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

- Review
  - Lecture 7: Formatted I/O, structured programming
  - Lecture 8: Structured programming, loops
  - Lecture 9: Functions, concepts, hierarchy, stack
  - Lecture 10: Functions, scoping, library functions
  - Lecture 11: Data structures, arrays
  - Lecture 12: Character arrays, strings
- Midterm Review Quiz
  - Top 5 “most difficult” questions
- Review Quiz
  - 25 new questions

Midterm 1 Review Quiz

- Top 5 “most difficult” questions:
- Which of the following C expressions yield the same result?
(Check all that apply!)

a) `4 << 8 % 5 / 2`
b) `(4 << 8) % 5 / 2`
c) `4 << 8 % (5 / 2)`
d) `(4 << 8 % 5) / 2`
e) `4 << (8 % 5) / 2`

Midterm 1 Review Quiz

- Top 5 “most difficult” questions:
- Which of the following C expressions yield the same result?
(Check all that apply!)

a) `4 << 8 % 5 / 2` (8)
b) `(4 << 8) % 5 / 2` (2)
c) `4 << 8 % (5 / 2)` (4)
d) `(4 << 8 % 5) / 2` (16)
e) `4 << (8 % 5) / 2` (8)

Midterm 1 Review Quiz

- Top 5 “most difficult” questions:
- Which of the following program fragments will *not* terminate? (Check all that apply!)

a)
```
int a = 1;
while(a < 1000000)
   { a++; }
```

b)
```
int a = 0;
while(a < 1000)
   { a = a * 3; }
```

c)
```
int a = 1;
while(a == 1)
   { a = a % 10; }
```

d)
```
int a = 10;
while(a > 0)
   { a = a / 3; }
```

e)
```
int a = 1;
while(a < 1000)
   { a = a << 1; }
```


Midterm 1 Review Quiz

- Top 5 “most difficult” questions:
- Which of the following program fragments will *not* terminate? (Check all that apply!)

a)
```
int a = 1;
while(a < 1000000)
   { a++; }
```

b)
```
int a = 0;
while(a < 1000)
   { a = a * 3; }
```

c)
```
int a = 1;
while(a == 1)
   { a = a % 10; }
```

d)
```
int a = 10;
while(a > 0)
   { a = a / 3; }
```

e)
```
int a = 1;
while(a < 1000)
   { a = a << 1; }
```

Midterm 1 Review Quiz

- Top 5 “most difficult” questions:
- What is the output of the following C program fragment?

```
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:
- What is the output of the following C program fragment?

```
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:

- Prime number test: Iterate over $2 \le i < x$ to find a divisor of x. What should go into the box in line 4?
  a) `i = 0;`
  b) `i = 1;`
  c) `i = 2;`
  d) `i = x;`
  e) `x = 0;`

```
int x, i;
printf("Please input a number: ");
scanf("%d", &x);
initialize variable i
while(i < x)
{ if(x % i == 0)
  { printf("%d is not prime\n", x);
    break;
  }
  i++;
}
if( none of the i is a divisor of x )
  { printf("%d is prime\n", x);
  }
```


Midterm 1 Review Quiz

- Top 5 “most difficult” questions:

- Prime number test: Iterate over $2 \le i < x$ to find a divisor of x. What should go into the box in line 4?
  a) `i = 0;`
  b) `i = 1;`
  c) `i = 2;`
  d) `i = x;`
  e) `x = 0;`

```
int x, i;
printf("Please input a number: ");
scanf("%d", &x);
initialize variable i
while(i < x)
{ if(x % i == 0)
  { printf("%d is not prime\n", x);
    break;
  }
  i++;
}
if( none of the i is a divisor of x )
  { printf("%d is prime\n", x);
  }
```

Midterm 1 Review Quiz

- Top 5 “most difficult” questions:

