

EECS 10: Computational Methods in Electrical and Computer Engineering

Lecture 10

Rainer Dömer

doemer@uci.edu

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

Lecture 10: Overview

- Midterm 1 Review Quiz
 - Top 5 most “difficult” questions
- Structured Programming
 - Control flow charts
 - Sequential statements
 - Conditional statements
 - `if` statement
 - `if-else` statement
 - `switch` statement
 - Structured Program Composition
 - Example `Grade.c`
 - Example `Grade2.c`

Midterm 1 Review Quiz

- Top 5 most “difficult” questions:
 - Rank 5: Question 6 (64% wrong answers)
- Which of the following are valid keywords in C?
(Check all that apply! 2 pts.)
 - a) `return`
 - b) `main`
 - c) `integer`
 - d) `keyword`
 - e) `short`

EECS10: Computational Methods in ECE, Lecture 10

(c) 2006 R. Doemer

3

Midterm 1 Review Quiz

- Top 5 most “difficult” questions:
 - Rank 5: Question 6 (64% wrong answers)
- Which of the following are valid keywords in C?
(Check all that apply! 2 pts.)
 - a) `return`
 - b) `main`
 - c) `integer`
 - d) `keyword`
 - e) `short`

EECS10: Computational Methods in ECE, Lecture 10

(c) 2006 R. Doemer

4

Midterm 1 Review Quiz

- Top 5 most “difficult” questions:
 - Rank 4: Question 12 (65% wrong answers)
- What is output by the following C statement?


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

- a) 1 + 2 + 3
- b) %1 + %2 + %3
- c) %d + %d + %d,1,2,1+2
- d) 1 + 2 + 1+2
- e) 123

Midterm 1 Review Quiz

- Top 5 most “difficult” questions:
 - Rank 4: Question 12 (65% wrong answers)
- What is output by the following C statement?

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

-  a) 1 + 2 + 3
- b) %1 + %2 + %3
- c) %d + %d + %d,1,2,1+2
- d) 1 + 2 + 1+2
- e) 123

Midterm 1 Review Quiz

- Top 5 most “difficult” questions:
 - Rank 3: Question 9 (75% wrong answers)
- Which of the following statements is true about data types in ANSI-C?
(Check all that apply!)
 - a) `float` can store a greater value than `long int`
 - b) `int` has a larger range than `char`
 - c) `float` has a higher precision than `double`
 - d) `long` has a smaller range than `unsigned int`
 - e) `char` can store a smaller value than `unsigned int`

EECS10: Computational Methods in ECE, Lecture 10

(c) 2006 R. Doemer

7

Midterm 1 Review Quiz

- Top 5 most “difficult” questions:
 - Rank 3: Question 9 (75% wrong answers)
- Which of the following statements is true about data types in ANSI-C?
(Check all that apply!)
 - a) `float` can store a greater value than `long int`
 - b) `int` has a larger range than `char`
 - c) `float` has a higher precision than `double`
 - d) `long` has a smaller range than `unsigned int`
 - e) `char` can store a smaller value than `unsigned int`

EECS10: Computational Methods in ECE, Lecture 10

(c) 2006 R. Doemer

8

Midterm 1 Review Quiz



- Top 5 most “difficult” questions:
 - Rank 2: Question 14 (80% wrong answers)
- Which of the following expressions yield a result type of `double`?
(Check all that apply!)
 - a) `1.0 / 3`
 - b) `10 / 3`
 - c) `3 * 1000000`
 - d) `(int)3.3 > 3.0`
 - e) `3 * 100.00`

EECS10: Computational Methods in ECE, Lecture 10

(c) 2006 R. Doemer

9

Midterm 1 Review Quiz

- Top 5 most “difficult” questions:
 - Rank 2: Question 14 (80% wrong answers)
- Which of the following expressions yield a result type of `double`?
(Check all that apply!)
 -  a) `1.0 / 3`
 - b) `10 / 3`
 - c) `3 * 1000000`
 - d) `(int)3.3 > 3.0`
 -  e) `3 * 100.00`

EECS10: Computational Methods in ECE, Lecture 10

(c) 2006 R. Doemer

10

Midterm 1 Review Quiz

- Top 5 most “difficult” questions:
 - Top Rank: Question 8 (88% wrong answers)
- Which of the following constructs denotes a valid type name in C?
(Check all that apply! 2 pts.)
 - a) `signed char`
 - b) `short double`
 - c) `long double`
 - d) `short char`
 - e) `unsigned long long int`

