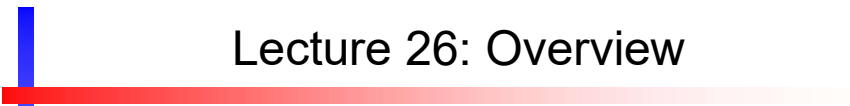# EECS 22: Advanced C Programming
Lecture 26

Rainer Dömer

doemer@uci.edu

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

Lecture 26: Overview

- Course Administration
  - Reminder: Final course evaluation
- Programming Courses in EECS
  - Outlook to EECS 22L

- Review Quiz

Course Administration

- Final Course Evaluation
  - Open until end of 10th week (Sunday night)
  - Nov. 14, 2016, through Dec. 4, 2016, 11:45pm
  - Online via EEE Evaluation application
- Mandatory Evaluation of Course and Instructor
  - Voluntary
  - Anonymous
  - Very valuable
- Please spend 5 minutes for this survey!
  - Your feedback is appreciated!

EECS22: Advanced C Programming, Lecture 26   (c) 2016 R. Doemer   3

Programming Courses in EECS

- Introductory Programming
  - EECS 10:   uses C programming language (for EE)
  - EECS 12:   uses Python programming language (for CpE)
- Programming from the Ground Up
  - EECS 20:   starts with Assembly language (on bare CPU),
                        then introduces C programming language
- Advanced Programming Courses
  - EECS 22:   *"Advanced C Programming"* (in ANSI C)
  - ➢ EECS 22L: *"Software Engineering Project in C"* (ANSI C/C++)
- Object-Oriented Programming
  - EECS 40:   introduces objects and classes, hierarchy,
                        and higher object-oriented programming concepts
                        using Java

EECS22: Advanced C Programming, Lecture 26   (c) 2016 R. Doemer   4

EECS 22L: Software Eng. Project in C

- *"Developing real C Programs in a Team"*
  - Hands-on experience with larger software projects
  - Introduction to software engineering
    - Specification, documentation, implementation, testing
  - Team work
- Features
  - Design efficient data structures, APIs
  - Utilize programming modules, build libraries, GUIs
  - Develop and optimize contemporary software applications
- Tools
  - Software development, version control: `ssh`, `gcc`, `cvs`, `chmod`
  - Compilation, scripting, packaging: `make`, `bash`, `groff`, `gtar`
  - Testing and debugging with `gdb`, `ddd`, `gprof`, `gcov`, …

EECS22: Advanced C Programming, Lecture 26   (c) 2016 R. Doemer   5

EECS 22L: Software Eng. Project in C

- Catalog Data
  - **EECS 22L Software Engineering Project in C Language (Credit Units: 3) W.**
  - Hands-on experience with the ANSI-C programming language.
  - Medium-sized programming projects, team work.
  - Software specification, documentation, implementation, testing.
  - Definition of data structures and application programming interface.
  - Creation of program modules, linking with external libraries.
  - Rule-based compilation, version control.
  - Prerequisites: EECS 22
  - (Design Units: 3)

EECS22: Advanced C Programming, Lecture 26   (c) 2016 R. Doemer   6

EECS 22L: Software Eng. Project in C

- Course Contents
  - Software engineering topics, including specification, documentation, implementation, testing, debugging, project planning, organization, maintenance, version control, organization of source files, header files, modules
  - Compilation flow, Makefile, shell scripting
  - Definition of data structures and application programming interface
  - External libraries, system programming, POSIX API, interrupts
  - Introduction to C++ language, syntax and semantics, references, inline functions, default arguments, classes, members, and methods, object creation and deletion (constructors, destructors)

EECS22: Advanced C Programming, Lecture 26   (c) 2016 R. Doemer   7

Review Quiz: Question 1

- Which of the following statements is true for an *algorithm*?
  (Check all that apply!)

  a) An algorithm must be indeterministic.
  b) An algorithm solves a problem quickly.
  c) An algorithm is historically based on Al Gore's rythm.
  d) An algorithm executes a program using pseudo code.
  e) An algorithm must terminate after a finite number of steps.

