EECS 10: Computational Methods in Electrical and Computer Engineering

Lecture 9

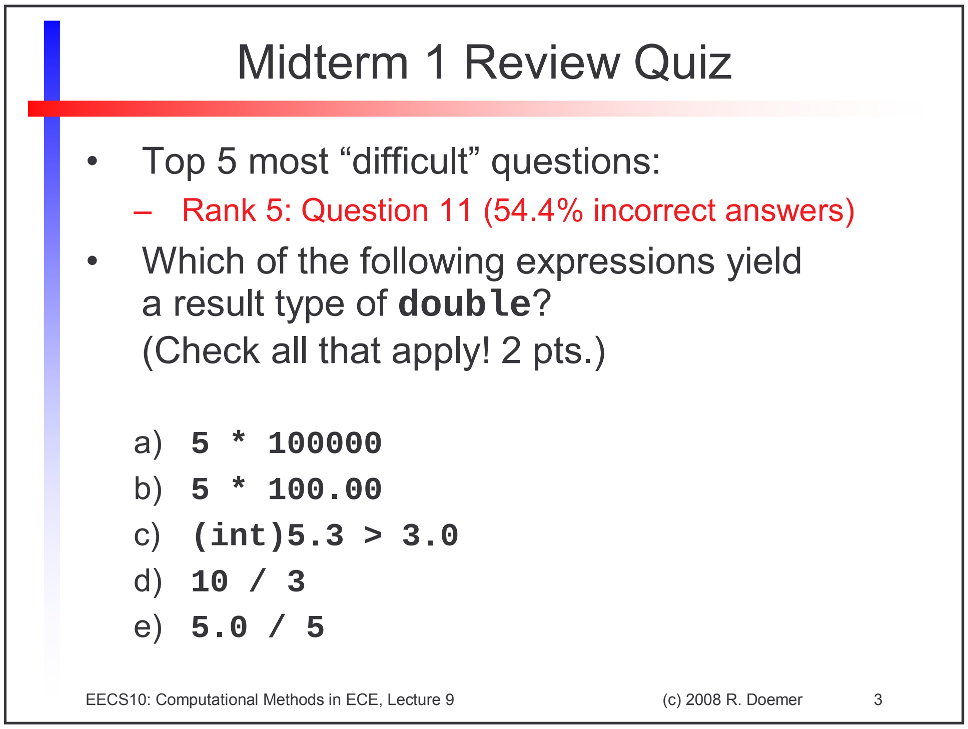
Rainer Dömer

doemer@uci.edu

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

Lecture 9: Overview

- Midterm 1 Review Quiz
  - Top 5 most “difficult” questions
- Formatted output
  - Formatting of integral values
  - Formatting of floating-point values
  - Example **`Formatting.c`**
- Programming Principles
  - Algorithm
  - Control flow

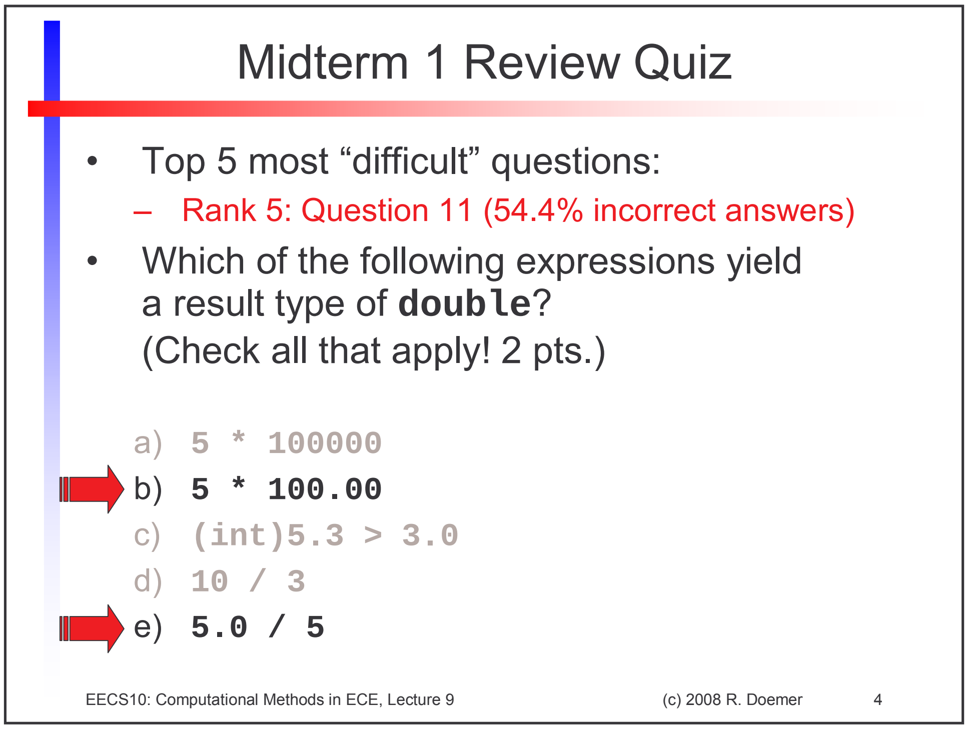


Midterm 1 Review Quiz

- Top 5 most "difficult" questions:
  - Rank 5: Question 11 (54.4% incorrect answers)
- Which of the following expressions yield a result type of **`double`**? (Check all that apply! 2 pts.)

