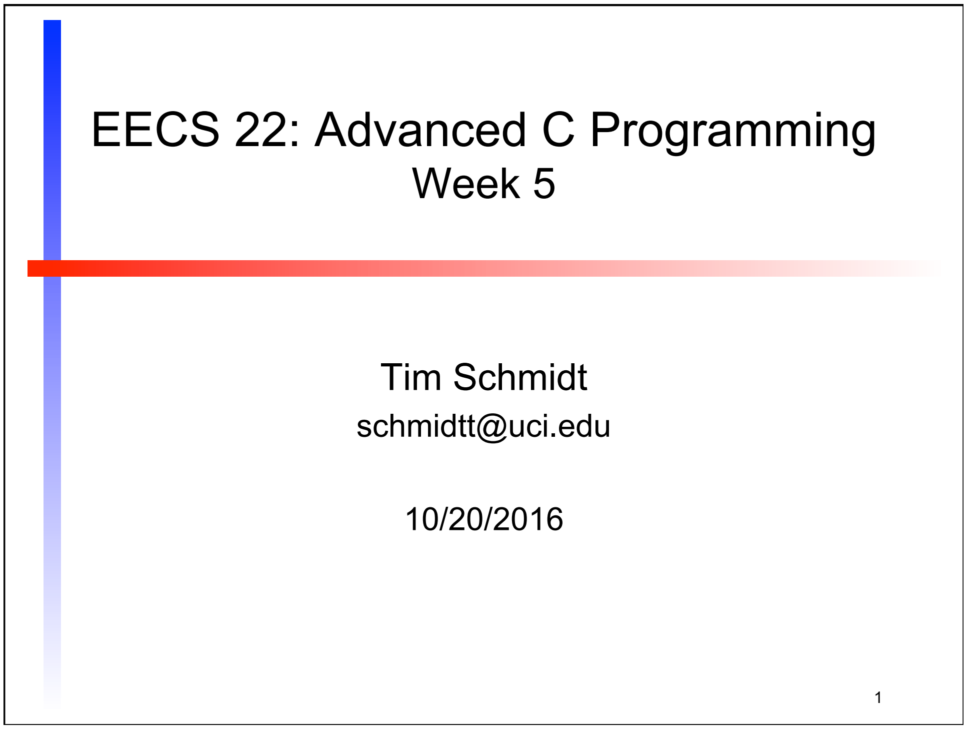
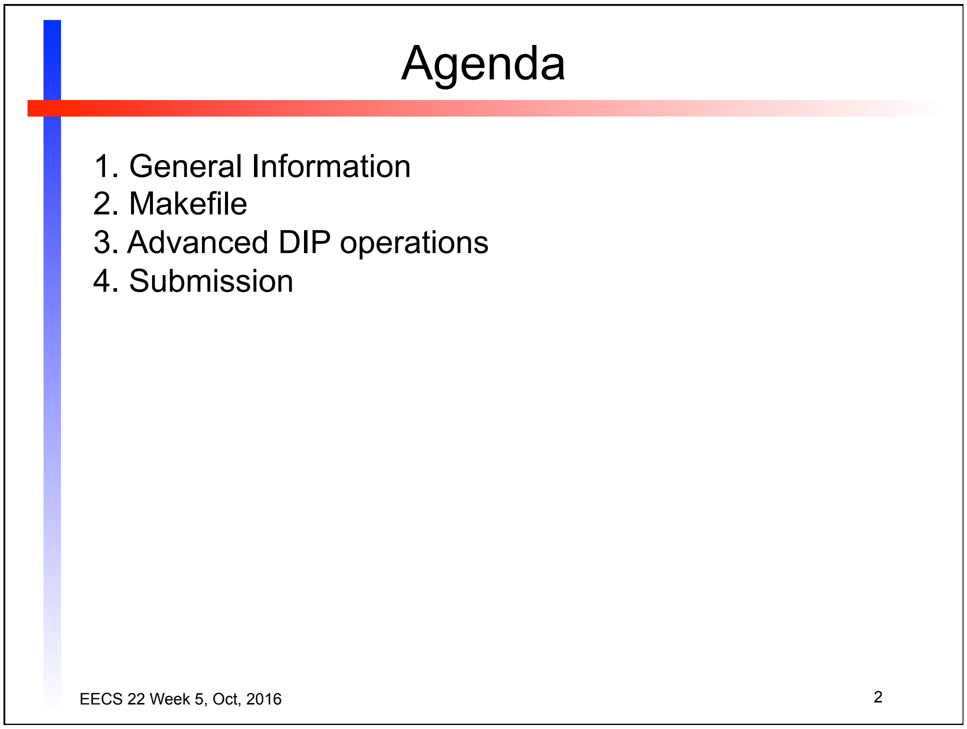