EECS22: Advanced C Programming, Lecture 26   (c) 2016 R. Doemer   8

Review Quiz: Question 1

- Which of the following statements is true for an *algorithm*?
  (Check all that apply!)

  a) An algorithm must be indeterministic.
  b) An algorithm solves a problem quickly.
  c) An algorithm is historically based on Al Gore's rythm.
  d) An algorithm executes a program using pseudo code.
  e) An algorithm must terminate after a finite number of steps.

Review Quiz: Question 2

- In C, which properties does every object have?
  (Check all that apply!)

  a) A size.
  b) A value.
  c) A weight.
  d) A type.
  e) A location.

Review Quiz: Question 2

- In C, which properties does every object have?
  (Check all that apply!)

a) A size.
b) A value.
c) A weight.
d) A type.
e) A location.

Review Quiz: Question 3

- What is the result type of the following expression?

```
-1 + 2.3f * (4.5 / 67f) - (short)89
```

a) `short int`
b) `int`
c) `long int`
d) `float`
e) `double`

Review Quiz: Question 3

- What is the result type of the following expression?

```
-1 + 2.3f * (4.5 / 67f) - (short)89
```

a) short int
b) int
c) long int
d) float
→ e) **double**

EECS22: Advanced C Programming, Lecture 26   (c) 2016 R. Doemer   13

Review Quiz: Question 4

- Given the following code fragment,

```
double x;
double y;

x = (int)(y + 0.5);
```

which of the following statements is true? (Check all that apply!)

a) for **y=5.0**, **x** is set to **5.0**
b) for **y=5.1**, **x** is set to **5.0**
c) for **y=5.49**, **x** is set to **5.0**
d) for **y=5.5**, **x** is set to **6.0**
e) for **y=5.95**, **x** is set to **6.0**

EECS22: Advanced C Programming, Lecture 26   (c) 2016 R. Doemer   14

Review Quiz: Question 4

- Given the following code fragment,

```
double x;
double y;

x = (int)(y + 0.5);
```

  which of the following statements is true?
  (Check all that apply!)

  a) for **y=5.0**,   **x** is set to **5.0**
  b) for **y=5.1**,   **x** is set to **5.0**
  c) for **y=5.49**, **x** is set to **5.0**
  d) for **y=5.5**,   **x** is set to **6.0**
  e) for **y=5.95**, **x** is set to **6.0**

Review Quiz: Question 5

- What is the value of **x** after the following code fragment is executed?

```
int x = 10;
while(x > 0)
   { x -= 2;
    }
```

  a) **-2**
  b) **-1**
  c) **0**
  d) **1**
  e) **2**

Review Quiz: Question 5

- What is the value of **x** after the following code fragment is executed?

```
int x = 10;
while(x > 0)
   { x -= 2;
    }
```

    a) -2
    b) -1
    c) 0
    d) 1
    e) 2

Review Quiz: Question 6

- Given that the C standard math library is included, which of the following expressions results in the value **4.0**?
  (Check all that apply!)

    a) **pow(16.0, .5)**
    b) **4.0 * cos(0.0)**
    c) **3 + sin(0.0)**
    d) **log10(10000.00)**
    e) **sqrt(15.0) + 1**

Review Quiz: Question 6

- Given that the C standard math library is included, which of the following expressions results in the value **4.0**?
  (Check all that apply!)

a) **pow(16.0, .5)**
b) **4.0 * cos(0.0)**
c) 3 + sin(0.0)
d) **log10(10000.00)**
e) sqrt(15.0) + 1