a) `5 * 100000`
b) `5 * 100.00`
c) `(int)5.3 > 3.0`
d) `10 / 3`
e) `5.0 / 5`

EECS10: Computational Methods in ECE, Lecture 9 (c) 2008 R. Doemer 3

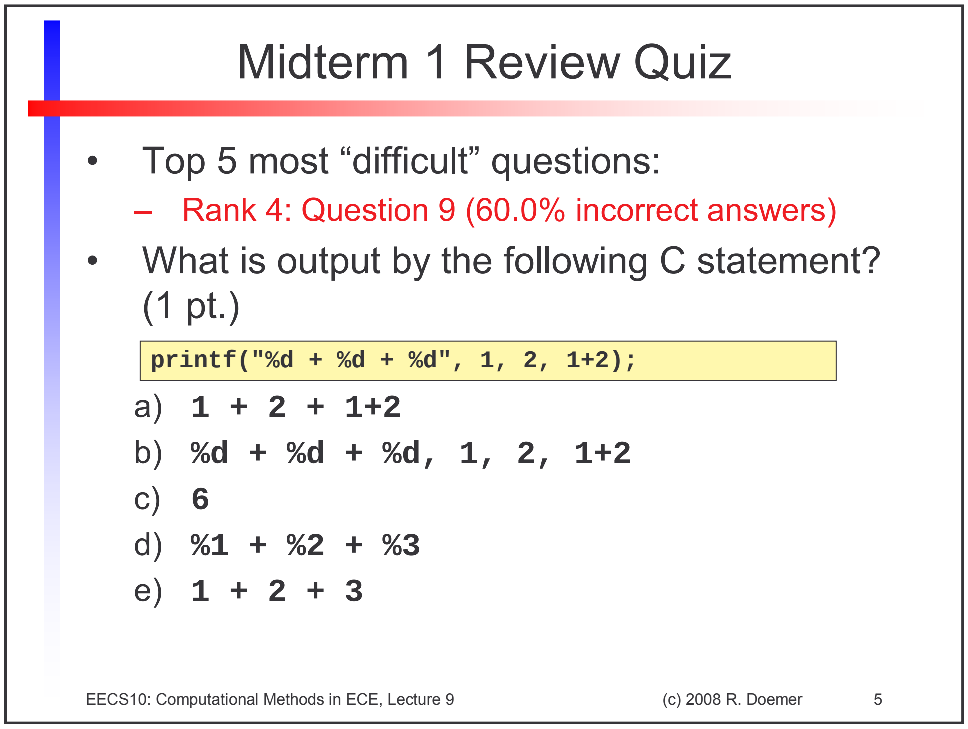


Midterm 1 Review Quiz

- Top 5 most "difficult" questions:
  - Rank 5: Question 11 (54.4% incorrect answers)
- Which of the following expressions yield a result type of **`double`**? (Check all that apply! 2 pts.)

a) `5 * 100000`
[✓ b)] `5 * 100.00`
c) `(int)5.3 > 3.0`
d) `10 / 3`
[✓ e)] `5.0 / 5`

EECS10: Computational Methods in ECE, Lecture 9 (c) 2008 R. Doemer 4



Midterm 1 Review Quiz

- Top 5 most “difficult” questions:
  - Rank 4: Question 9 (60.0% incorrect answers)
- What is output by the following C statement? (1 pt.)

```
printf("%d + %d + %d", 1, 2, 1+2);
```

a) `1 + 2 + 1+2`

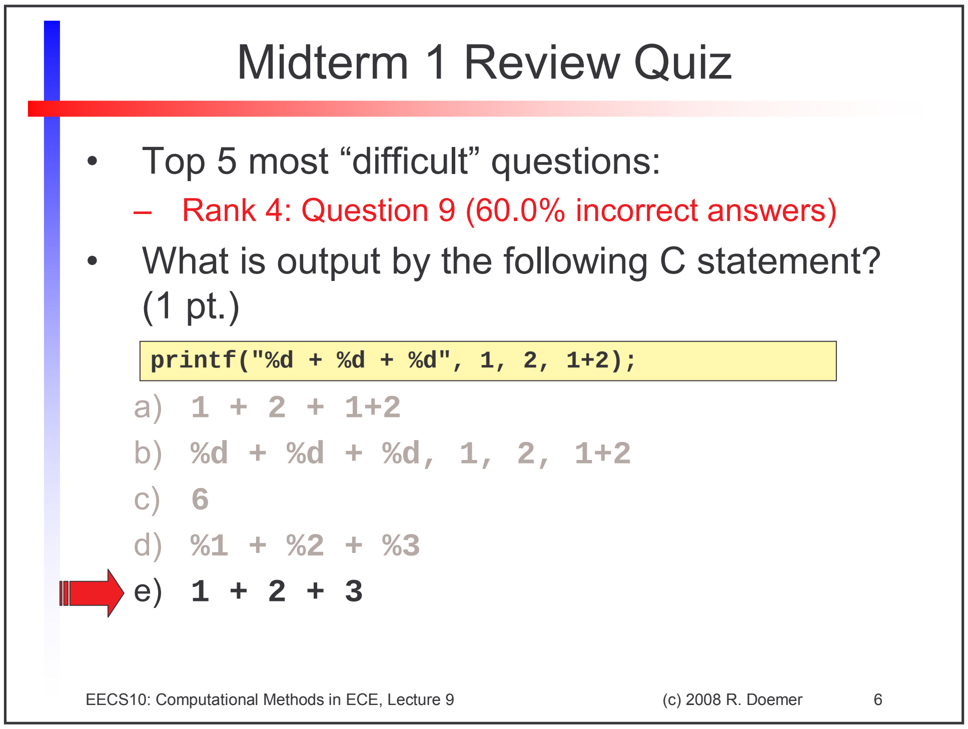
b) `%d + %d + %d, 1, 2, 1+2`

c) `6`

d) `%1 + %2 + %3`

e) `1 + 2 + 3`

EECS10: Computational Methods in ECE, Lecture 9 (c) 2008 R. Doemer 5



Midterm 1 Review Quiz

- Top 5 most “difficult” questions:
  - Rank 4: Question 9 (60.0% incorrect answers)
- What is output by the following C statement? (1 pt.)

```
printf("%d + %d + %d", 1, 2, 1+2);
```

