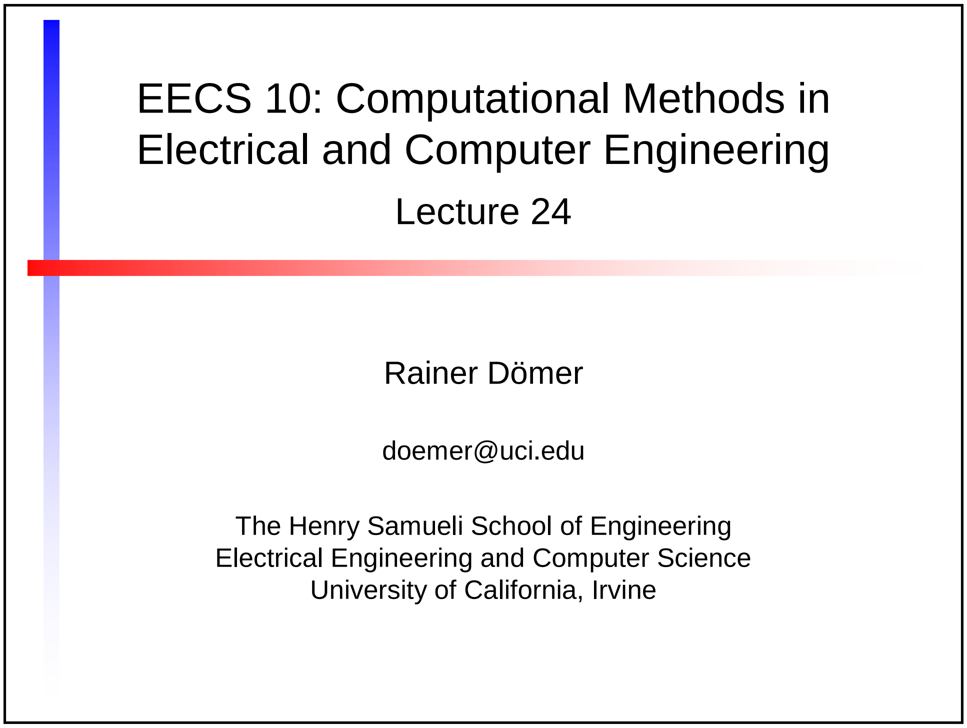


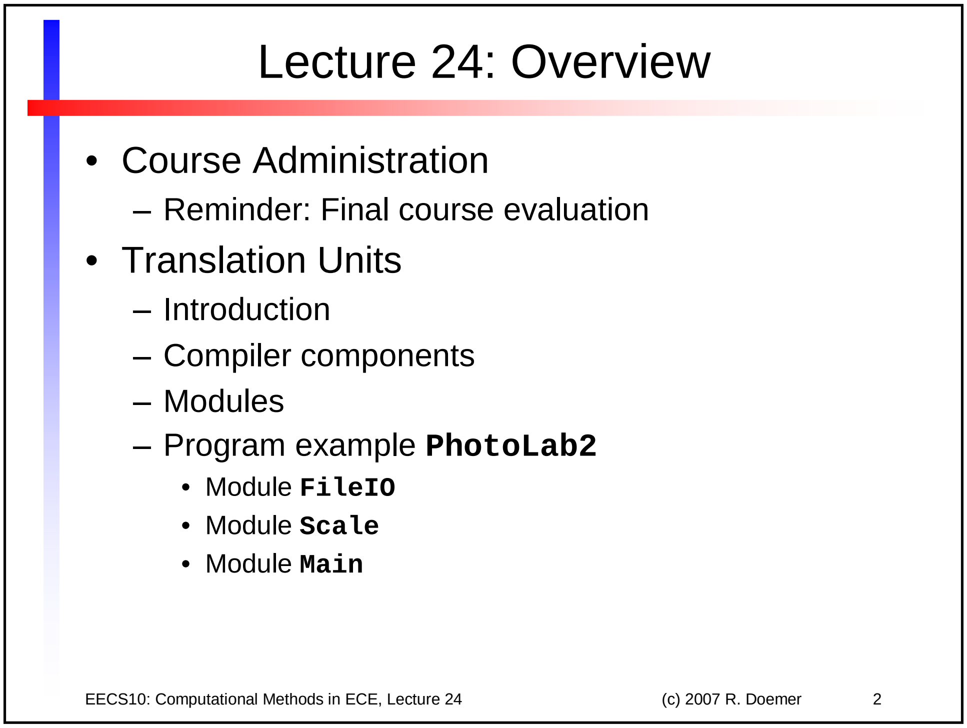
EECS 10: Computational Methods in Electrical and Computer Engineering