EECS10: Computational Methods in ECE, Lecture 10

(c) 2006 R. Doemer

11

Midterm 1 Review Quiz

- Top 5 most “difficult” questions:
 - Top Rank: Question 8 (88% wrong answers)
- Which of the following constructs denotes a valid type name in C?
(Check all that apply! 2 pts.)
 - a) `signed char`
 - b) `short double`
 - c) `long double`
 - d) `short char`
 - e) `unsigned long long int`

EECS10: Computational Methods in ECE, Lecture 10

(c) 2006 R. Doemer

12

Structured Programming

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

```

graph TD
    Start([Start]) --> Input[Input]
    Input --> Compute[Compute]
    Compute --> Done{Done?}
    Done -- Loop --> Compute
    Done --> Output[Output]
    Output --> Finish([Finish])
    
```

EECS10: Computational Methods in ECE, Lecture 10 (c) 2006 R. Doemer 13

Structured Programming

- Sequential execution in C
 - Statement blocks: *Compound statements*
 - Sequence of statements grouped by braces: { }
- Example:

```

{
  /* statement 1 */

  /* statement 2 */

  /* statement 3 */

  /* ... */

  /* statement n */
}
                
```

Flow chart:

```

graph TD
    Start(( )) --> S1[Statement 1]
    S1 --> S2[Statement 2]
    S2 --> S3[Statement 3]
    S3 -.-> S4[Statement n]
    S4 --> End(( ))
    
```

EECS10: Computational Methods in ECE, Lecture 10 (c) 2006 R. Doemer 14

Structured Programming

- Sequential execution in C
 - Statement blocks: *Compound statements*
 - Sequence of statements grouped by braces: { }
- *Indentation* increases readability of the code
 - proper indentation is highly recommended!
- Example:

```

/* some statements... */
if (x < 0) {
    printf("%d is negative!", x);
    /* handle negative values of x... */
    if (x < 100) {
        printf("%d is too small!", x);
        /* handle the problem... */
    } /* fi */
} /* fi */
if (x > 0) {
    printf("%d is positive!", x);
    /* handle positive values of x... */
} /* fi */
/* more statements... */

```

EECS10: Computational Methods in ECE, Lecture 10

(c) 2006 R. Doemer

15

Structured Programming

- Sequential execution in C
 - Statement blocks: *Compound statements*
 - Sequence of statements grouped by braces: { }
- *Indentation* increases readability of the code
 - proper indentation is highly recommended!
- Example:

```

/* some statements... */
if (x < 0) {
    printf("%d is negative!", x);
    /* handle negative values of x... */
    if (x < 100) {
        printf("%d is too small!", x);
        /* handle the problem... */
    } /* fi */
} /* fi */
if (x > 0) {
    printf("%d is positive!", x);
    /* handle positive values of x... */
} /* fi */
/* more statements... */

```

indentation level 0

indentation level 1 →

indentation level 2 → →

indentation level 1 →

indentation level 0

indentation level 1 →

indentation level 0

EECS10: Computational Methods in ECE, Lecture 10

(c) 2006 R. Doemer

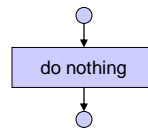
16

Structured Programming

- Empty statement blocks
 - empty compound statement
 - does nothing (no operation, no-op)
 - Example:

Flow chart:

```
{
  /* nothing */
}
```



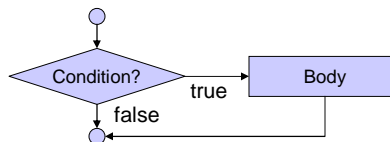
EECS10: Computational Methods in ECE, Lecture 10

(c) 2006 R. Doemer

17

Structured Programming

- Selection: **if** statement
 - Flow chart:



- Example:

```
if (grade >= 60)
{ printf("You passed.");
} /* fi */
```

EECS10: Computational Methods in ECE, Lecture 10

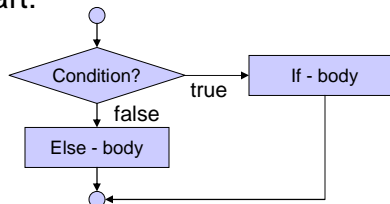
(c) 2006 R. Doemer

18

Structured Programming

- Selection: **if-else** statement

– Flow chart:



– Example:

```

if (grade >= 60)
{ printf("You passed.");
} /* fi */
else
{ printf("You failed.");
} /* esle */
  