a) `1 + 2 + 1+2`

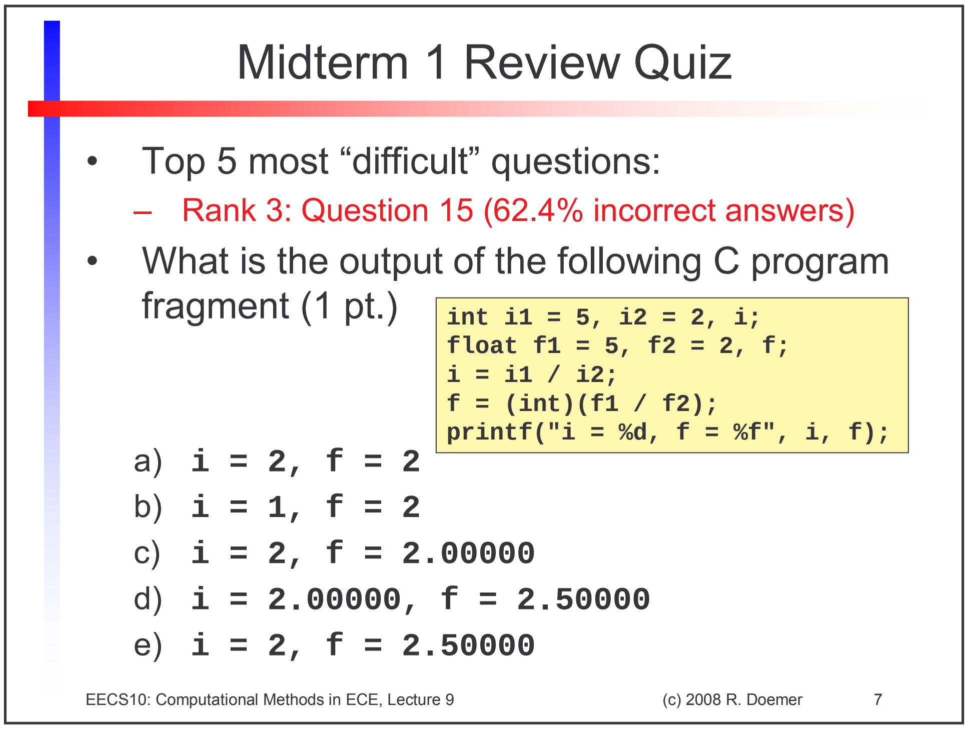
b) `%d + %d + %d, 1, 2, 1+2`

c) `6`

d) `%1 + %2 + %3`

[✓ e) `1 + 2 + 3`]

EECS10: Computational Methods in ECE, Lecture 9 (c) 2008 R. Doemer 6

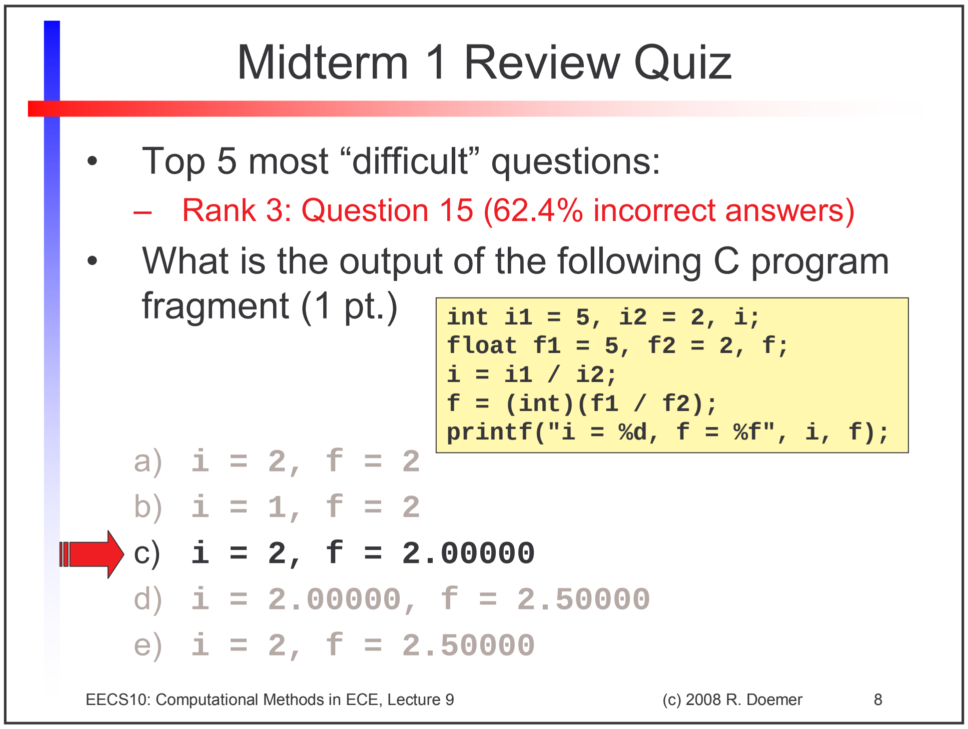


Midterm 1 Review Quiz

- Top 5 most “difficult” questions:
  - Rank 3: Question 15 (62.4% incorrect answers)
- What is the output of the following C program fragment (1 pt.)

```
int i1 = 5, i2 = 2, i;
float f1 = 5, f2 = 2, f;
i = i1 / i2;
f = (int)(f1 / f2);
printf("i = %d, f = %f", i, f);
```

a) `i = 2, f = 2`
b) `i = 1, f = 2`
c) `i = 2, f = 2.00000`
d) `i = 2.00000, f = 2.50000`
e) `i = 2, f = 2.50000`