EECS 22: Advanced C Programming
Week 5

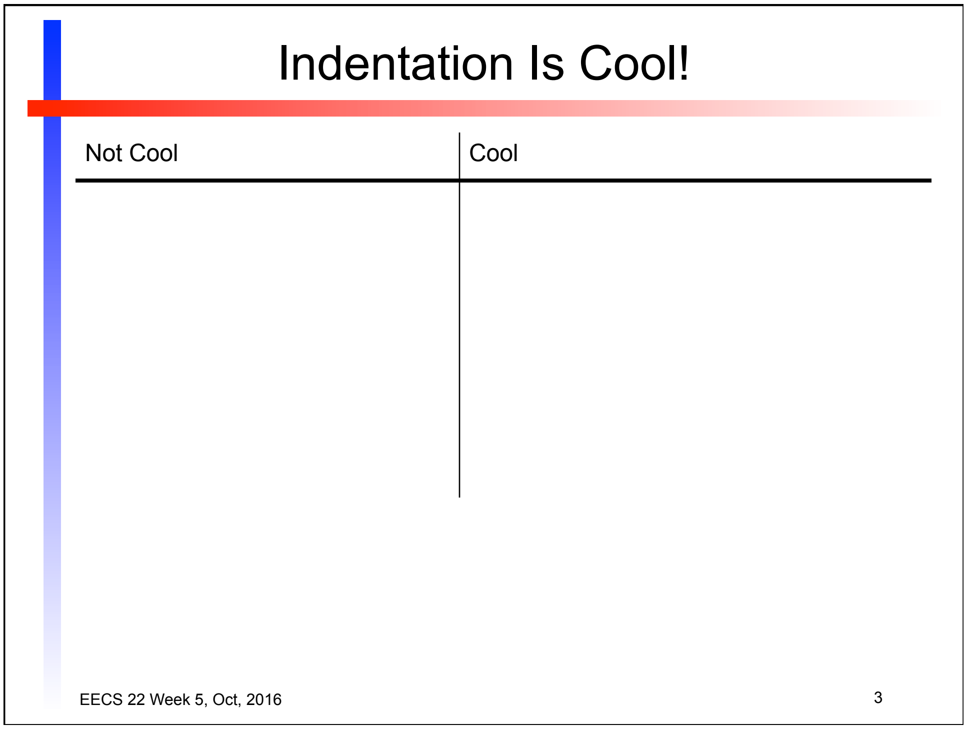
Tim Schmidt
schmidtt@uci.edu

10/20/2016



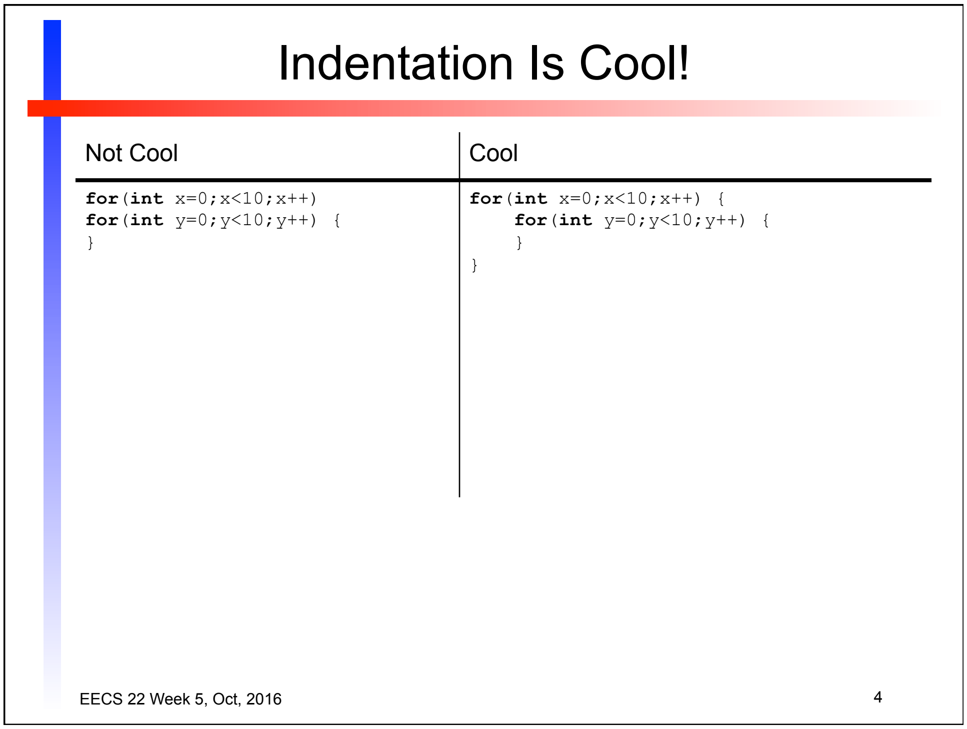
Agenda

