

EECS 22: Advanced C Programming

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

- Warm-up Quiz
- Assertions
 - Header file `assert.h`
 - Assertion `assert()`
 - Disabling assertions with `NDEBUG`

Quiz: Question 11


- Today's computers run at which clock speed?
 - a) 85 MPH
 - b) 1 kHz
 - c) 1 ms
 - d) 1 GHz
 - e) 1 MHz

EECS10: Computational Methods in ECE, Quiz 1-8

(c) 2010 R. Doemer

3

Quiz: Question 11

- Today's computers run at which clock speed?
 - a) 85 MPH
 - b) 1 kHz
 - c) 1 ms
 -  d) 1 GHz
 - e) 1 MHz

EECS10: Computational Methods in ECE, Quiz 1-8

(c) 2010 R. Doemer

4

Quiz: Question 12

- Which of the following names are valid keywords in C? (Check all that apply!)
- a) `if`
 - b) `when`
 - c) `void`
 - d) `main`
 - e) `Int`

EECS10: Computational Methods in ECE, Quiz 1-8

(c) 2010 R. Doemer

5

Quiz: Question 12

- Which of the following names are valid keywords in C? (Check all that apply!)
- a) `if`
 - b) `when`
 - c) `void`
 - d) `main`
 - e) `Int`

EECS10: Computational Methods in ECE, Quiz 1-8

(c) 2010 R. Doemer

6

Quiz: Question 13

- Which of the following names are valid identifiers in C? (Check all that apply!)
 - a) `xyz`
 - b) `IBM`
 - c) `dollar amount`
 - d) `My_Very_Long_Variable_Name`
 - e) `2fast4you`

EECS10: Computational Methods in ECE, Quiz 1-8

(c) 2010 R. Doemer

7

Quiz: Question 13

- Which of the following names are valid identifiers in C? (Check all that apply!)
 - a) `xyz`
 - b) `IBM`
 - c) `dollar amount`
 - d) `My_Very_Long_Variable_Name`
 - e) `2fast4you`

EECS10: Computational Methods in ECE, Quiz 1-8

(c) 2010 R. Doemer

8

Quiz: Question 14

- Which of the following constructs are valid type names in C? (Check all that apply!)
 - a) `long char`
 - b) `long double`
 - c) `signed long long`
 - d) `unsigned float`
 - e) `signed integer`

EECS10: Computational Methods in ECE, Quiz 1-8

(c) 2010 R. Doemer

9

Quiz: Question 14

- Which of the following constructs are valid type names in C? (Check all that apply!)
 - a) `long char`
 - b) `long double`
 - c) `signed long long`
 - d) `unsigned float`
 - e) `signed integer`

EECS10: Computational Methods in ECE, Quiz 1-8

(c) 2010 R. Doemer

10

Quiz: Question 15




- Which of the following constants is of type **double**?
(Check all that apply!)
 - a) **42**
 - b) **.42**
 - c) **4e2**
 - d) **4E2**
 - e) **42f**

EECS10: Computational Methods in ECE, Quiz 1-8

(c) 2010 R. Doemer

11

Quiz: Question 15

- Which of the following constants is of type **double**?
(Check all that apply!)
 - a) 42
 -  b) **.42**
 -  c) **4e2**
 -  d) **4E2**
 - e) 42f

EECS10: Computational Methods in ECE, Quiz 1-8

(c) 2010 R. Doemer

12

Quiz: Question 16

- What is the value of the integer x after the following statement?

```
x = 3 << 2 >> 1;
```

- a) **Syntax Error!**
- b) 3
- c) 6
- d) 12
- e) 321

EECS10: Computational Methods in ECE, Quiz 1-8


(c) 2010 R. Doemer

13

Quiz: Question 16

- What is the value of the integer x after the following statement?

```
x = 3 << 2 >> 1;
```

- a) Syntax Error!
- b) 3
-  c) 6
- d) 12
- e) 321

EECS10: Computational Methods in ECE, Quiz 1-8

(c) 2010 R. Doemer

14

Quiz: Question 17



- Which of the following expressions correctly computes the polynomial $p = 2x^2 - 3x + 4$?
(Check all that apply!)
 - `p = 2x^2 - 3x + 4;`
 - `p = 2xx - 3x + 4;`
 - `p = x*x*2 - 3*x + 4.0;`
 - `p = 2*(x*x + 3)*x + 4;`
 - `p = (2*x - 3)*x + 4;`