- Prime number test: Iterate over $2 \le i < x$ to find a divisor of x. What should go into the box in line 12?
  a) `x / i == 0`
  b) `x < i`
  c) `i / x == 0`
  d) `i + 1 == x`
  e) `i == x`

```
int x, i;
printf("Please input a number: ");
scanf("%d", &x);
initialize variable i
while(i < x)
{ if(x % i == 0)
  { printf("%d is not prime\n", x);
    break;
  }
  i++;
}
if( none of the i is a divisor of x )
  { printf("%d is prime\n", x);
  }
```


Midterm 1 Review Quiz

- Top 5 “most difficult” questions:

- Prime number test: Iterate over $2 \le i < x$ to find a divisor of x. What should go into the box in line 12?
  a) `x / i == 0`
  b) `x < i`
  c) `i / x == 0`
  d) `i + 1 == x`
  e) `i == x`

```
int x, i;
printf("Please input a number: ");
scanf("%d", &x);
initialize variable i
while(i < x)
{ if(x % i == 0)
  { printf("%d is not prime\n", x);
    break;
  }
  i++;
}
if( none of the i is a divisor of x )
  { printf("%d is prime\n", x);
  }
```

EECS10: Computational Methods in ECE, Lecture 13 (c) 2018 R. Doemer 12

Quiz: Question 1

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

a) `(int)5.99 > 5`
b) `1 || 0 && 1`
c) `5 >= 5`
d) `(1 + 2 + 3) == (3 << 2 >> 1)`
e) `5 - 5`

Quiz: Question 1

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

a) `(int)5.99 > 5`
b) `1 || 0 && 1`
c) `5 >= 5`
d) `(1 + 2 + 3) == (3 << 2 >> 1)`
e) `5 - 5`

Quiz: Question 2

- If **`count`** is an integer counter that counts upwards in steps of 1, how could one update the value of **`count`**?
  (Check all that apply!)

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

Quiz: Question 2

- If **`count`** is an integer counter that counts upwards in steps of 1, how could one update the value of **`count`**?
  (Check all that apply!)

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

Quiz: Question 3

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

```
int x = 0;
for(x = 1; x <= 10; x++)
   {  }
```

a) **0**
b) **1**
c) **9**
d) **10**
e) **11**

Quiz: Question 3

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

```
int x = 0;
for(x = 1; x <= 10; x++)
   {  }
```

a) 0
b) 1
c) 9
d) 10
e) **11**

Quiz: Question 4

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

```
int x = 0;
do { x++;
    } while(x < 9);
```

a) 0
b) 1
c) 9
d) 10
e) 11

Quiz: Question 4

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

```
int x = 0;
do { x++;
    } while(x < 9);
```

a) 0
b) 1
c) 9
d) 10
e) 11

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

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

Quiz: Question 6

- Given the following function `g`, what is the result of `g(85)`?

a) `'A'`
b) `'B'`
c) `'C'`
d) `'D'`
e) `'F'`

```
char g(int n)
{
  switch(n/10)
  { case 10:
    case  9: return('A');
    case  8: return('B');
    case  7: return('C');
    case  6: return('D');
    default: return('F');
   }
}
```


Quiz: Question 6

- Given the following function `g`, what is the result of `g(85)`?

a) `'A'`
b) `'B'` ⇐
c) `'C'`
d) `'D'`
e) `'F'`

```
char g(int n)
{
  switch(n/10)
  { case 10:
    case  9: return('A');
    case  8: return('B');
    case  7: return('C');
    case  6: return('D');
    default: return('F');
   }
}
```

Quiz: Question 7

- What is output by the following C statement?

```
printf("x = %03d", 3 + 4);
```

a) `x = 034`
b) `x = 037`
c) `x = 007`
d) `x = 7`
e) `x = 347`

Quiz: Question 7

- What is output by the following C statement?

```
printf("x = %03d", 3 + 4);
```

a) `x = 034`
b) `x = 037`
c) `x = 007`
d) `x = 7`
e) `x = 347`

Quiz: Question 8

- In the **`gdb`** debugger, what does **`next`** do?

a) It moves to the next argument of the function.
b) It calls the next function in the program.
c) It executes the next statement in the program.
d) It prints the value of the next variable.
e) It loads the next program into the debugger.