1. General Information
2. Makefile
3. Advanced DIP operations
4. Submission



Indentation Is Cool!

Not Cool	Cool



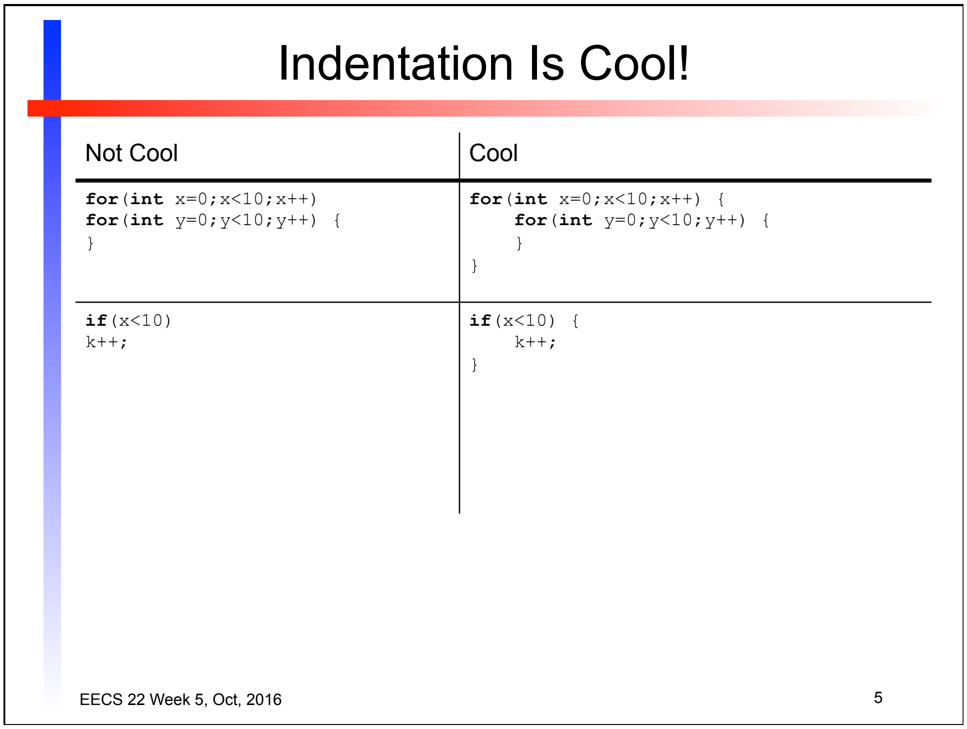
Indentation Is Cool!