Review Quiz: Question 7

- What is output by the following program fragment?

```
char s[] = "EECS22";

s[4] = 0;
printf("%s %c", s, s[2]);
```

a) **EECS02 2**
b) **EEC 22 0**
c) **E E**
d) **EECS C**
e) **EEC C**

Review Quiz: Question 7

- What is output by the following program fragment?

```
char s[] = "EECS22";

s[4] = 0;
printf("%s %c", s, s[2]);
```

a) EECS02 2
b) EEC 22 0
c) E E
d) **EECS C**
e) EEC C

Review Quiz: Question 8

- In the program below, what is printed by the function call **g(1)**?

a) 1 2
b) 2 3
c) 1 1
d) 2
e) 1

```
1  int f(int x)
2  { printf("%d ", x);
3     return x + 1;
4  }
5  int g(int x)
6  { printf("%d ", f(x));
7     return x + 2;
8  }
```

Review Quiz: Question 8

- In the program below, what is printed by the function call **g(1)**?

a) **1 2**

b) 2 3

c) 1 1

d) 2

e) 1

```
1  int f(int x)
2  { printf("%d ", x);
3      return x + 1;
4  }
5  int g(int x)
6  { printf("%d ", f(x));
7      return x + 2;
8  }
```

EECS22: Advanced C Programming, Lecture 26   (c) 2016 R. Doemer   23

Review Quiz: Question 9

- What is recursion?
  (Check all that apply!)

a) A function that does not terminate.

b) A function that calls itself.

c) A function that contains a loop.

d) A function **f** that calls a function **g** which calls **f**.

e) A function that returns no value.

EECS22: Advanced C Programming, Lecture 26   (c) 2016 R. Doemer   24

Review Quiz: Question 9

- What is recursion?
  (Check all that apply!)

    a) A function that does not terminate.
    b) A function that calls itself.
    c) A function that contains a loop.
    d) A function **f** that calls a function **g** which calls **f**.
    e) A function that returns no value.

EECS22: Advanced C Programming, Lecture 26   (c) 2016 R. Doemer   25

Review Quiz: Question 10

- Given the function definition below, what is printed for the function call **f(3)**?

    a) 1 2 3

    b) 1 2 3 4

    c) 3 2 1 0

    d) 4 3 2 1

    e) 3 2 1

```
1  void f(int x)
2  {
3    printf("%d ", x);
4    if (x > 0)
5      { f(x-1); }
6  }
```

EECS22: Advanced C Programming, Lecture 26   (c) 2016 R. Doemer   26

Review Quiz: Question 10

- Given the function definition below, what is printed for the function call **f(3)** ?

a) 1 2 3

b) 1 2 3 4

➡ c) **3 2 1 0**

d) 4 3 2 1

e) 3 2 1

```
1  void f(int x)
2  {
3    printf("%d ", x);
4    if (x > 0)
5      { f(x-1); }
6  }
```

EECS22: Advanced C Programming, Lecture 26   (c) 2016 R. Doemer   27

Review Quiz: Question 11

- Given the following definition of the vectors **v1**, **v2** and **v3**, what is a correct way to perform a vector addition of **v1** and **v2**?

```
struct v {int x, y;} v1, v2, v3;
```

a) `v3 = v1 + v2;`

b) `v3 = v1[x]*v2[y] + v1[y]*v2[x]`

c) `v3[0] = v1[0] + v2[0];`
   `v3[1] = v1[1] + v2[1];`

d) `v3.x = v1.x + v2.x;`
   `v3.y = v1.y + v2.y;`

e) `v3->x = v1->x + v2->x;`
   `v3->y = v1->y + v2->y;`

EECS22: Advanced C Programming, Lecture 26   (c) 2016 R. Doemer   28

Review Quiz: Question 11

- Given the following definition of the vectors **v1**, **v2** and **v3**, what is a correct way to perform a vector addition of **v1** and **v2**?

```
struct v {int x, y;} v1, v2, v3;
```

a) `v3 = v1 + v2;`
b) `v3 = v1[x]*v2[y] + v1[y]*v2[x]`
c) `v3[0] = v1[0] + v2[0];`
    `v3[1] = v1[1] + v2[1];`
d) `v3.x = v1.x + v2.x;`
    `v3.y = v1.y + v2.y;`
e) `v3->x = v1->x + v2->x;`
    `v3->y = v1->y + v2->y;`