Quiz: Question 8

- In the **`gdb`** debugger, what does **`next`** do?

a) It moves to the next argument of the function.
b) It calls the next function in the program.
c) It executes the next statement in the program.
d) It prints the value of the next variable.
e) It loads the next program into the debugger.

Quiz: Question 9

- Given the following code fragment, which of the following statements are true? (Check all that apply!)

```
double f(int x);
int g(int x, int y)
{
  int z;

  z = f(x) + 2*y;
  return z;
}
```

a) Function **f** is declared.
b) Function **g** calls function **f**
c) Variable **z** is a local variable of function **g**
d) Function **g** is declared and defined.
e) **y** is a parameter of function **g**.

Quiz: Question 9

- Given the following code fragment, which of the following statements are true? (Check all that apply!)

```
double f(int x);
int g(int x, int y)
{
  int z;

  z = f(x) + 2*y;
  return z;
}
```

a) Function **f** is declared.
b) Function **g** calls function **f**
c) Variable **z** is a local variable of function **g**
d) Function **g** is declared and defined.
e) **y** is a parameter of function **g**.

Quiz: Question 10

- 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`

Quiz: Question 10

- 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`

Quiz: Question 11

- Given the following program fragment, what is the value of `g(2,f(3,4))`?

a) `8`
b) `9`
c) `10`
d) `11`
e) `12`

```
int x = 7;

int f(int x, int y)
{
  return x + y;
}

int g(int x, int y)
{
  return f(y, x);
}
```


Quiz: Question 11

- Given the following program fragment, what is the value of `g(2,f(3,4))`?

a) `8`
b) `9`
c) `10`
d) `11`
e) `12`

```
int x = 7;

int f(int x, int y)
{
  return x + y;
}

int g(int x, int y)
{
  return f(y, x);
}
```

Quiz: Question 12

- What is output by the following program fragment?

```
char s[] = "EECS10";

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

a) `EECS00 1`
b) `EEC 10 0`
c) `E E`
d) `EECS C`
e) `EEC C`

Quiz: Question 12

- What is output by the following program fragment?

```
char s[] = "EECS10";

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

a) `EECS00 1`
b) `EEC 10 0`
c) `E E`
d) `EECS C`
e) `EEC C`

Quiz: Question 13

- Given the definition `double p=0.0125;` which of the following C statements will print out `p = 1.25%` ?
  (Check all that apply!)

a) `printf("p = %d.25%%", (int)(p*100.0));`
b) `printf("p = %p", 100.0*p);`
c) `printf("p = %.2f%%", p*100.0);`
d) `printf("p = %.2f%c", p*100.0, '%');`
e) `printf("p = ", 100.0 * p, "%%");`

Quiz: Question 13

- Given the definition `double p=0.0125;` which of the following C statements will print out `p = 1.25%` ?
  (Check all that apply!)

a) `printf("p = %d.25%%", (int)(p*100.0));`
b) `printf("p = %p", 100.0*p);`
c) `printf("p = %.2f%%", p*100.0);`
d) `printf("p = %.2f%c", p*100.0, '%');`
e) `printf("p = ", 100.0 * p, "%%");`

Quiz: Question 14

- 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.

Quiz: Question 14

- 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.