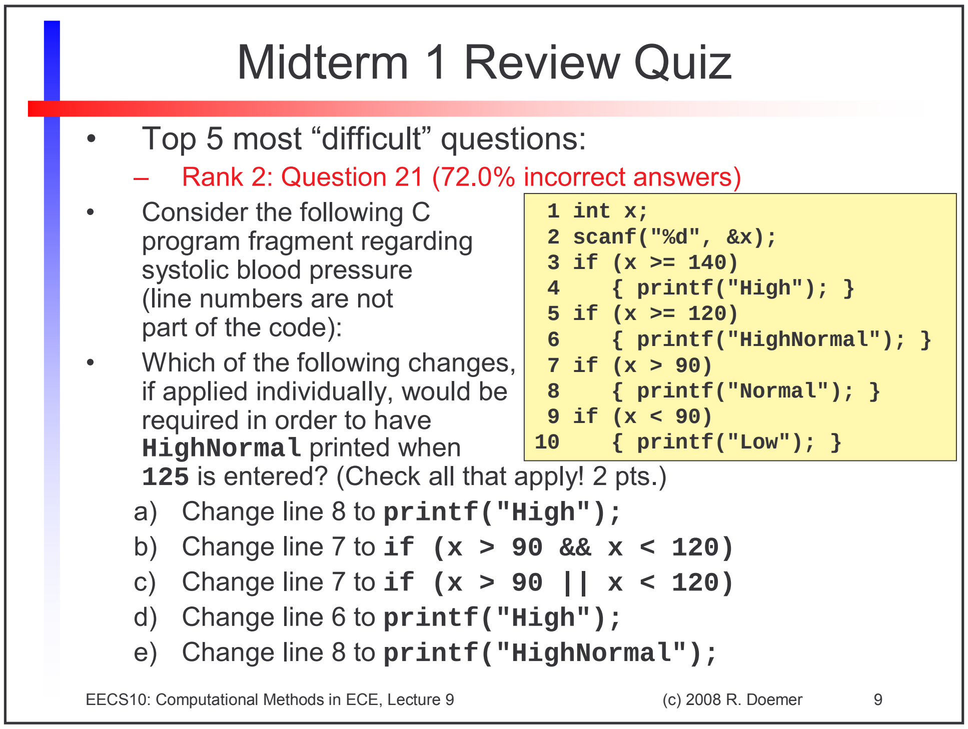


Midterm 1 Review Quiz

- Top 5 most “difficult” questions:
  - Rank 3: Question 15 (62.4% incorrect answers)
- What is the output of the following C program fragment (1 pt.)

```
int i1 = 5, i2 = 2, i;
float f1 = 5, f2 = 2, f;
i = i1 / i2;
f = (int)(f1 / f2);
printf("i = %d, f = %f", i, f);
```

a) `i = 2, f = 2`
b) `i = 1, f = 2`
[✓ c) `i = 2, f = 2.00000`]
d) `i = 2.00000, f = 2.50000`
e) `i = 2, f = 2.50000`



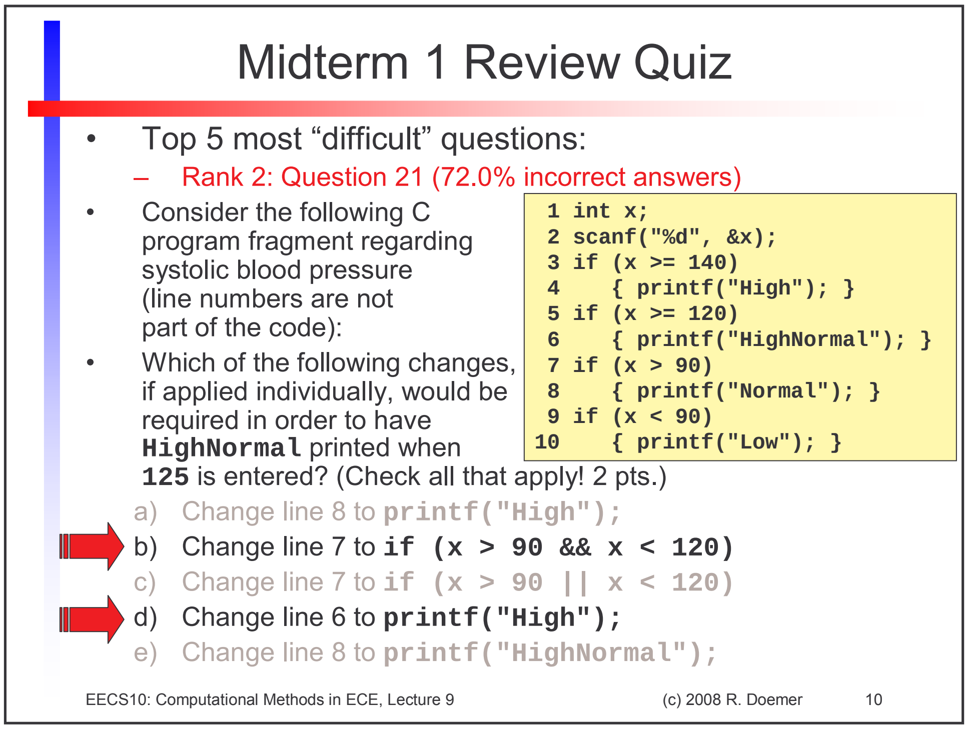
Midterm 1 Review Quiz

- Top 5 most “difficult” questions:
  - Rank 2: Question 21 (72.0% incorrect answers)
- Consider the following C program fragment regarding systolic blood pressure (line numbers are not part of the code):

```
 1 int x;
 2 scanf("%d", &x);
 3 if (x >= 140)
 4    { printf("High"); }
 5 if (x >= 120)
 6    { printf("HighNormal"); }
 7 if (x > 90)
 8    { printf("Normal"); }
 9 if (x < 90)
10    { printf("Low"); }
```

