EECS 10: Computational Methods in Electrical and Computer Engineering

Lecture 13

Rainer Dömer

doemer@uci.edu

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

Lecture 13: Overview

- Warm-up Quiz
  - Revisiting “difficult” midterm questions
- Course Administration
  - Midterm course evaluation
- Functions
  - Introduction
  - Terms and concepts
  - Example **`Square.c`**

Quiz: Question 1

- What is output by the following C statement?

```
printf("20%d + 30%d", 02, 03);
```

a) **2002 + 2003**
b) **20% + 30%**
c) **5005**
d) **202 + 303**
e) **20%d + 30%d, 02, 03**

Quiz: Question 1

- What is output by the following C statement?

```
printf("20%d + 30%d", 02, 03);
```

a) **2002 + 2003**
b) **20% + 30%**
c) **5005**
d) **202 + 303**
e) **20%d + 30%d, 02, 03**

Quiz: Question 2

- Assume that **x** is an integer in the range of 1 through 10 inclusively. Which of the following expressions can be used as a test for **x** being an odd number?
  (Check all that apply!)
  a) `x % 2`
  b) `x / 2 * 2 == x`
  c) `x % 2 == 1`
  d) `x==1 || x==3 || x==5 || x==7 || x==9`
  e) `x % 2 == 0`

Quiz: Question 2

- Assume that **x** is an integer in the range of 1 through 10 inclusively. Which of the following expressions can be used as a test for **x** being an odd number?
  (Check all that apply!)
  a) `x % 2`
  b) `x / 2 * 2 == x`
  c) `x % 2 == 1`
  d) `x==1 || x==3 || x==5 || x==7 || x==9`
  e) `x % 2 == 0`

Quiz: Question 3

- What needs to be changed in the program below in order to have **B** printed when **85** is entered?
  a) Change line 8 to **`printf("B");`**
  b) Delete line 7
  c) Change line 7 to **`if (x == 85)`**
  d) Change line 5 to **`if (x > 80 && x <= 90)`**
  e) Change line 7 to **`if (x > 70 && x <= 80)`**

```
1 int x;
2 scanf("%d", &x);
3 if (x > 90)
4   { printf("A"); }
5 if (x > 80)
6   { printf("B"); }
7 if (x > 70)
8   { printf("C"); }
```

EECS10: Computational Methods in ECE, Lecture 13 (c) 2004 R. Doemer 7

Quiz: Question 3

- What needs to be changed in the program below in order to have **B** printed when **85** is entered?
  a) Change line 8 to **`printf("B");`**
  b) Delete line 7
  c) Change line 7 to **`if (x == 85)`**
  d) Change line 5 to **`if (x > 80 && x <= 90)`**
  e) Change line 7 to **`if (x > 70 && x <= 80)`**

```
1 int x;
2 scanf("%d", &x);
3 if (x > 90)
4   { printf("A"); }
5 if (x > 80)
6   { printf("B"); }
7 if (x > 70)
8   { printf("C"); }
```

EECS10: Computational Methods in ECE, Lecture 13 (c) 2004 R. Doemer 8

Quiz: Question 4

- Which of the following expressions would be treated as a true condition when used with an **`if`** statement?
  (Check all that apply!)

a) `(int)5.5 > 5`
b) `1 || 0 && 0`
c) `5 == 5`
d) `(1 + 2 + 3) == (3 << 2 >> 1)`
e) `5 - 5`

Quiz: Question 4

- Which of the following expressions would be treated as a true condition when used with an **`if`** statement?
  (Check all that apply!)

a) `(int)5.5 > 5`
b) `1 || 0 && 0`
c) `5 == 5`
d) `(1 + 2 + 3) == (3 << 2 >> 1)`
e) `5 - 5`

Quiz: Question 5

- Which of the following names are valid keywords in C? (Check all that apply!)

a) **`short`**
b) **`continue`**
c) **`printf`**
d) **`return`**
e) **`Break`**

Quiz: Question 5

- Which of the following names are valid keywords in C? (Check all that apply!)

a) **`short`**
b) **`continue`**
c) **`printf`**
d) **`return`**
e) **`Break`**

Quiz: Question 6

- If **`cnt`** is an integer counter, how could one update the value of **`cnt`**?
  (Check all that apply!)

a) `cnt += 1;`
b) `++cnt;`
c) `cnt++;`
d) `cnt += cnt;`
e) `cnt = cnt + 1;`

Quiz: Question 6

- If **`cnt`** is an integer counter, how could one update the value of **`cnt`**?
  (Check all that apply!)

a) `cnt += 1;`
b) `++cnt;`
c) `cnt++;`
d) `cnt += cnt;`
e) `cnt = cnt + 1;`

Course Administration

- Midterm Course Evaluation
  - One week, starting today!
  - Oct 27, 2004, 10am through Nov. 3, 2004, 10am
  - Online via EEE Evaluation application
- Feedback from students to instructors
  - Completely voluntary
  - Completely anonymous
  - Very valuable
    - Help to improve this class!
- Final Course Evaluation
  - expected for week 10 (TBA)

Functions

- Introduction
  - Programming concepts
    - Hierarchy
    - Encapsulation
    - Information hiding
    - Divide and conquer
  - Software reuse
    - Don't re-invent the wheel!
  - Program composition
    - C program = Set of functions
      - starting point: function named **`main`**
    - Libraries = Set of functions
      - predefined functions (possibly written by somebody else)

Functions

- Terms and Concepts
  - Function declaration
    - function prototype with argument and return types
  - Function definition
    - definition of function body
  - Function call
    - expression invoking a function
  - Function arguments
    - arguments supplied in a function call
  - Function parameters
    - formal parameters holding the data supplied to a function
  - Return value
    - output data computed by the function
  - Local variables
    - variables used locally in a function (temporary variables)

Functions

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

```
/* Square.c: example demonstrating functions    */
/*                                              */
/* author: Rainer Doemer                        */
/*                                              */
/* modifications:                               */
/* 10/27/04 RD  initial version                 */

#include <stdio.h>

/* function declaration */

double square(double a);

/* function definition */

double square(double a)
{
    return a * a;
} /* end of square */

...
```

Functions

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

```
...
/* main function */

int main(void)
{  /* variable definitions */
    double x, y;

    /* input section */
    printf("Please enter a value for x: ");
    scanf("%lf", &x);

    /* computation section */
    y = square(x);

    /* output section */
    printf("The square of %g is %g\n", x, y);

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

/* EOF */
```


Example Program

- Example session: **`Square.c`**

```
% vi Square.c
% gcc Square.c -o Square -Wall -ansi
% Square
Please enter a value for x: 3
The square of 3 is 9
% Square
Please enter a value for x: 5.5
The square of 5.5 is 30.25
%
```