EECS22: Advanced C Programming, Lecture 26   (c) 2016 R. Doemer   29

Review Quiz: Question 12

- What could cause a **bus error**?
  (Check all that apply!)

  a) Waking up late and missing the bus.
  b) Calling a recursive function.
  c) Accessing an array with an index out of range.
  d) Referencing a pointer variable with invalid value.
  e) Accessing an integer variable with invalid value.

EECS22: Advanced C Programming, Lecture 26   (c) 2016 R. Doemer   30

Review Quiz: Question 12

- What could cause a **bus error**?
  (Check all that apply!)

  a) Waking up late and missing the bus.
  b) Calling a recursive function.
  c) Accessing an array with an index out of range.
  d) Referencing a pointer variable with invalid value.
  e) Accessing an integer variable with invalid value.

Review Quiz: Question 13

- Given the program segment below, what is
  the value of ***p** at the end?

  a) 1
  b) 2
  c) 3
  d) 4
  e) 5

```
1  int x[] = {1,2,3,4,5};
2  int *p = &x[2];
3
4  p++;
5  p -= 2;
```

Review Quiz: Question 13

- Given the program segment below, what is the value of ***p** at the end?

  a) 1

  b) **2**

  c) 3

  d) 4

  e) 5

```
1  int x[] = {1,2,3,4,5};
2  int *p = &x[2];
3
4  p++;
5  p -= 2;
```

Review Quiz: Question 14

- Given the function and variable definitions shown below, which function call is valid? (Check all that apply!)

  a) **StrLen(cp);**

  b) **StrLen(ca);**

  c) **StrLen(c);**

  d) **StrLen(i);**

  e) **StrLen("abc");**

```
1  int StrLen(
2          const char *s)
3  {   int l = 0;
4      while(*s)
5      { s++;
6        l++;
7      }
8      return l;
9  }
10 char *cp  = "hello";
11 char ca[] = "world";
12 char c = 'c';
13 int  i = 42;
```

Review Quiz: Question 14

- Given the function and variable definitions shown below, which function call is valid? (Check all that apply!)

a) **StrLen(cp);**

b) **StrLen(ca);**

c) StrLen(c);

d) StrLen(i);

e) **StrLen("abc");**

```
1  int StrLen(
2        const char *s)
3  {  int l = 0;
4     while(*s)
5     { s++;
6       l++;
7     }
8     return l;
9  }
10 char *cp  = "hello";
11 char ca[] = "world";
12 char c = 'c';
13 int  i = 42;
```

Review Quiz: Question 15

- What does the following code segment print?

```
1  char s[] = "Hppe!Mvdl!boe!Ibqqz!Ipmjebzt";
2  char *p;
3  p = &s[0];
4  while(*p)
5  { printf("%c", *p - 1);
6    p++;
7  }
```

a) **Hppe!Mvdl!boe!Ibqqz!Ipmjebzt**

b) **Happy Holidays and Good Luck**

c) **Happy Luck and Good Holidays**

d) **Good Holidays and Happy Luck**

e) **Good Luck and Happy Holidays**

Review Quiz: Question 15

- What does the following code segment print?

```
1  char s[] = "Hppe!Mvdl!boe!Ibqqz!Ipmjebzt";
2  char *p;
3  p = &s[0];
4  while(*p)
5  { printf("%c", *p - 1);
6    p++;
7  }
```

  a) Hppe!Mvdl!boe!Ibqqz!Ipmjebzt
  b) Happy Holidays and Good Luck
  c) Happy Luck and Good Holidays
  d) Good Holidays and Happy Luck
  e) Good Luck and Happy Holidays

EECS22: Advanced C Programming, Lecture 26   (c) 2016 R. Doemer   37