- Which of the following changes, if applied individually, would be required in order to have **`HighNormal`** printed when **`125`** is entered? (Check all that apply! 2 pts.)

a) Change line 8 to **`printf("High");`**
b) Change line 7 to **`if (x > 90 && x < 120)`**
c) Change line 7 to **`if (x > 90 || x < 120)`**
d) Change line 6 to **`printf("High");`**
e) Change line 8 to **`printf("HighNormal");`**



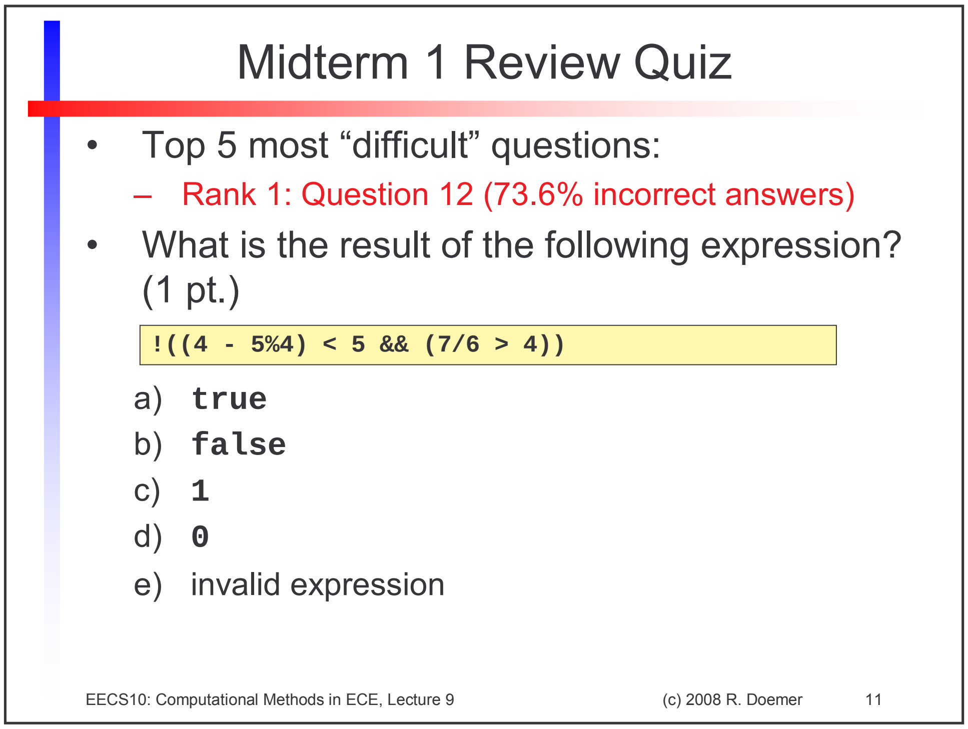
Midterm 1 Review Quiz

- Top 5 most “difficult” questions:
  - Rank 2: Question 21 (72.0% incorrect answers)
- Consider the following C program fragment regarding systolic blood pressure (line numbers are not part of the code):

```
 1 int x;
 2 scanf("%d", &x);
 3 if (x >= 140)
 4    { printf("High"); }
 5 if (x >= 120)
 6    { printf("HighNormal"); }
 7 if (x > 90)
 8    { printf("Normal"); }
 9 if (x < 90)
10    { printf("Low"); }
```

- Which of the following changes, if applied individually, would be required in order to have **`HighNormal`** printed when **`125`** is entered? (Check all that apply! 2 pts.)

a) Change line 8 to **`printf("High");`**
[✓ b) Change line 7 to **`if (x > 90 && x < 120)`**]
c) Change line 7 to **`if (x > 90 || x < 120)`**
[✓ d) Change line 6 to **`printf("High");`**]
e) Change line 8 to **`printf("HighNormal");`**

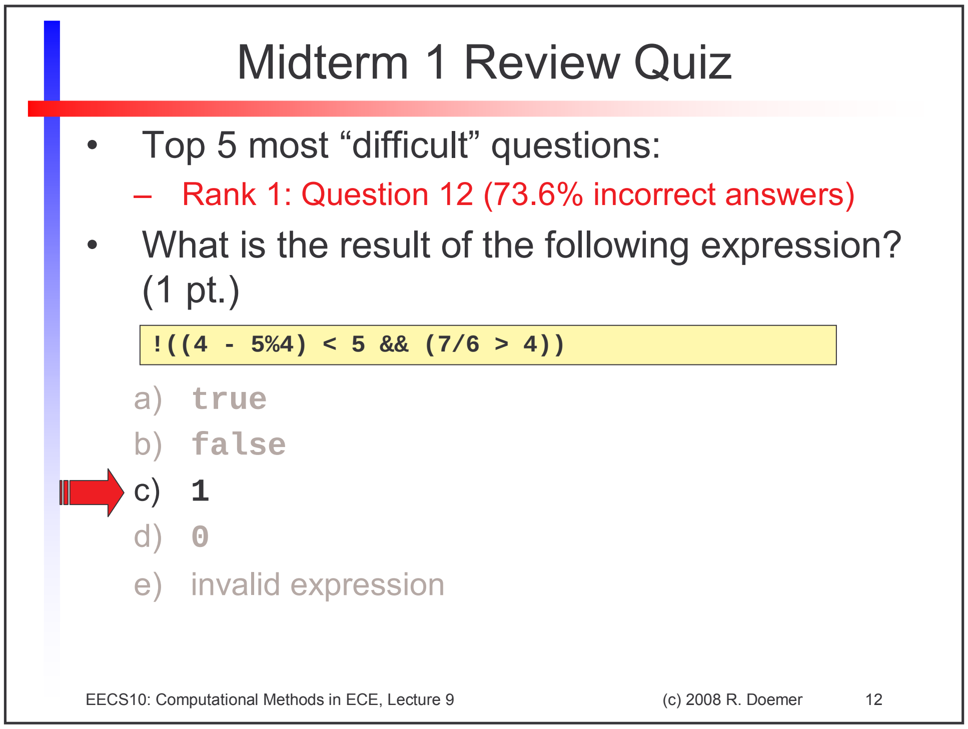


Midterm 1 Review Quiz

- Top 5 most “difficult” questions:
  - Rank 1: Question 12 (73.6% incorrect answers)
- What is the result of the following expression? (1 pt.)

```
!((4 - 5%4) < 5 && (7/6 > 4))
```