Lecture 24

Rainer Dömer

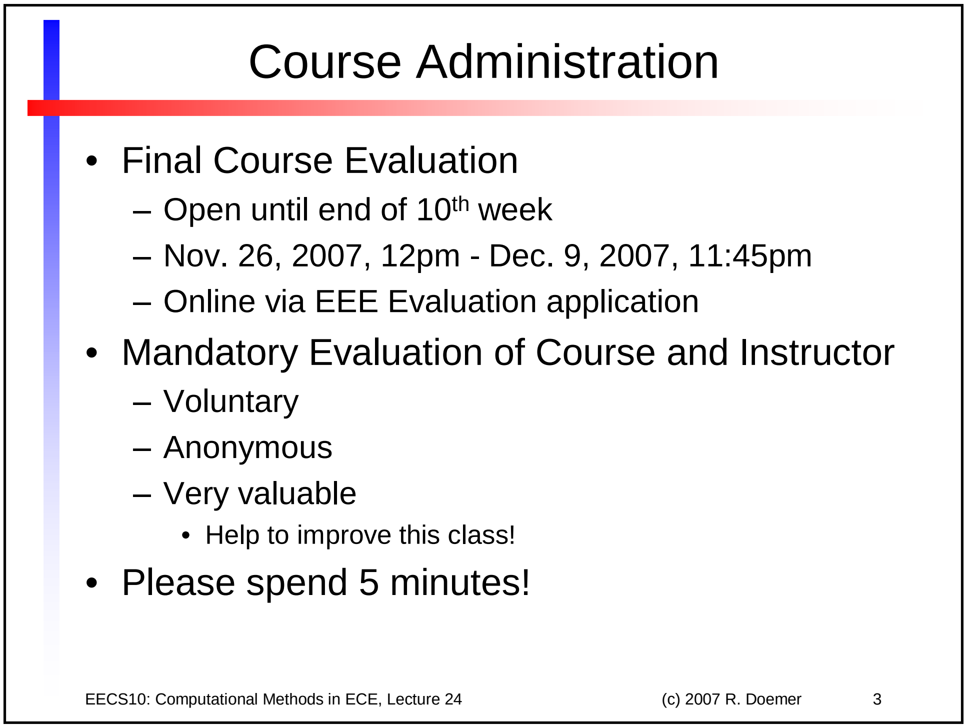
doemer@uci.edu

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



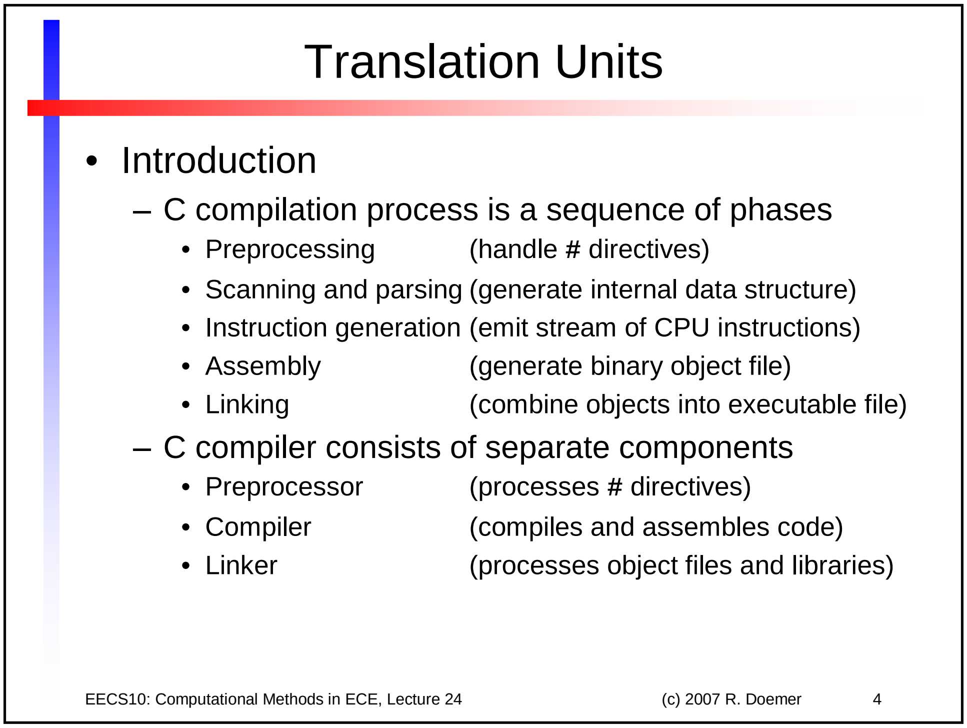
Lecture 24: Overview

- Course Administration
  - Reminder: Final course evaluation
- Translation Units
  - Introduction
  - Compiler components
  - Modules
  - Program example **`PhotoLab2`**
    - Module **`FileIO`**
    - Module **`Scale`**
    - Module **`Main`**



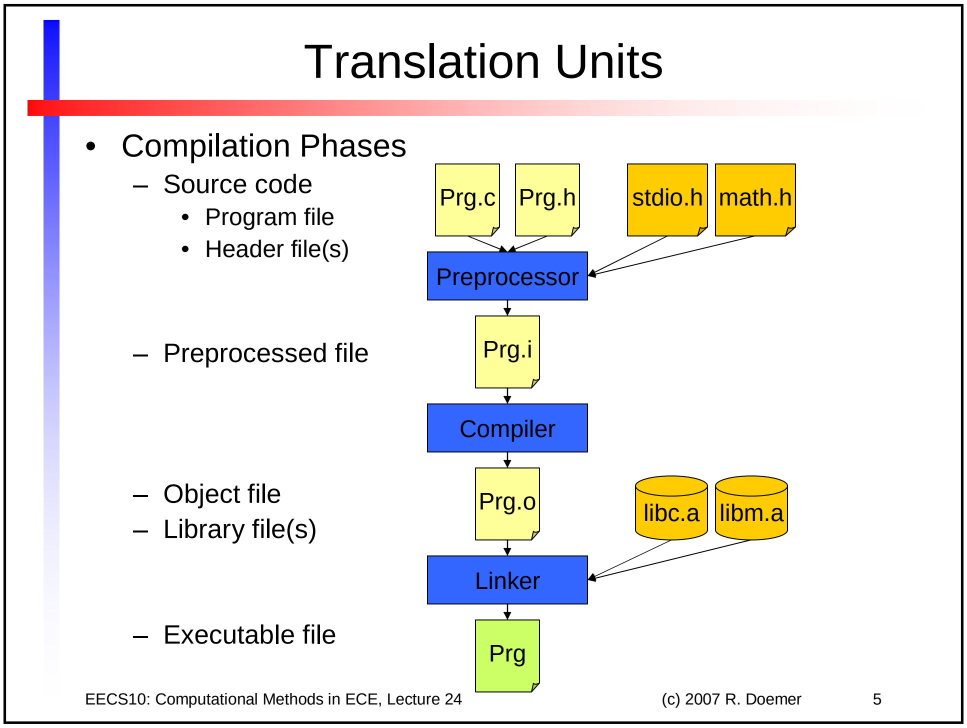
Course Administration

- Final Course Evaluation
  - Open until end of 10th week
  - Nov. 26, 2007, 12pm - Dec. 9, 2007, 11:45pm
  - Online via EEE Evaluation application
- Mandatory Evaluation of Course and Instructor
  - Voluntary
  - Anonymous
  - Very valuable
    - Help to improve this class!
- Please spend 5 minutes!



