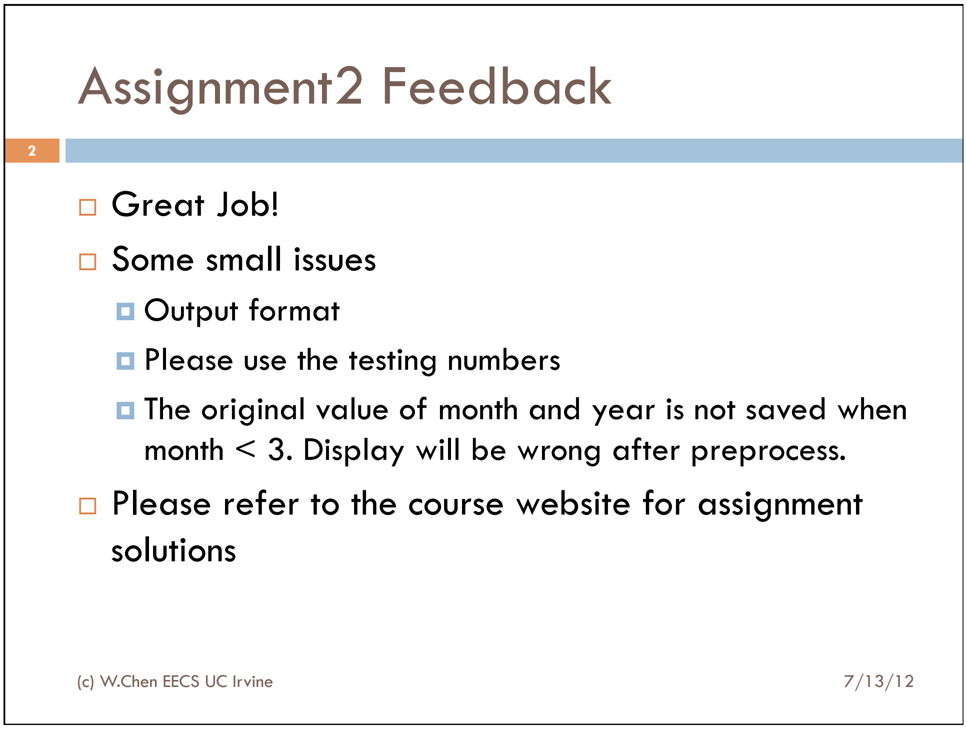
(c) W.Chen EECS UC Irvine 1

EECS10 DISCUSSION

7/13/12

Week3 Session2

Weiwei Chen

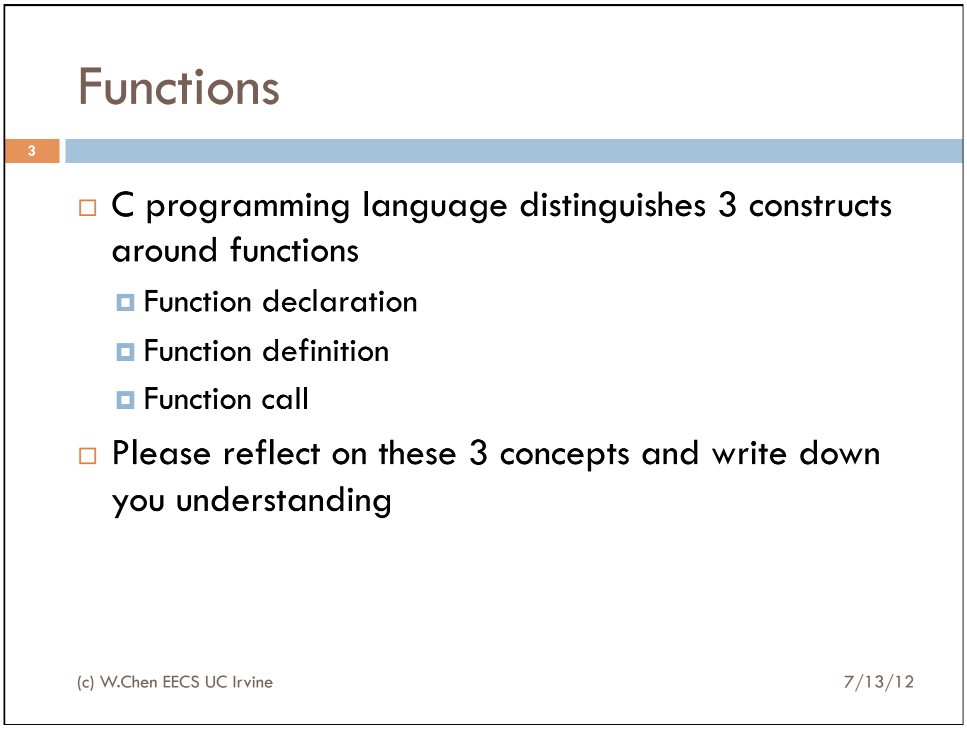


Assignment2 Feedback

2

- Great Job!
- Some small issues
  - Output format
  - Please use the testing numbers
  - The original value of month and year is not saved when month < 3. Display will be wrong after preprocess.
- Please refer to the course website for assignment solutions

(c) W.Chen EECS UC Irvine 7/13/12

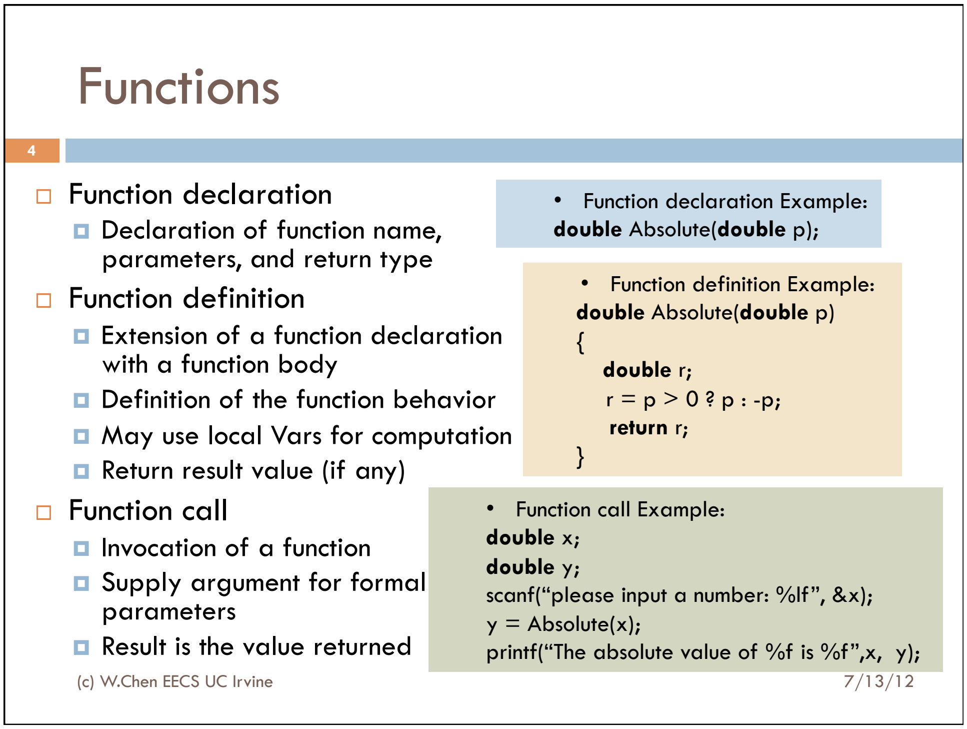


Functions

3

- C programming language distinguishes 3 constructs around functions
  - Function declaration
  - Function definition
  - Function call
- Please reflect on these 3 concepts and write down you understanding

(c) W.Chen EECS UC Irvine 7/13/12



Functions

4

- Function declaration
  - Declaration of function name, parameters, and return type
- Function definition
  - Extension of a function declaration with a function body
  - Definition of the function behavior
  - May use local Vars for computation
  - Return result value (if any)
- Function call
  - Invocation of a function
  - Supply argument for formal parameters
  - Result is the value returned