a) `true`
b) `false`
c) `1`
d) `0`
e) invalid expression

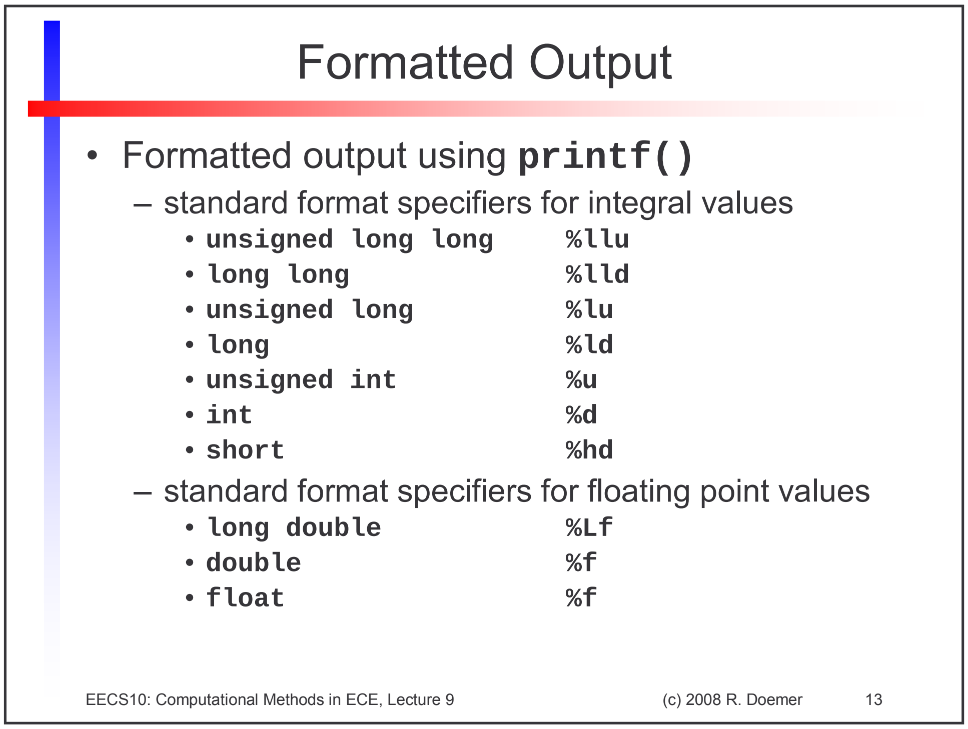


Midterm 1 Review Quiz

- Top 5 most “difficult” questions:
  - Rank 1: Question 12 (73.6% incorrect answers)
- What is the result of the following expression? (1 pt.)