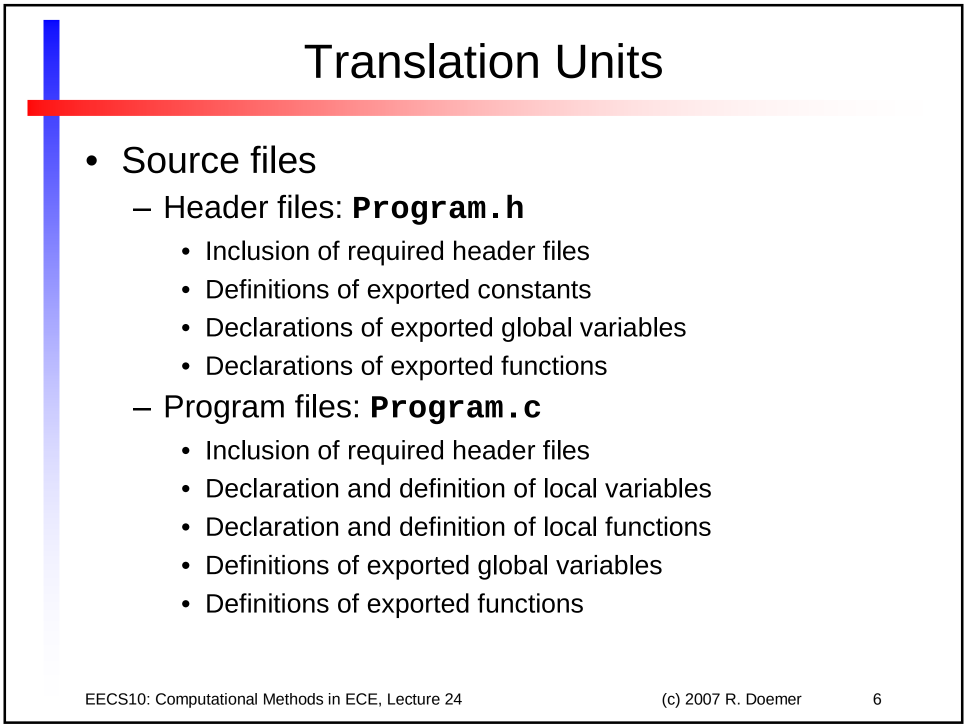
Translation Units

- Introduction
  - C compilation process is a sequence of phases
    - Preprocessing (handle # directives)
    - Scanning and parsing (generate internal data structure)
    - Instruction generation (emit stream of CPU instructions)
    - Assembly (generate binary object file)
    - Linking (combine objects into executable file)
  - C compiler consists of separate components
    - Preprocessor (processes # directives)
    - Compiler (compiles and assembles code)
    - Linker (processes object files and libraries)



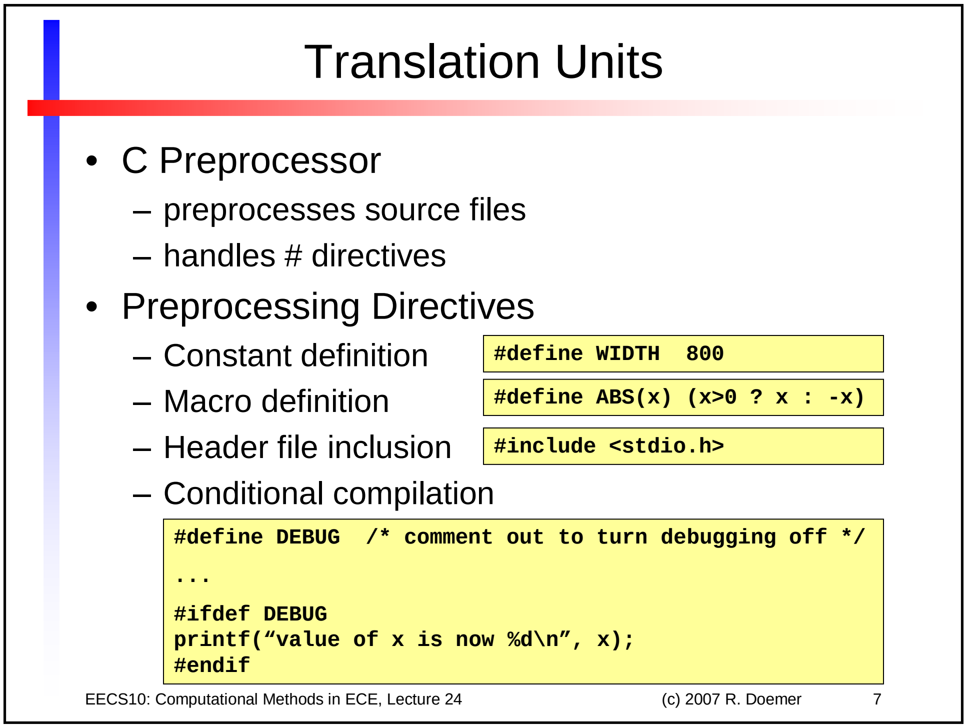
Translation Units

- Compilation Phases
  - Source code
    - Program file
    - Header file(s)
  - Preprocessed file
  - Object file
  - Library file(s)
  - Executable file