- Function declaration Example:

```
double Absolute(double p);
```

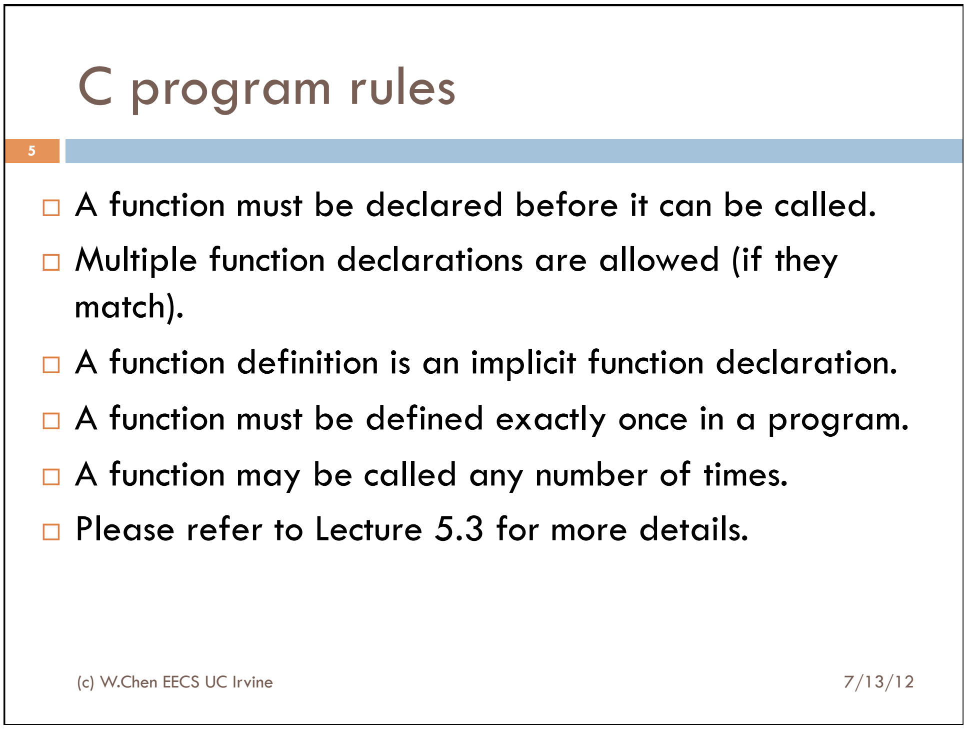
- Function definition Example:

```
double Absolute(double p)
{
   double r;
   r = p > 0 ? p : -p;
   return r;
}
```

- Function call Example:

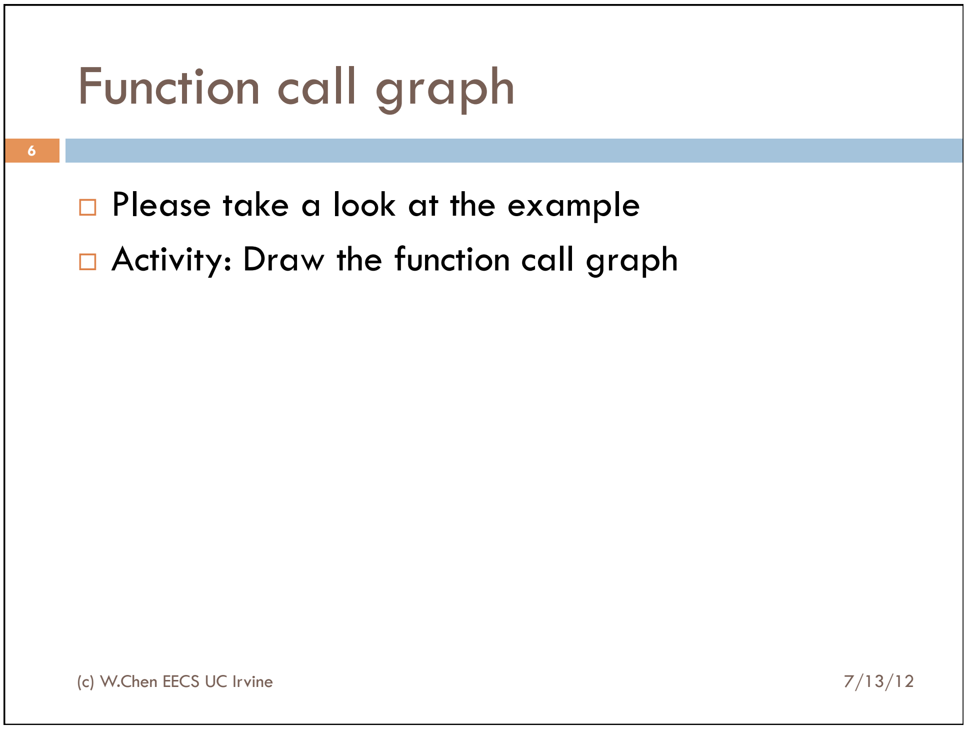
```
double x;
double y;
scanf("please input a number: %lf", &x);
y = Absolute(x);
printf("The absolute value of %f is %f",x,  y);
```