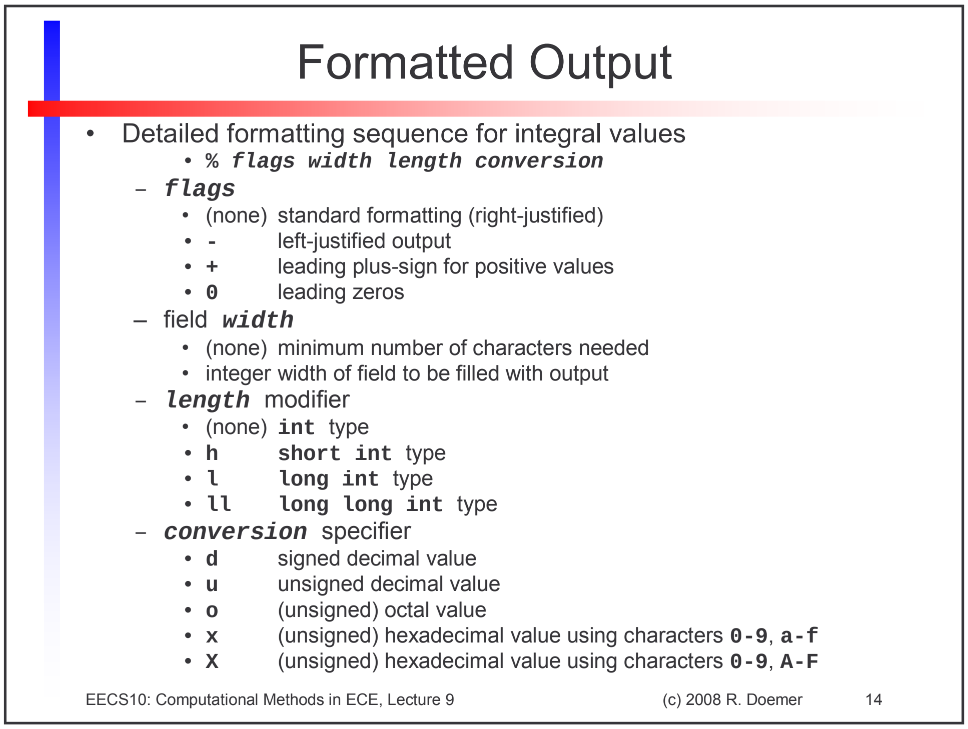
```
!((4 - 5%4) < 5 && (7/6 > 4))
```

a) `true`
b) `false`
[✓ c) `1`]
d) `0`
e) invalid expression



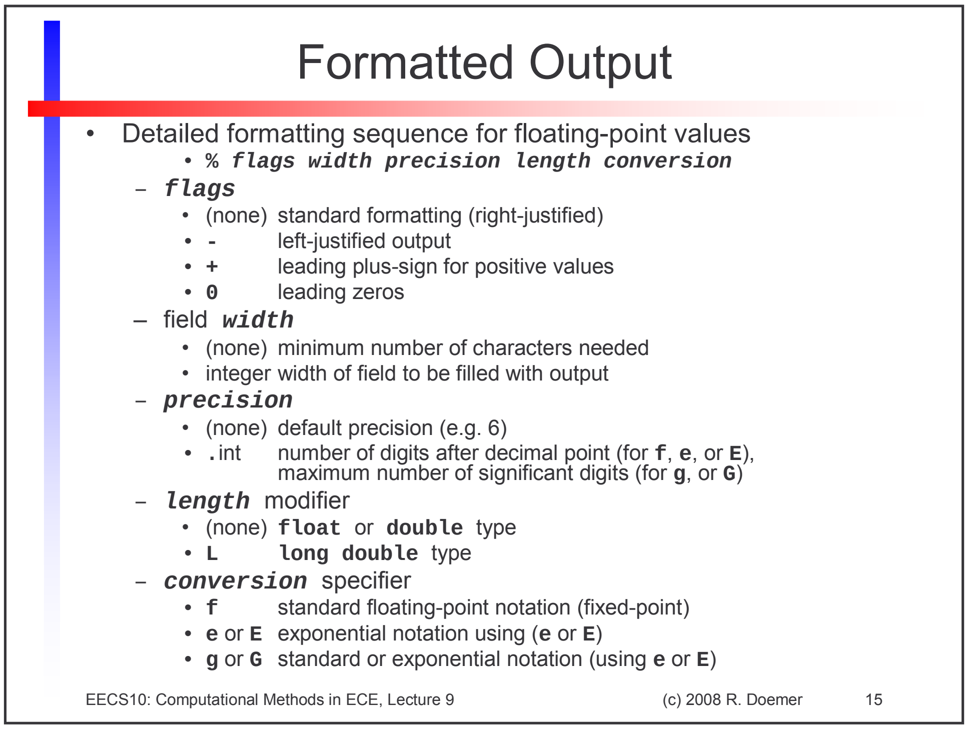
Formatted Output

- 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`**



Formatted Output

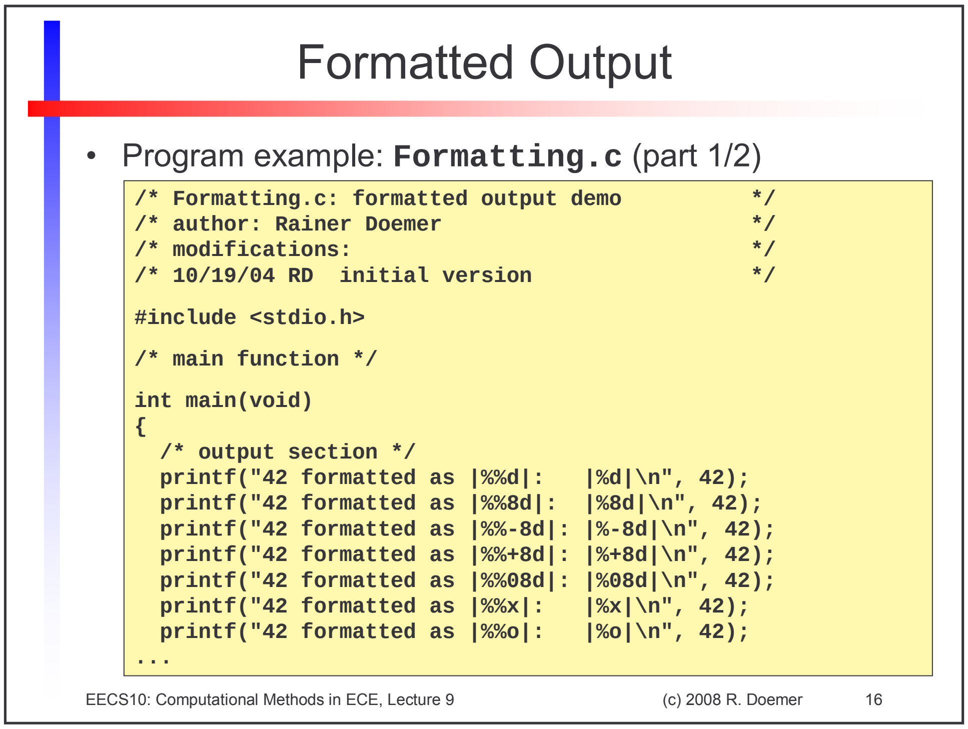
- Detailed formatting sequence for integral values
    - **`%`** ***`flags width length conversion`***
  - ***`flags`***
    - (none) standard formatting (right-justified)
    - **`-`** left-justified output
    - **`+`** leading plus-sign for positive values
    - **`0`** leading zeros
  - field ***`width`***
    - (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`**