Translation Units

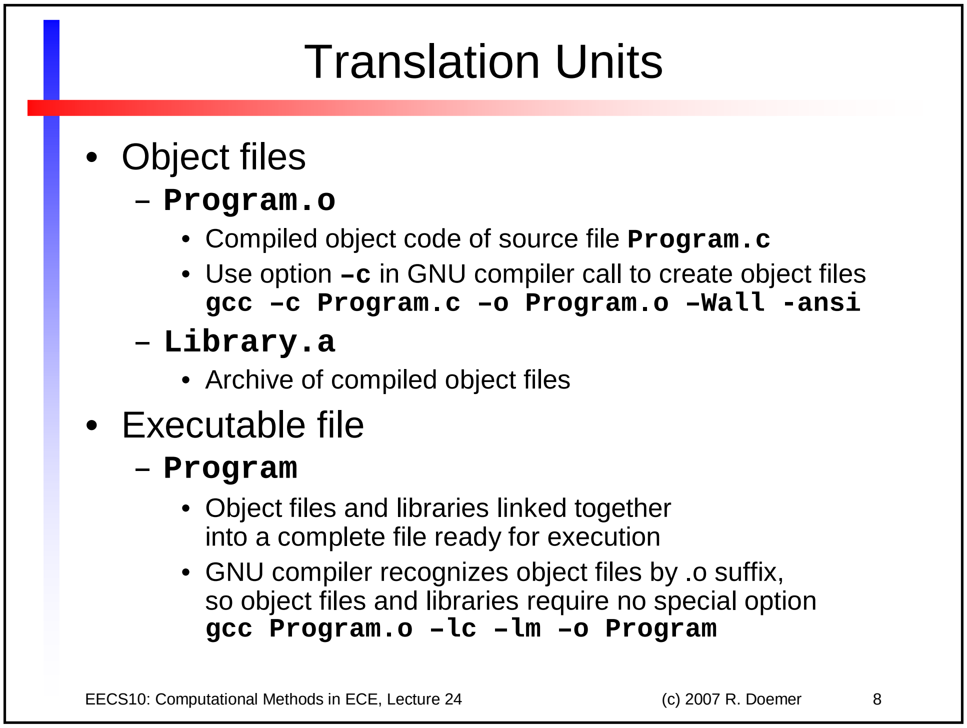
- Source files
  - Header files: **`Program.h`**
    - Inclusion of required header files
    - Definitions of exported constants
    - Declarations of exported global variables
    - Declarations of exported functions
  - Program files: **`Program.c`**
    - Inclusion of required header files
    - Declaration and definition of local variables
    - Declaration and definition of local functions
    - Definitions of exported global variables
    - Definitions of exported functions



Translation Units

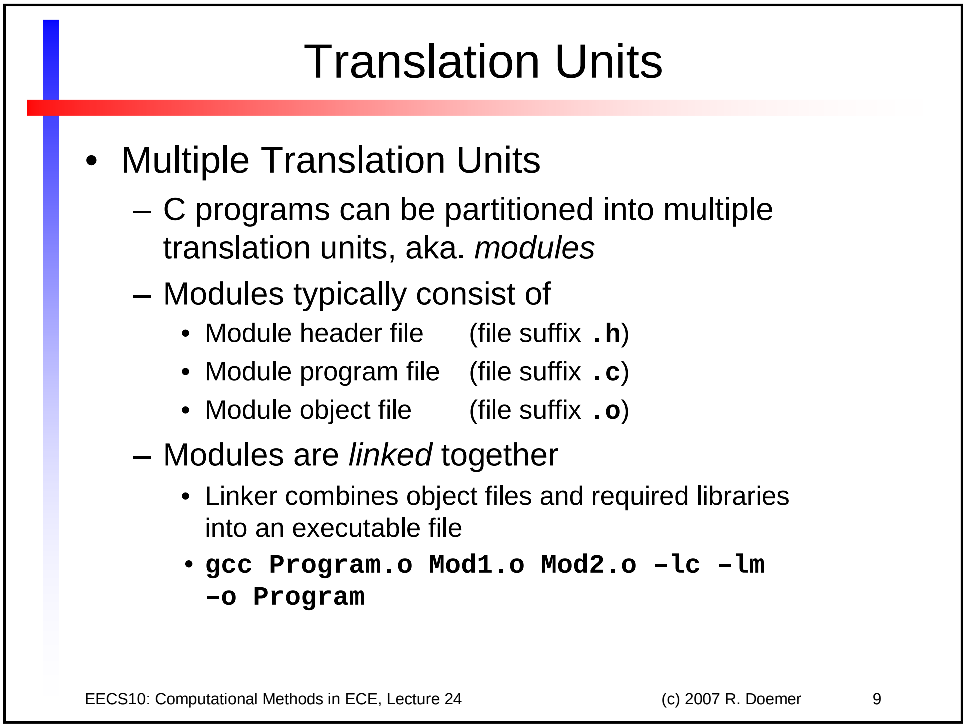
- C Preprocessor
  - preprocesses source files
  - handles # directives
- Preprocessing Directives
  - Constant definition `#define WIDTH  800`
  - Macro definition `#define ABS(x) (x>0 ? x : -x)`
  - Header file inclusion `#include <stdio.h>`
  - Conditional compilation

```
#define DEBUG  /* comment out to turn debugging off */
...
#ifdef DEBUG
printf(“value of x is now %d\n”, x);
#endif
```



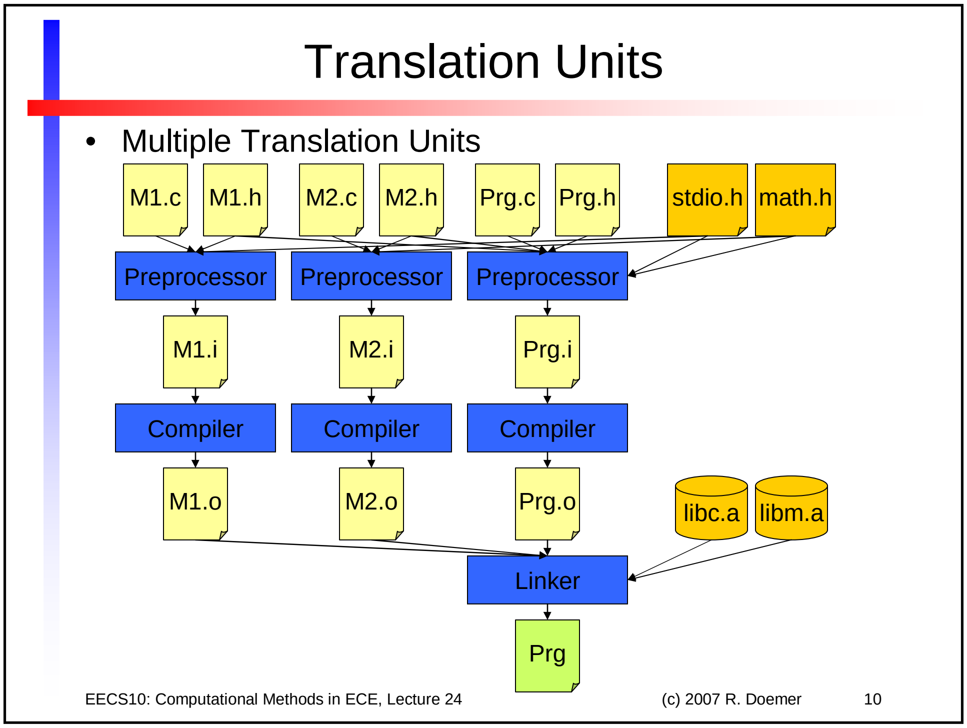
Translation Units

- Object files
  - **`Program.o`**
    - Compiled object code of source file **`Program.c`**
    - Use option **`–c`** in GNU compiler call to create object files
      `gcc –c Program.c –o Program.o –Wall -ansi`
  - **`Library.a`**
    - Archive of compiled object files
- Executable file
  - **`Program`**
    - Object files and libraries linked together
      into a complete file ready for execution
    - GNU compiler recognizes object files by .o suffix,
      so object files and libraries require no special option
      `gcc Program.o –lc –lm –o Program`



