author: Rainer Doemer                       */
/* modifications:                              */
/* 10/19/04 RD initial version                 */

#include <stdio.h>

/* main function */

int main(void)
{
    /* output section */
    printf("42 formatted as |%d|: |%d|\n", 42);
    printf("42 formatted as |%8d|: |%8d|\n", 42);
    printf("42 formatted as |%-8d|: |%-8d|\n", 42);
    printf("42 formatted as |%+8d|: |%+8d|\n", 42);
    printf("42 formatted as |%08d|: |%08d|\n", 42);
    printf("42 formatted as |%x|: |%x|\n", 42);
    printf("42 formatted as |%o|: |%o|\n", 42);
    ...
}

```

EECS10: Computational Methods in ECE, Lecture 9

(c) 2006 R. Doemer

6

Formatted Output

- Program example: `Formatting.c` (part 2/2)

```

...
printf("\n");
printf("123.456 formatted as |%f|:      |%f|\n", 123.456);
printf("123.456 formatted as |%e|:      |%e|\n", 123.456);
printf("123.456 formatted as |%g|:      |%g|\n", 123.456);
printf("123.456 formatted as |%12.4f|: |%12.4f|\n",
      123.456);
printf("123.456 formatted as |%12.4e|: |%12.4e|\n",
      123.456);
printf("123.456 formatted as |%12.4g|: |%12.4g|\n",
      123.456);

/* exit */
return 0;
} /* end of main */

/* EOF */

```

EECS10: Computational Methods in ECE, Lecture 9

(c) 2006 R. Doemer

7

Formatted Output

- Example session: `Formatting.c`

```

% vi Formatting.c
% gcc Formatting.c -o Formatting -Wall -ansi
% Formatting
42 formatted as |%d|: |42|
42 formatted as |%8d|: |      42|
42 formatted as |%-8d|: |42      |
42 formatted as |%+8d|: |      +42|
42 formatted as |%08d|: |00000042|
42 formatted as |%x|: |2a|
42 formatted as |%o|: |52|

123.456 formatted as |%f|: |123.456000|
123.456 formatted as |%e|: |1.234560e+02|
123.456 formatted as |%g|: |123.456|
123.456 formatted as |%12.4f|: |      123.4560|
123.456 formatted as |%12.4e|: |  1.2346e+02|
123.456 formatted as |%12.4g|: |      123.5|
%

```

EECS10: Computational Methods in ECE, Lecture 9

(c) 2006 R. Doemer

8

Programming Principles

- Thorough *understanding* of the problem
- *Problem definition*
 - Input data
 - Output data
- *Algorithm*: Procedure to solve the problem
 - Detailed set of *actions* to perform
 - Specification of *order* in which to perform the actions
 - Termination after a *finite* number of steps
- *Pseudo code*: Planning a program
 - Informal (English) description of steps in an algorithm
 - Example: Cake baking recipe
- *Control flow*
 - Execution order of statements in the program
- *Program*: Instructions for the computer
 - Formal description in programming language
 - Statements (steps, actions)
 - Control structures (flow of control)

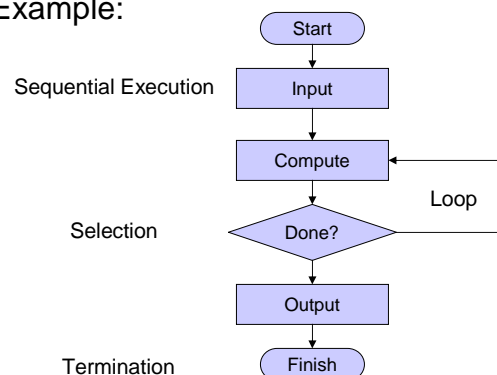
EECS10: Computational Methods in ECE, Lecture 9

(c) 2006 R. Doemer

9

Control Flow

- Control flow charts
 - Graphical representation of program control flow
 - Example:



EECS10: Computational Methods in ECE, Lecture 9

(c) 2006 R. Doemer

10