Quiz: Question 15

- Which of the following declarations can be added to the program in line 8 without creating a compilation error?
  (Check all that apply!)

a) `int f(int v, double w);`
b) `int g = 0;`
c) `int g(int x, int y);`
d) `int x = 2;`
e) `int f(double v, double w);`

```
1  int x = 2;
2  int f(int v, double w);
3  int g(int x, int y)
4  { int z;
5    z = 2*x + 5*y - 42;
6    return z;
7  }
8
```


Quiz: Question 15

- Which of the following declarations can be added to the program in line 8 without creating a compilation error?
  (Check all that apply!)

a) `int f(int v, double w);`
b) `int g = 0;`
c) `int g(int x, int y);`
d) `int x = 2;`
e) `int f(double v, double w);`

```
1  int x = 2;
2  int f(int v, double w);
3  int g(int x, int y)
4  { int z;
5    z = 2*x + 5*y - 42;
6    return z;
7  }
8
```

Quiz: Question 16

- The following function **`issorted`** is supposed to return true if and only if the given array **`L`** is sorted in increasing order.
- What should go into *Box1* in line 3?

a) `i=1; i<10; i++`
b) `i=0; i<10; i++`
c) `i=0; i<9; i++`
d) `i=10; i>0; i--`
e) `i=9; i>=0; i--`

```
1  int issorted(int L[10])
2  { int i;
3    for( Box1 )
4    { if(L[i] >= L[i+1])
5      { Box2 ; }
6    }
7    Box3 ;
8  }
```


Quiz: Question 16

- The following function **`issorted`** is supposed to return true if and only if the given array **`L`** is sorted in increasing order.
- What should go into *Box1* in line 3?

a) `i=1; i<10; i++`
b) `i=0; i<10; i++`
c) **`i=0; i<9; i++`**
d) `i=10; i>0; i--`
e) `i=9; i>=0; i--`

```
1  int issorted(int L[10])
2  { int i;
3    for( Box1 )
4    { if(L[i] >= L[i+1])
5      { Box2 ; }
6    }
7    Box3 ;
8  }
```

Quiz: Question 17

- The following function **`issorted`** is supposed to return true if and only if the given array **`L`** is sorted in increasing order.
- What should go into *Box2* in line 5?

a) **`return 0`**
b) **`return 1`**
c) **`continue`**
d) **`break`**
e) **`return`**

```
1  int issorted(int L[10])
2  { int i;
3    for( Box1 )
4    { if(L[i] >= L[i+1])
5      { Box2 ; }
6    }
7    Box3 ;
8  }
```


Quiz: Question 17

- The following function **`issorted`** is supposed to return true if and only if the given array **`L`** is sorted in increasing order.
- What should go into *Box2* in line 5?

a) **`return 0`**
b) **`return 1`**
c) **`continue`**
d) **`break`**
e) **`return`**

```
1  int issorted(int L[10])
2  { int i;
3    for( Box1 )
4    { if(L[i] >= L[i+1])
5      { Box2 ; }
6    }
7    Box3 ;
8  }
```

Quiz: Question 18

- The following function **`issorted`** is supposed to return true if and only if the given array **`L`** is sorted in increasing order.
- What should go into *Box3* in line 7?

a) **`return 0`**
b) **`return 1`**
c) **`continue`**
d) **`break`**
e) **`return`**

```
1  int issorted(int L[10])
2  { int i;
3    for( Box1 )
4    { if(L[i] >= L[i+1])
5      { Box2 ; }
6    }
7    Box3 ;
8  }
```


Quiz: Question 18

- The following function **`issorted`** is supposed to return true if and only if the given array **`L`** is sorted in increasing order.
- What should go into *Box3* in line 7?

a) **`return 0`**
b) **`return 1`**
c) **`continue`**
d) **`break`**
e) **`return`**

```
1  int issorted(int L[10])
2  { int i;
3    for( Box1 )
4    { if(L[i] >= L[i+1])
5      { Box2 ; }
6    }
7    Box3 ;
8  }
```

Quiz: Question 19

- What is output by the following C statement?

```
int x = 0, y = 5;
x = y++;
printf("x = %d, y = %d", x, y);
```

a) `x = 0, y = 5`
b) `x = 5, y = 5`
c) `x = 5, y = 6`
d) `x = 6, y = 5`
e) `x = 6, y = 6`