Translation Units

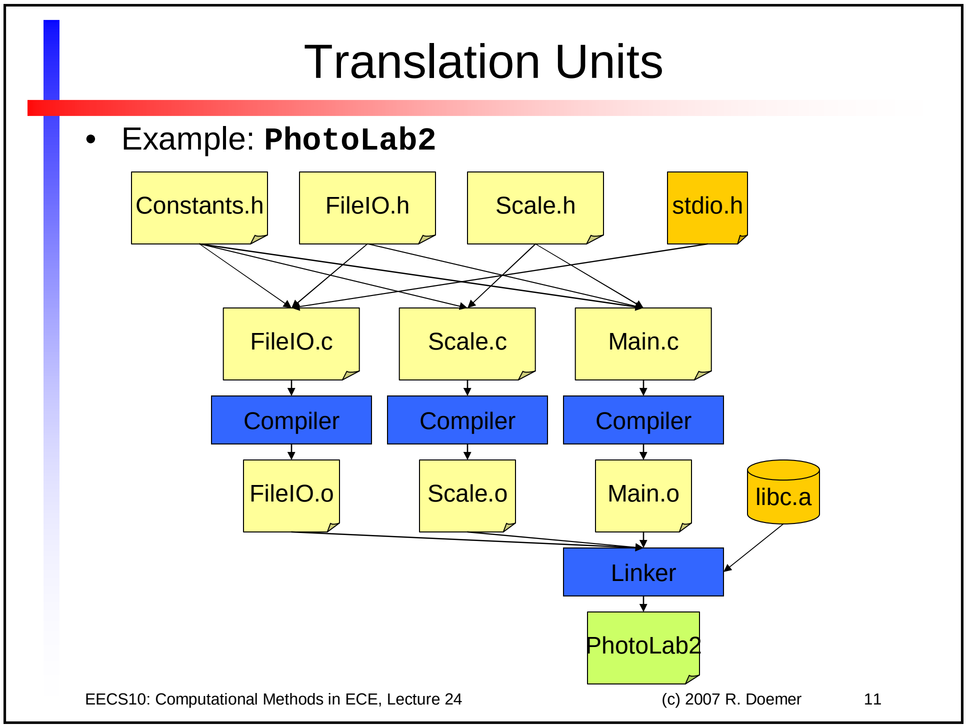
- Multiple Translation Units
  - C programs can be partitioned into multiple translation units, aka. *modules*
  - Modules typically consist of
    - Module header file (file suffix **`.h`**)
    - Module program file (file suffix **`.c`**)
    - Module object file (file suffix **`.o`**)
  - Modules are *linked* together
    - Linker combines object files and required libraries into an executable file
    - **`gcc Program.o Mod1.o Mod2.o –lc –lm –o Program`**



Translation Units

- Multiple Translation Units

M1.c
M1.h
M2.c
M2.h
Prg.c
Prg.h
stdio.h
math.h
Preprocessor
Preprocessor
Preprocessor
M1.i
M2.i
Prg.i
Compiler
Compiler
Compiler
M1.o
M2.o
Prg.o
libc.a
libm.a
Linker
Prg

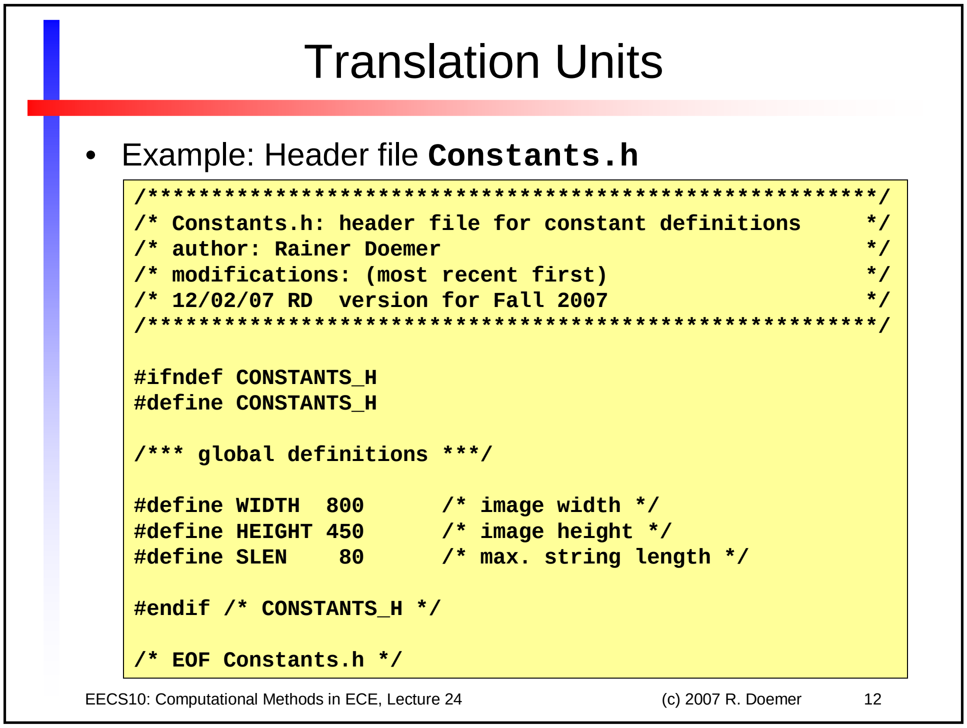


Translation Units

- Example: **PhotoLab2**

Constants.h
FileIO.h
Scale.h
stdio.h
FileIO.c
Scale.c
Main.c
Compiler
Compiler
Compiler
FileIO.o
Scale.o
Main.o
libc.a
Linker
PhotoLab2