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) 2008 R. Doemer 15



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) 2008 R. Doemer 16

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) 2008 R. Doemer 17

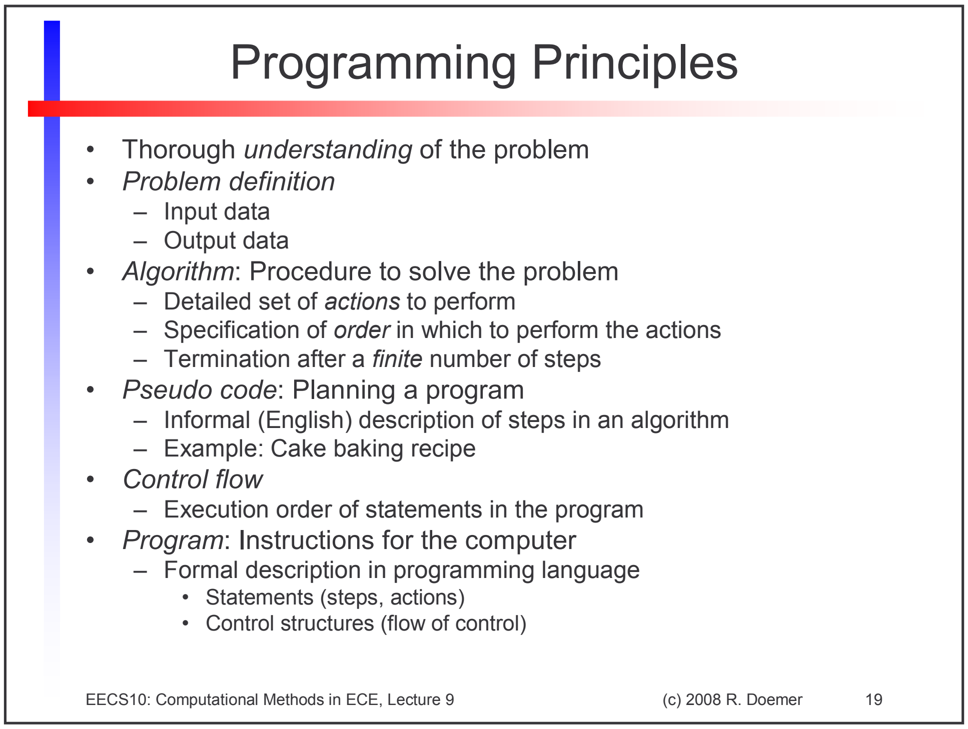
Formatted Output

- 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|
%
```

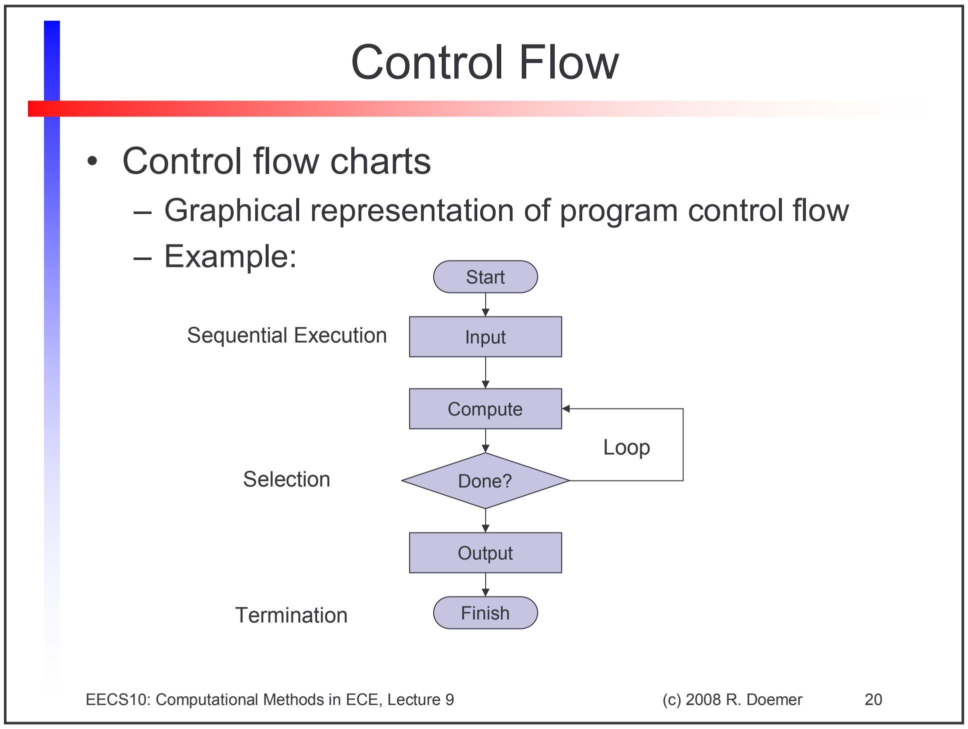
EECS10: Computational Methods in ECE, Lecture 9 (c) 2008 R. Doemer 18



Programming Principles

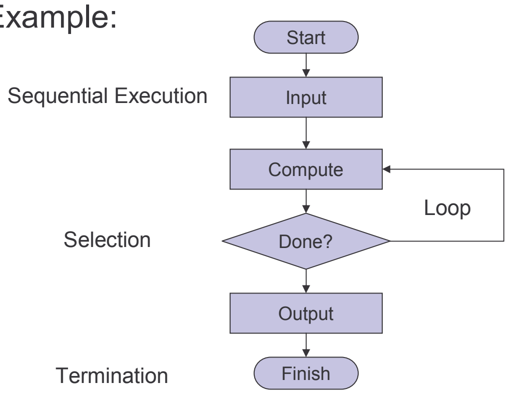
- Thorough *understanding* of the problem
- *Problem definition*
  - Input data
  - Output data
- *Algorithm*: Procedure to solve the problem
  - Detailed set of *actions* to perform
  - Specification of *order* in which to perform the actions
  - Termination after a *finite* number of steps
- *Pseudo 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) 2008 R. Doemer 19



Control Flow

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



EECS10: Computational Methods in ECE, Lecture 9 (c) 2008 R. Doemer 20