```

EECS10: Computational Methods in ECE, Lecture 10

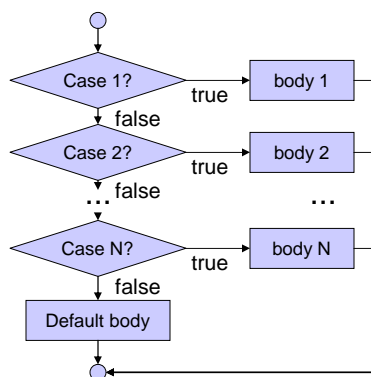
(c) 2006 R. Doemer

19

Structured Programming

- Selection: **switch** statement

– Flow chart:



Example:

```

switch(LetterGrade)
{ case 'A':
  { printf("Excellent!");
    break; }
  case 'B':
  case 'C':
  case 'D':
  { printf("Passed.");
    break; }
  case 'F':
  { printf("Failed!");
    break; }
  default:
  { printf("Invalid grade!");
    break; }
} /* hctiws */
  
```

EECS10: Computational Methods in ECE, Lecture 10

(c) 2006 R. Doemer

20

Structured Program Composition

- Initial flow chart
 - Start
 - Program body
 - Finish
- Statement sequences
 - Statement blocks can be concatenated
 - Sequential execution
- Nested control structures
 - control structures can be placed wherever statement blocks can be placed in the code

EECS10: Computational Methods in ECE, Lecture 10 (c) 2006 R. Doemer 21

Structured Program Composition

- Example:
 - Initial flow chart

EECS10: Computational Methods in ECE, Lecture 10 (c) 2006 R. Doemer 22

Structured Program Composition

- Example:
 - Sequential composition

```

    graph TD
      Start([Start]) --> P1[ ]
      P1 --> P2[ ]
      P2 --> End([End])
      subgraph Box [ ]
        P1
        P2
      end
    
```

EECS10: Computational Methods in ECE, Lecture 10 (c) 2006 R. Doemer 23

Structured Program Composition

- Example:
 - insertion of another sequential statement

```

    graph TD
      Start([Start]) --> P1[ ]
      P1 --> P2[ ]
      subgraph Box [ ]
        P1
        P2
      end
      Box --> P3[ ]
      P3 --> End([End])
    
```

EECS10: Computational Methods in ECE, Lecture 10 (c) 2006 R. Doemer 24

Structured Program Composition

- Example:
 - insertion of **if - else** statement

The flowchart shows a sequence of operations starting from a start node (oval) and ending at an end node (oval). The main flow consists of a start node, a process node (rectangle), a decision node (diamond), another process node, and a final process node. A dashed box highlights the decision node and the process node that follows it. An arrow from the decision node points to a process node to the right, which then loops back to the process node immediately following the decision node. This represents an if-else loop structure.

EECS10: Computational Methods in ECE, Lecture 10 (c) 2006 R. Doerner 25

Structured Program Composition

- Example:
 - insertion of sequential statement

The flowchart shows a sequence of operations starting from a start node (oval) and ending at an end node (oval). The main flow consists of a start node, a process node, a decision node, another process node, and a final process node. A dashed box highlights the decision node and two process nodes stacked vertically to its right. An arrow from the decision node points to the top process node, which then points to the bottom process node. This represents a sequential execution path within a conditional block.

EECS10: Computational Methods in ECE, Lecture 10 (c) 2006 R. Doerner 26

Structured Program Composition

- Example:
 - insertion of **if - else** statement

The flowchart shows a sequence of operations starting from a start node (oval), followed by a process node (rectangle), a decision node (diamond), another process node, and a final process node leading to an end node (oval). A dashed box highlights a section where a decision node is inserted. This decision node has two paths: one leading to a process node and another leading to another process node. Both paths then merge back into the main flow before the final process node.

EECS10: Computational Methods in ECE, Lecture 10 (c) 2006 R. Doemer 27

Structured Program Composition

- Example:
 - insertion of sequential statement

The flowchart shows a sequence of operations starting from a start node (oval), followed by a process node, a decision node, another process node, and a final process node leading to an end node (oval). A dashed box highlights a section where a sequential block of two process nodes is inserted. The flow enters the dashed box from the top, goes through the two process nodes in sequence, and then exits the dashed box to merge back into the main flow before the final process node.

EECS10: Computational Methods in ECE, Lecture 10 (c) 2006 R. Doemer 28

Structured Program Composition

- Example:
 - insertion of sequential statement (twice)

The flowchart shows a main vertical sequence of nodes: a start oval, a rectangle, a diamond, a rectangle, a diamond, a rectangle, a diamond, a rectangle, a diamond, a rectangle, and an end oval. A branch from the first diamond leads to a sequence of three rectangles. A second branch from the second diamond leads to a sequence of three rectangles, with the last one enclosed in a dashed box. Arrows indicate the flow from the main sequence to these branches and back.