EECS10: Computational Methods in ECE, Lecture 24 (c) 2007 R. Doemer 11

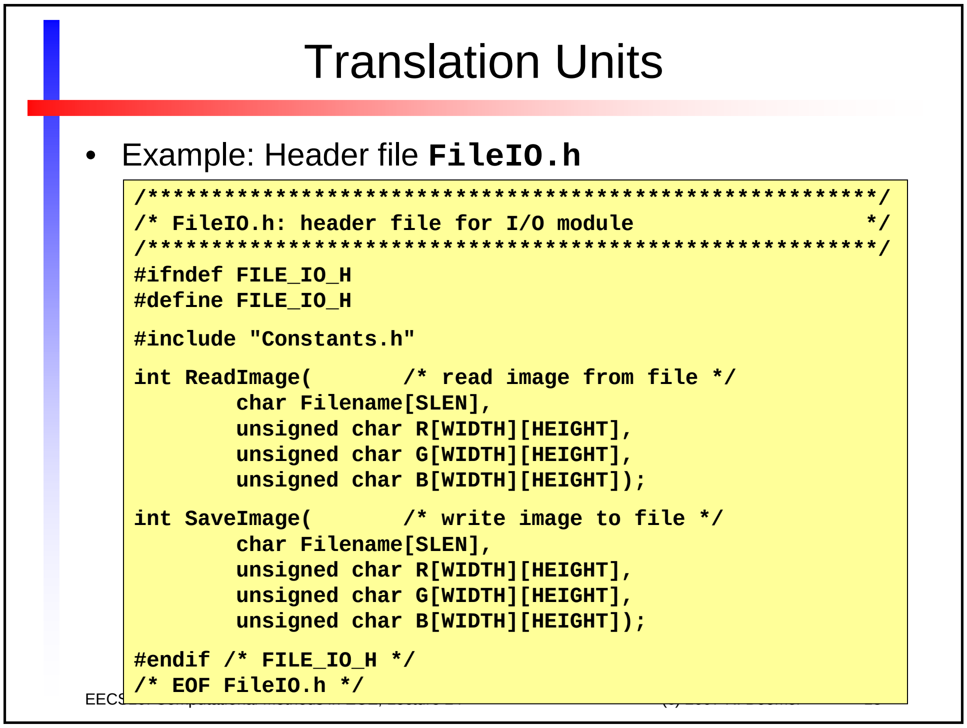


Translation Units

- Example: Header file **Constants.h**

```
/*********************************************************/
/* Constants.h: header file for constant definitions     */
/* author: Rainer Doemer                                 */
/* modifications: (most recent first)                    */
/* 12/02/07 RD  version for Fall 2007                    */
/*********************************************************/

#ifndef CONSTANTS_H
#define CONSTANTS_H

/*** global definitions ***/

#define WIDTH  800      /* image width */
#define HEIGHT 450      /* image height */
#define SLEN    80      /* max. string length */

#endif /* CONSTANTS_H */

/* EOF Constants.h */
```

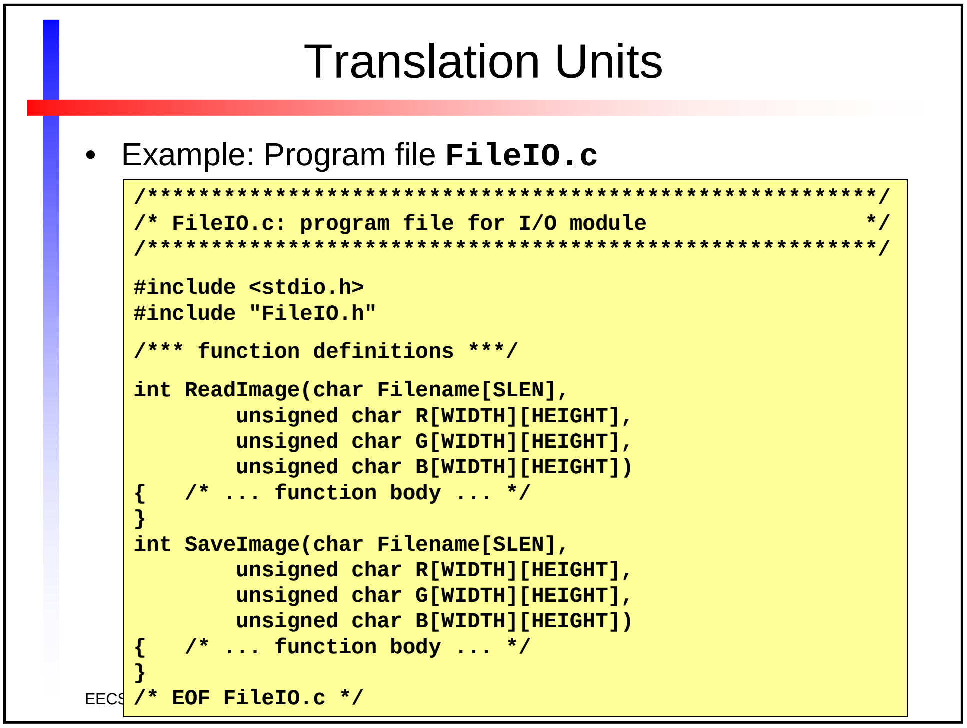
EECS10: Computational Methods in ECE, Lecture 24 (c) 2007 R. Doemer 12