EECS10: Computational Methods in ECE, Quiz 1-8

(c) 2010 R. Doemer

15

Quiz: Question 17

- Which of the following expressions correctly computes the polynomial $p = 2x^2 - 3x + 4$?
(Check all that apply!)
 - `p = 2x^2 - 3x + 4;`
 - `p = 2xx - 3x + 4;`
 -  `p = x*x*2 - 3*x + 4.0;`
 - `p = 2*(x*x + 3)*x + 4;`
 -  `p = (2*x - 3)*x + 4;`

EECS10: Computational Methods in ECE, Quiz 1-8

(c) 2010 R. Doemer

16

Quiz: Question 18

- What is the result of the evaluation of the following expression?

```
1 == 2 || 3 < 4 && 5 > 6
```

- a) 123456
- b) true
- c) false
- d) 1
- e) 0

EECS10: Computational Methods in ECE, Quiz 1-8


(c) 2010 R. Doemer

17

Quiz: Question 18

- What is the result of the evaluation of the following expression?

```
1 == 2 || 3 < 4 && 5 > 6
```

- a) 123456
- b) true
- c) false
- d) 1
-  e) 0

EECS10: Computational Methods in ECE, Quiz 1-8

(c) 2010 R. Doemer

18

Quiz: Question 19

- What is the result of the evaluation of the following expression?


```
17 < 42 ? 17 : 42
```

- a) 1742
- b) 17
- c) 42
- d) true
- e) false

Quiz: Question 19

- What is the result of the evaluation of the following expression?

```
17 < 42 ? 17 : 42
```

- a) 1742
-  b) 17
- c) 42
- d) true
- e) false

Quiz: Question 20

- For integer $x = 1$ at the beginning, what is the value of x after the following statement?

```
x += x + 1;
```

- a) 0
- b) 1
- c) 2
- d) 3
- e) 4

EECS10: Computational Methods in ECE, Quiz 1-8


(c) 2010 R. Doemer

21

Quiz: Question 20

- For integer $x = 1$ at the beginning, what is the value of x after the following statement?

```
x += x + 1;
```

- a) 0
- b) 1
- c) 2
-  d) 3
- e) 4

EECS10: Computational Methods in ECE, Quiz 1-8

(c) 2010 R. Doemer

22

Assertions

- Run-time Checks for Diagnostics and Debugging
 - Can be manually implemented

```
...
#ifdef DEBUG
if (value > 100)
{ printf("Value is over 100!");
  abort();
} /* fi */
#endif /* DEBUG */
...
```

- Assertions
 - Diagnostics provided by standard C library

```
...
#include <assert.h>
...
assert(value <= 100);
...
```

EECS22: Advanced C Programming, Lecture 9

(c) 2011 R. Doemer

23

Assertions

- Diagnostics provided by standard C library

```
#include <assert.h>
...
assert(value <= 100);
```

- Header file **assert.h**
 - Defines `assert(condition)` as a preprocessor macro
- Assertion failure
 - At run-time, if `condition` evaluates to `false`, the program is aborted with a diagnostic message

```
assertion: program.c:12: main: Assertion `value <= 100' failed.
Abort
```

- Disabling assertions
 - If `NDEBUG` is defined when `assert.h` is included, the `assert()` macro is ignored (empty)
 - Use `NDEBUG` for final programs delivered to the end user!

EECS22: Advanced C Programming, Lecture 9

(c) 2011 R. Doemer

24

Assertions

- Recommendations on Using Assertions
 - Use assertions often
 - Confirm assumptions about parameters, calculated values
 - Assertions are cheap (low run-time overhead)
 - Use assertions from beginning during software development
 - Diagnostic messages are very helpful in development
 - Program aborts as soon as a value is out of expected range
 - Location and reason of problem are shown
 - Often avoids larger problems later
 - Disable assertions for final program
 - Diagnostic messages are of no use to the end user!
 - User has no idea about condition and source location
 - Beware of side-effects in assertions
 - Implemented as a macro!
 - Can lead to *Heisenbugs* which disappear when debugging is on!