(c) W.Chen EECS UC Irvine 7/13/12



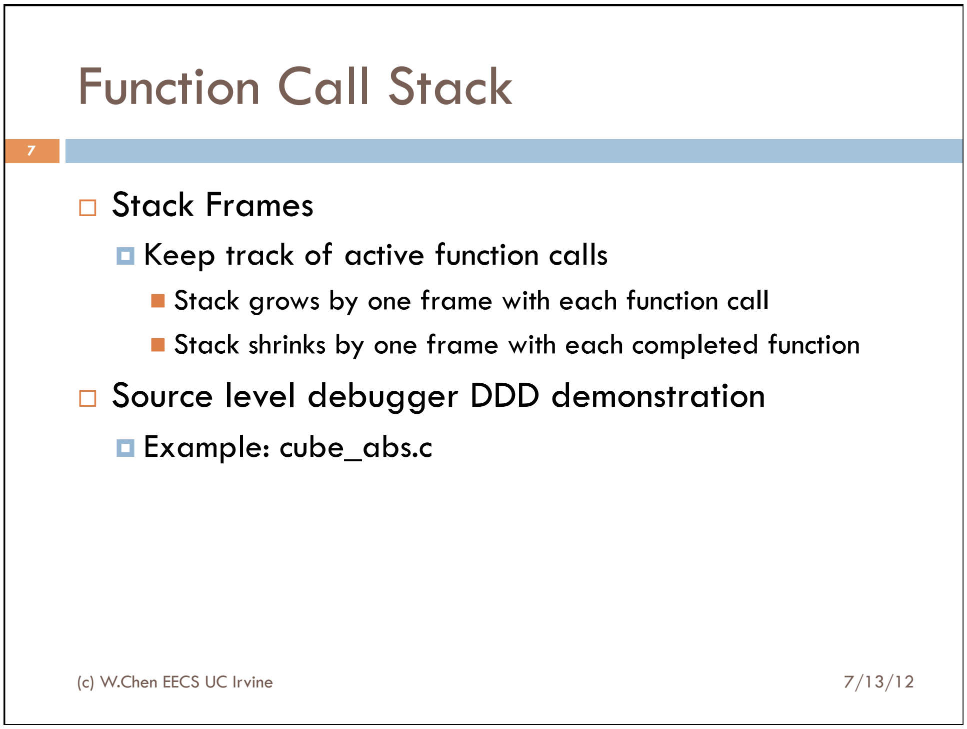
C program rules

- □ A function must be declared before it can be called.
- □ Multiple function declarations are allowed (if they match).
- □ A function definition is an implicit function declaration.
- □ A function must be defined exactly once in a program.
- □ A function may be called any number of times.
- □ Please refer to Lecture 5.3 for more details.