Quiz: Question 19

- What is output by the following C statement?

```
int x = 0, y = 5;
x = y++;
printf("x = %d, y = %d", x, y);
```

a) `x = 0, y = 5`
b) `x = 5, y = 5`

c) `x = 5, y = 6`
d) `x = 6, y = 5`
e) `x = 6, y = 6`

Quiz: Question 20

- What is output by the following C statement?

```
int x = 0, y = 5;
x = ++y;
printf("x = %d, y = %d", x, y);
```

a) `x = 0, y = 5`
b) `x = 5, y = 5`
c) `x = 5, y = 6`
d) `x = 6, y = 5`
e) `x = 6, y = 6`

Quiz: Question 20

- What is output by the following C statement?

```
int x = 0, y = 5;
x = ++y;
printf("x = %d, y = %d", x, y);
```

a) `x = 0, y = 5`
b) `x = 5, y = 5`
c) `x = 5, y = 6`
d) `x = 6, y = 5`
e) `x = 6, y = 6`

Quiz: Question 21

- Consider the following program fragment:

```
unsigned int x=0, y=0;
scanf("%d", &x);
while((x>>=1) != 0)
   {y += 1;}
printf("%d", y);
```

- When running the program, which of the following is correct? (Check all that apply!)
  a) If the user enters **6**, it will print **2**.
  b) If the user enters **6**, it will print **3**.
  c) If the user enters **4**, it will print **2**.
  d) If the user enters **4**, it will print **1**.
  e) If the user enters **4**, it will print **4**.

EECS10: Computational Methods in ECE, Lecture 7 (c) 2013 R. Doemer 53

Quiz: Question 21

- Consider the following program fragment:

```
unsigned int x=0, y=0;
scanf("%d", &x);
while((x>>=1) != 0)
   {y += 1;}
printf("%d", y);
```

- When running the program, which of the following is correct? (Check all that apply!)
  a) If the user enters **6**, it will print **2**.
  b) If the user enters **6**, it will print **3**.
  c) If the user enters **4**, it will print **2**.
  d) If the user enters **4**, it will print **1**.
  e) If the user enters **4**, it will print **4**.

EECS10: Computational Methods in ECE, Lecture 7 (c) 2013 R. Doemer 54

Quiz: Question 22

- Consider the following program fragment:

```
unsigned int x=0, y=0;
scanf("%d", &x);
while((x>>=1) != 0)
   {y += 1;}
printf("%d", y);
```

- Which of the following statements are true about the program? (Check all that apply!)
  a) **y** will be the integer part of $\log_2(x)$
  b) **y** will be equal to **x**
  c) It computes the product of **x** and **y**
  d) It sets **y** to the sum of **x** and **y**
  e) The condition in line 3 is equivalent to **(x/=2)!=0**

Quiz: Question 22

- Consider the following program fragment:

```
unsigned int x=0, y=0;
scanf("%d", &x);
while((x>>=1) != 0)
   {y += 1;}
printf("%d", y);
```

- Which of the following statements are true about the program? (Check all that apply!)
  a) **y** will be the integer part of $\log_2(x)$
  b) **y** will be equal to **x**
  c) It computes the product of **x** and **y**
  d) It sets **y** to the sum of **x** and **y**
  e) The condition in line 3 is equivalent to **(x/=2)!=0**

Quiz: Question 23

- Which of the following expressions yield a result type of **`double`**? (Check all that apply!)

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

Quiz: Question 23

- Which of the following expressions yield a result type of **`double`**? (Check all that apply!)

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

Quiz: Question 24

- What is output by the following C statement?

```
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`

Quiz: Question 24

- What is output by the following C statement?

```
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`

Quiz: Question 25

- Consider the following C program fragment regarding systolic blood pressure (line numbers are not part of the code):
- 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!)

```
 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"); }
```

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");`**

Quiz: Question 25

- Consider the following C program fragment regarding systolic blood pressure (line numbers are not part of the code):
- 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!)

```
 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"); }
```

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");`**