Translation Units

- Example: Header file **`FileIO.h`**

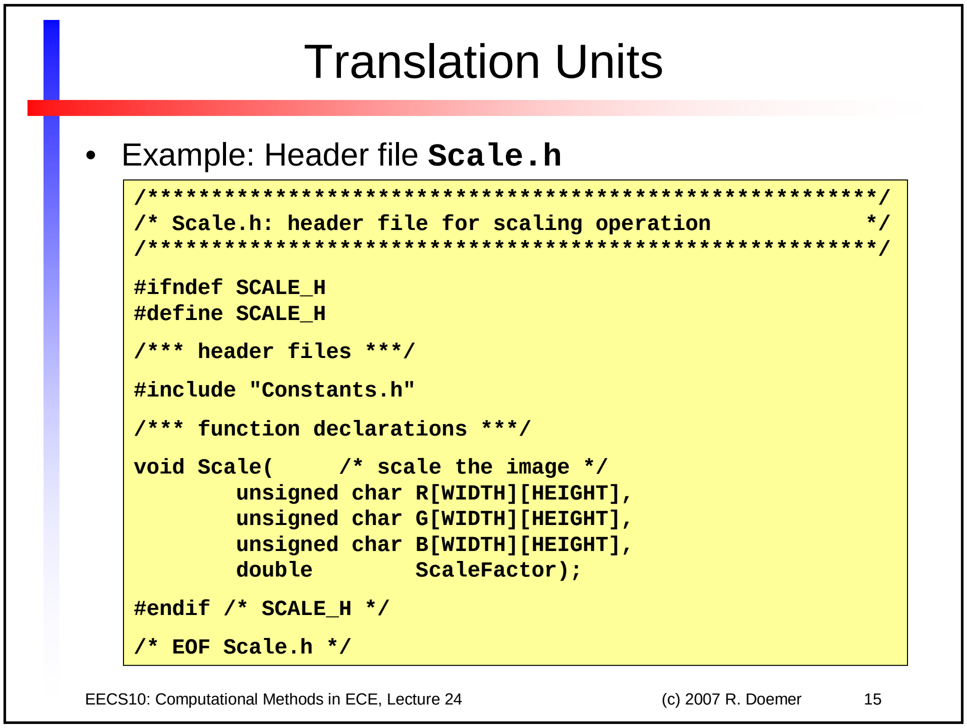
```
/************************************************************/
/* FileIO.h: header file for I/O module                     */
/************************************************************/
#ifndef FILE_IO_H
#define FILE_IO_H

#include "Constants.h"

int ReadImage(       /* read image from file */
        char Filename[SLEN],
        unsigned char R[WIDTH][HEIGHT],
        unsigned char G[WIDTH][HEIGHT],
        unsigned char B[WIDTH][HEIGHT]);

int SaveImage(       /* write image to file */
        char Filename[SLEN],
        unsigned char R[WIDTH][HEIGHT],
        unsigned char G[WIDTH][HEIGHT],
        unsigned char B[WIDTH][HEIGHT]);

#endif /* FILE_IO_H */
/* EOF FileIO.h */
```



Translation Units

- Example: Program file **`FileIO.c`**

```
/************************************************************/
/* FileIO.c: program file for I/O module                    */
/************************************************************/

#include <stdio.h>
#include "FileIO.h"

/*** function definitions ***/

int ReadImage(char Filename[SLEN],
        unsigned char R[WIDTH][HEIGHT],
        unsigned char G[WIDTH][HEIGHT],
        unsigned char B[WIDTH][HEIGHT])
{   /* ... function body ... */
}
int SaveImage(char Filename[SLEN],
        unsigned char R[WIDTH][HEIGHT],
        unsigned char G[WIDTH][HEIGHT],
        unsigned char B[WIDTH][HEIGHT])
{   /* ... function body ... */
}
/* EOF FileIO.c */
```

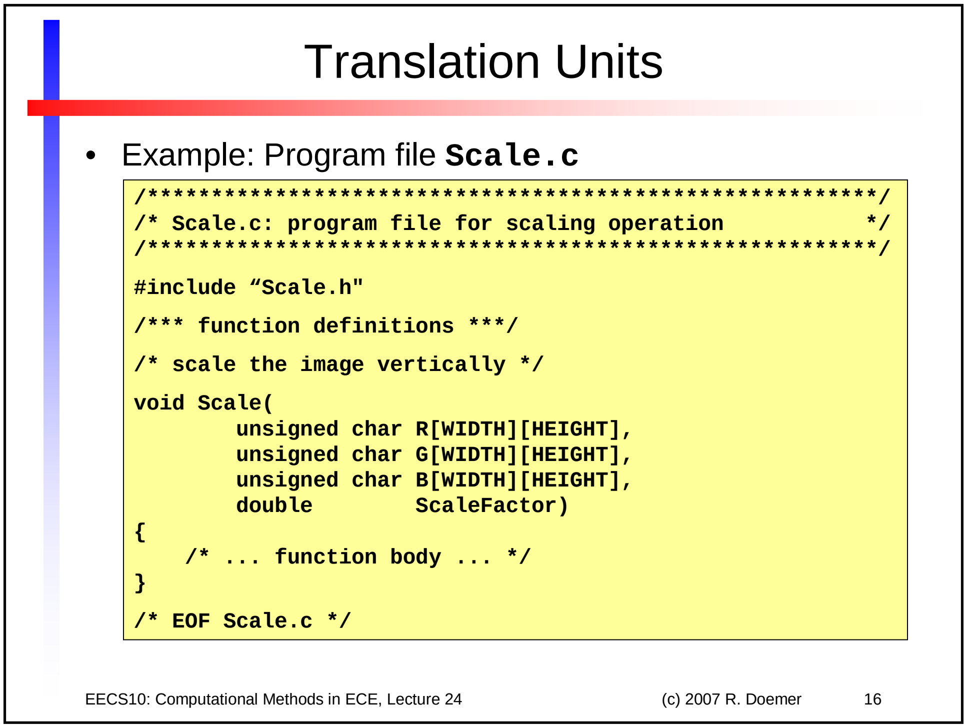


Translation Units

- Example: Header file **`Scale.h`**

```
/*********************************************************/
/* Scale.h: header file for scaling operation            */
/*********************************************************/

#ifndef SCALE_H
#define SCALE_H

/*** header files ***/

#include "Constants.h"

/*** function declarations ***/

void Scale(     /* scale the image */
        unsigned char R[WIDTH][HEIGHT],
        unsigned char G[WIDTH][HEIGHT],
        unsigned char B[WIDTH][HEIGHT],
        double        ScaleFactor);

#endif /* SCALE_H */

/* EOF Scale.h */
```