EECS10: Computational Methods in ECE, Lecture 10 (c) 2006 R. Doemer 29

Structured Program Composition

- Example:
 - insertion of **switch** statement
 - etc. ...

The flowchart shows a main vertical sequence of nodes: a start oval, a rectangle, a diamond, a rectangle, a diamond, a rectangle, a diamond, a rectangle, a diamond, a rectangle, and an end oval. A branch from the first diamond leads to a sequence of three rectangles. A second branch from the second diamond leads to a sequence of three rectangles, with the last one enclosed in a dashed box. Below the main sequence, a dashed box encloses a switch statement structure consisting of three diamonds, each pointing to a rectangle, with arrows from the rectangles merging back into the main sequence.

EECS10: Computational Methods in ECE, Lecture 10 (c) 2006 R. Doemer 30

Example Program

- Grade calculation: `Grade.c` (part 1/3)

```

/* Grade.c: convert score into letter grade */
/* author: Rainer Doemer */
/* modifications: */
/* 10/17/04 RD initial version */

#include <stdio.h>

/* main function */

int main(void)
{
    /* variable definitions */
    int score = 0;
    char grade;

    /* input section */
    while (score < 1 || score > 100)
    { printf("Please enter your score (1-100): ");
      scanf("%d", &score);
    } /* elihw */

    ...

```

EECS10: Computational Methods in ECE, Lecture 10

(c) 2006 R. Doemer

31

Example Program

- Grade calculation: `Grade.c` (part 2/3)

```

...
/* computation section */
if (score >= 90)
    { grade = 'A'; }
else
    { if (score >= 80)
      { grade = 'B'; }
      else
        { if (score >= 70)
          { grade = 'C'; }
          else
            { if (score >= 60)
              { grade = 'D'; }
              else
                { grade = 'F'; }
            } /* esle */
          } /* esle */
        } /* esle */

...

```

EECS10: Computational Methods in ECE, Lecture 10

(c) 2006 R. Doemer

32

Example Program

- Grade calculation: `Grade.c` (part 3/3)

```
...
/* output section */
printf("Your letter grade is %c.\n", grade);

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

/* EOF */
```

Example Program

- Example session: `Grade.c`

```
% vi Grade.c
% gcc Grade.c -o Grade -Wall -ansi
% Grade
Please enter your score (1-100): 111
Please enter your score (1-100): 99
Your letter grade is A.
% Grade
Please enter your score (1-100): 85
Your letter grade is B.
% Grade
Please enter your score (1-100): 71
Your letter grade is C.
% Grade
Please enter your score (1-100): 69
Your letter grade is D.
% Grade
Please enter your score (1-100): 55
Your letter grade is F.
%
```

Example Program

- Grade calculation: `Grade2.c` (part 1/3)

```

/* Grade2.c: convert score into letter grade */
/* author: Rainer Doemer */
/* modifications: */
/* 10/18/04 RD use 'switch' statement */
/* 10/17/04 RD initial version */

#include <stdio.h>

/* main function */
int main(void)
{
    /* variable definitions */
    int score = 0;
    char grade;

    /* input section */
    while (score < 1 || score > 100)
    { printf("Please enter your score (1-100): ");
      scanf("%d", &score);
    } /* elihw */

```

EECS ...

Example Program

- Grade calculation: `Grade2.c` (part 2/3)

```

.../* computation section */
switch (score / 10)
{ case 10:
  case 9:
    { grade = 'A';
      break; }
  case 8:
    { grade = 'B';
      break; }
  case 7:
    { grade = 'C';
      break; }
  case 6:
    { grade = 'D';
      break; }
  default:
    { grade = 'F';
      break; }
} /* hctiws */

```

EECS ...

Example Program

- Grade calculation: `Grade2.c` (part 3/3)

```
...  
  
/* output section */  
printf("Your letter grade is %c.\n", grade);  
  
/* exit */  
return 0;  
} /* end of main */  
  
/* EOF */
```

Example Program

- Example session: `Grade2.c`

```
% cp Grade.c Grade2.c  
% vi Grade2.c  
% gcc Grade2.c -o Grade2 -Wall -ansi  
% Grade2  
Please enter your score (1-100): 111  
Please enter your score (1-100): 99  
Your letter grade is A.  
% Grade2  
Please enter your score (1-100): 85  
Your letter grade is B.  
% Grade2  
Please enter your score (1-100): 71  
Your letter grade is C.  
% Grade2  
Please enter your score (1-100): 69  
Your letter grade is D.  
% Grade2  
Please enter your score (1-100): 55  
Your letter grade is F.  
%
```