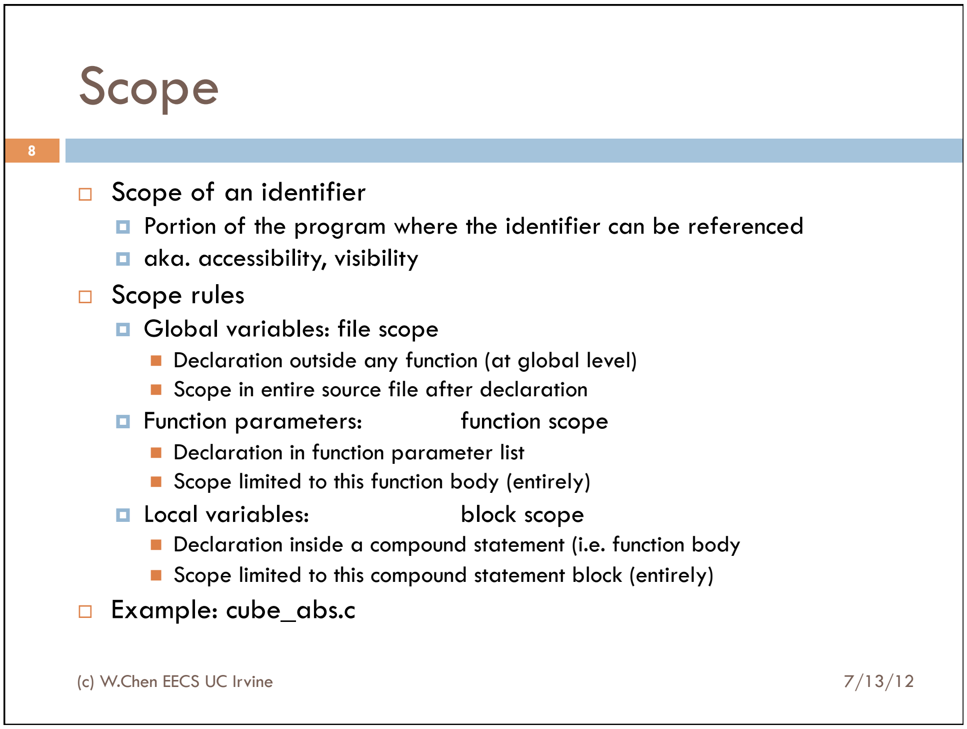
Function call graph

- □ Please take a look at the example
- □ Activity: Draw the function call graph



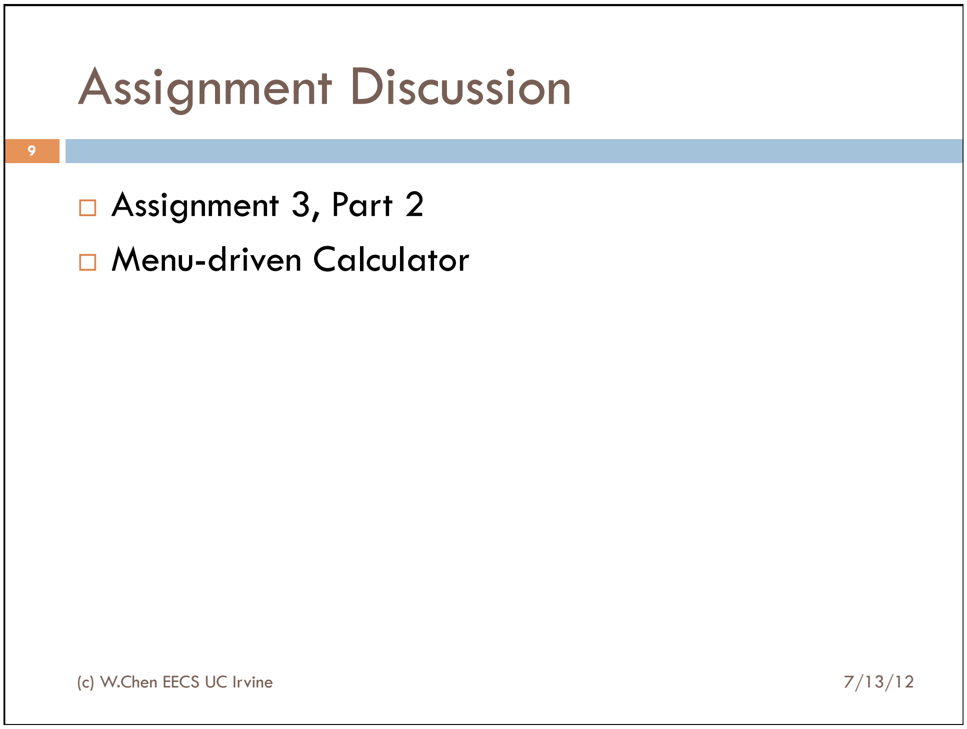
Function Call Stack

- Stack Frames
  - Keep track of active function calls
    - Stack grows by one frame with each function call
    - Stack shrinks by one frame with each completed function
- Source level debugger DDD demonstration
  - Example: cube_abs.c



Scope

- Scope of an identifier
  - Portion of the program where the identifier can be referenced
  - aka. accessibility, visibility
- Scope rules
  - Global variables: file scope
    - Declaration outside any function (at global level)
    - Scope in entire source file after declaration
  - Function parameters: function scope
    - Declaration in function parameter list
    - Scope limited to this function body (entirely)
  - Local variables: block scope
    - Declaration inside a compound statement (i.e. function body
    - Scope limited to this compound statement block (entirely)
- Example: cube_abs.c