EECS10: Computational Methods in ECE, Lecture 24 (c) 2007 R. Doemer 15

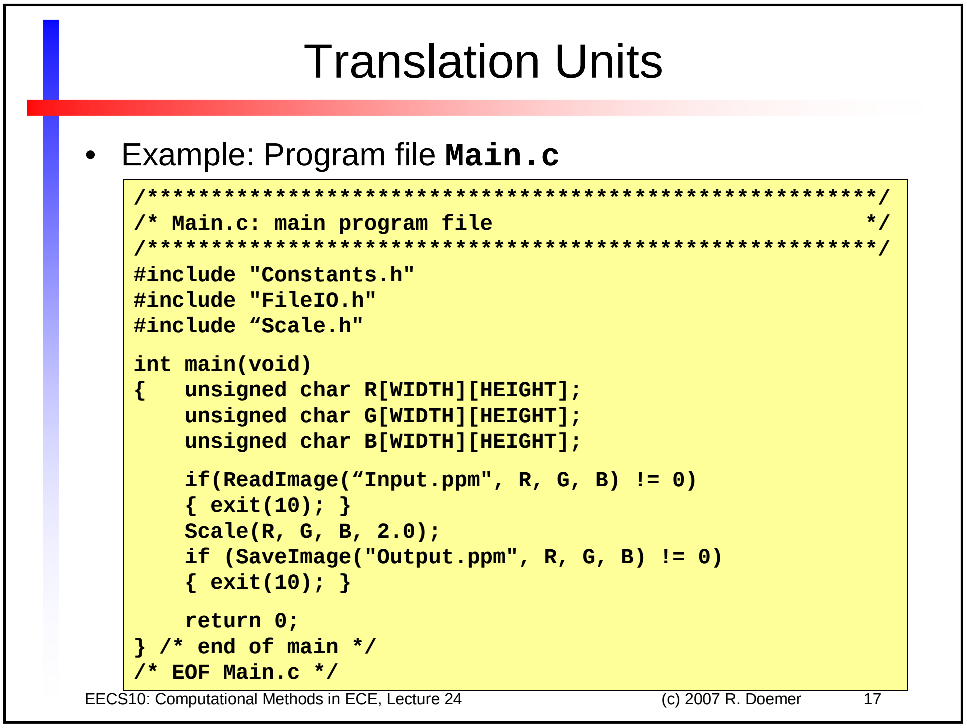


Translation Units

- Example: Program file **`Scale.c`**

```
/*********************************************************/
/* Scale.c: program file for scaling operation           */
/*********************************************************/

#include “Scale.h"

/*** function definitions ***/

/* scale the image vertically */

void Scale(
        unsigned char R[WIDTH][HEIGHT],
        unsigned char G[WIDTH][HEIGHT],
        unsigned char B[WIDTH][HEIGHT],
        double        ScaleFactor)
{
    /* ... function body ... */
}

/* EOF Scale.c */
```

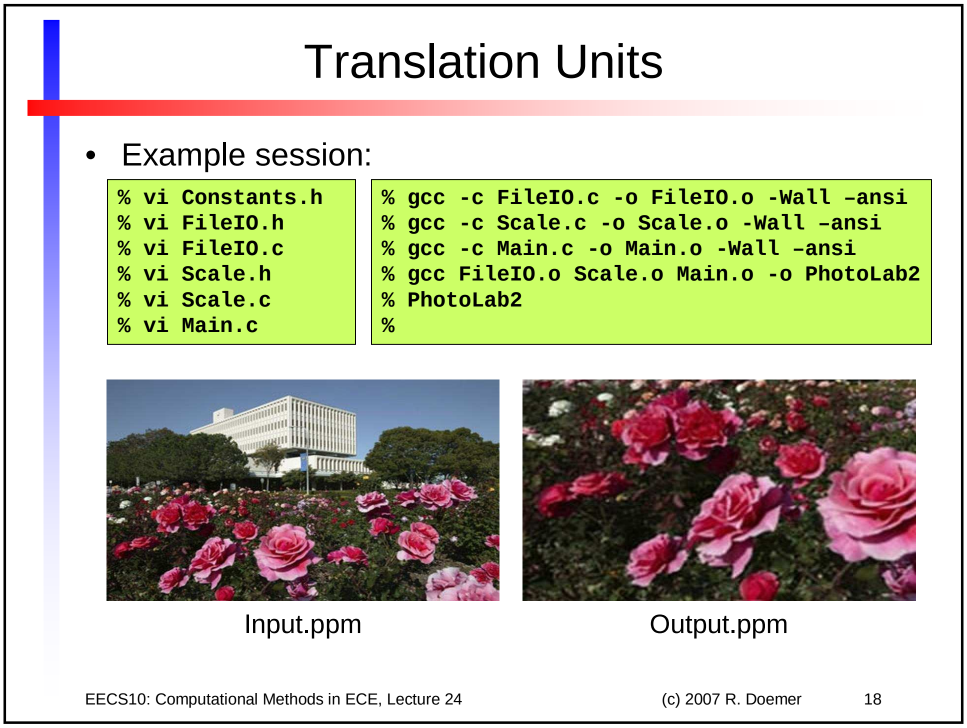
EECS10: Computational Methods in ECE, Lecture 24 (c) 2007 R. Doemer 16



Translation Units

- Example: Program file **`Main.c`**

```
/**********************************************************/
/* Main.c: main program file                              */
/**********************************************************/
#include "Constants.h"
#include "FileIO.h"
#include “Scale.h"

int main(void)
{   unsigned char R[WIDTH][HEIGHT];
    unsigned char G[WIDTH][HEIGHT];
    unsigned char B[WIDTH][HEIGHT];

    if(ReadImage(“Input.ppm", R, G, B) != 0)
    { exit(10); }
    Scale(R, G, B, 2.0);
    if (SaveImage("Output.ppm", R, G, B) != 0)
    { exit(10); }

    return 0;
} /* end of main */
/* EOF Main.c */
```



Translation Units

- Example session:

```
% vi Constants.h
% vi FileIO.h
% vi FileIO.c
% vi Scale.h
% vi Scale.c
% vi Main.c
```

```
% gcc -c FileIO.c -o FileIO.o -Wall –ansi
% gcc -c Scale.c -o Scale.o -Wall –ansi
% gcc -c Main.c -o Main.o -Wall –ansi
% gcc FileIO.o Scale.o Main.o -o PhotoLab2
% PhotoLab2
%
```

Input.ppm

Output.ppm