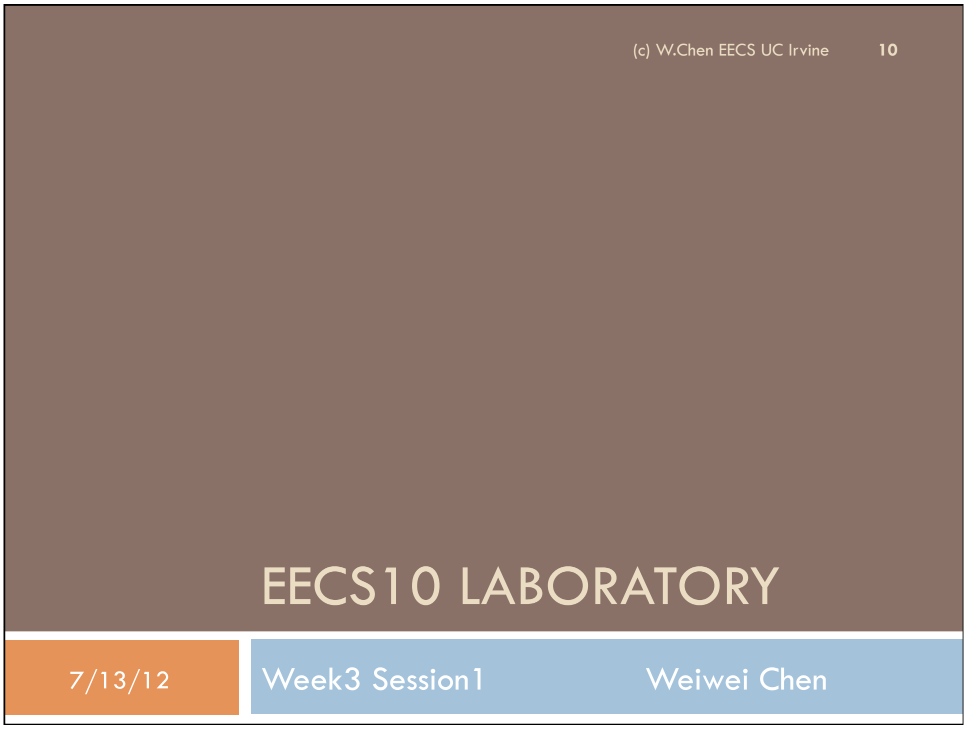


Assignment Discussion

9

- □ Assignment 3, Part 2
- □ Menu-driven Calculator

(c) W.Chen EECS UC Irvine 7/13/12

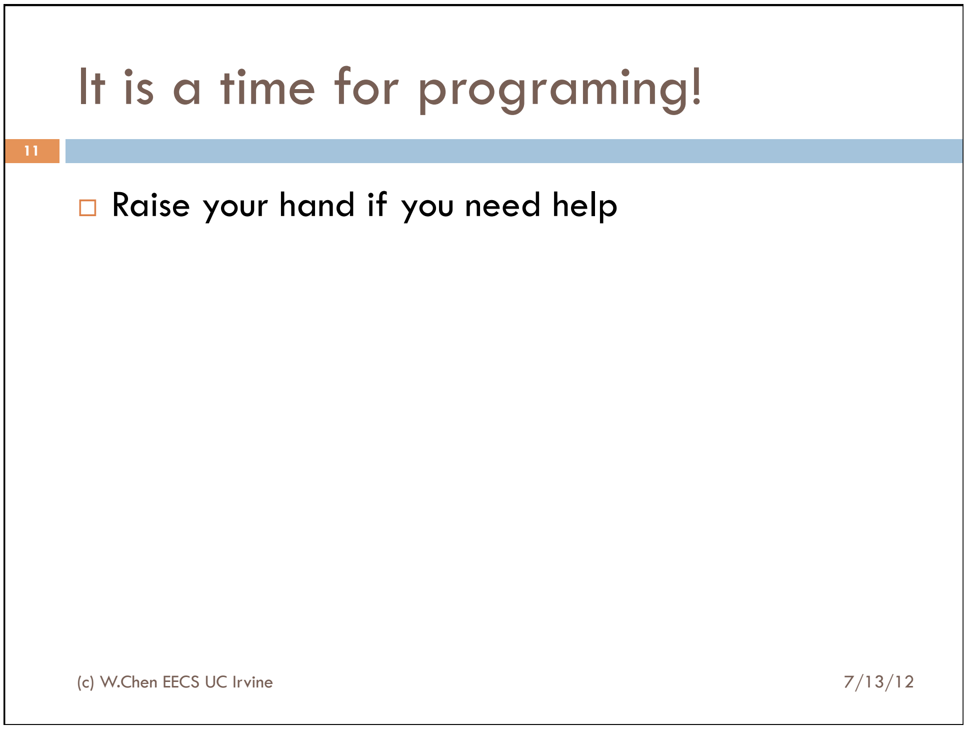


(c) W.Chen EECS UC Irvine 10

EECS10 LABORATORY

7/13/12

Week3 Session1 Weiwei Chen



It is a time for programing!

11

- □ Raise your hand if you need help

(c) W.Chen EECS UC Irvine

7/13/12