Not Cool

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

Cool

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

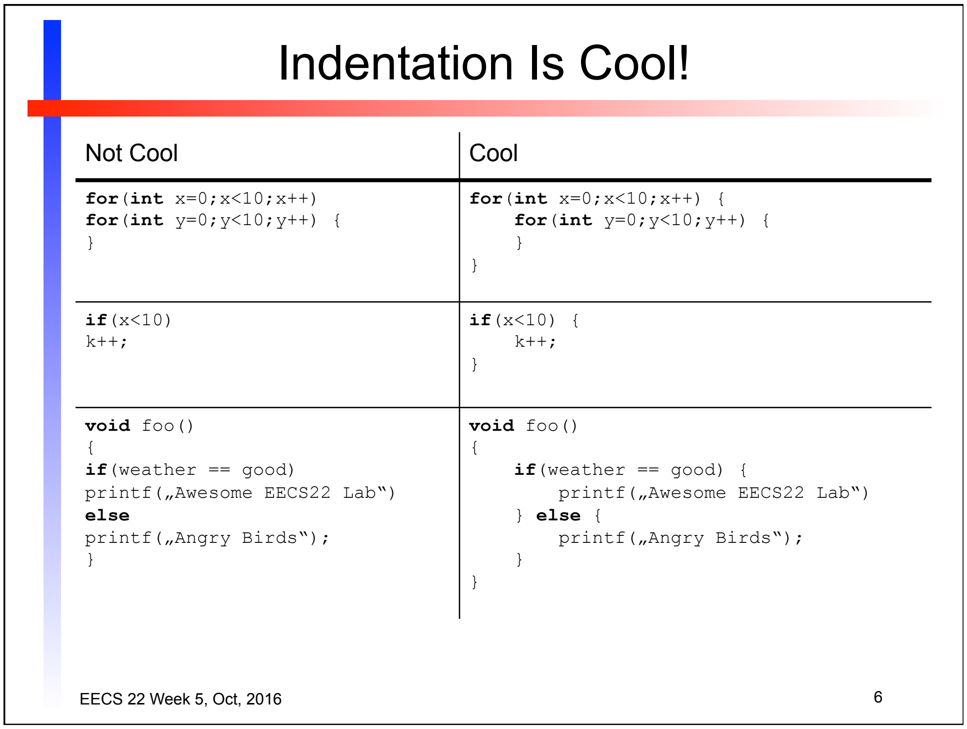


Indentation Is Cool!

Not Cool	Cool

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

`if(x<10)`
`k++;`
	`if(x<10) {`
	`    k++;`
	`}`



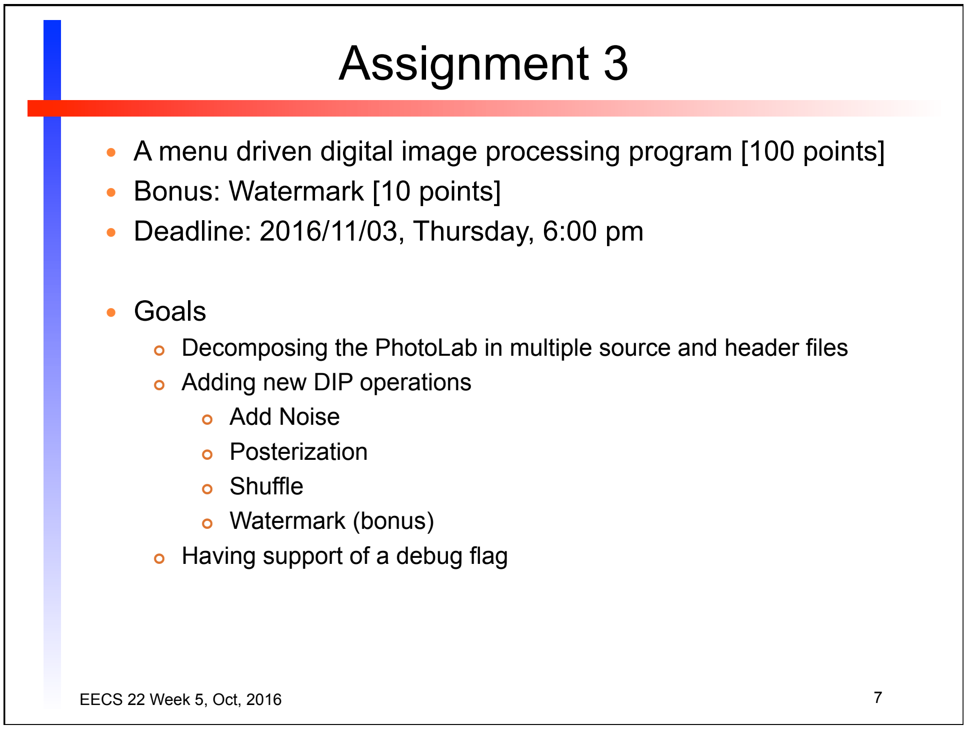
Indentation Is Cool!

Not Cool	Cool

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

`if(x<10)`
`k++;`
	`if(x<10) {`
	`    k++;`
	`}`

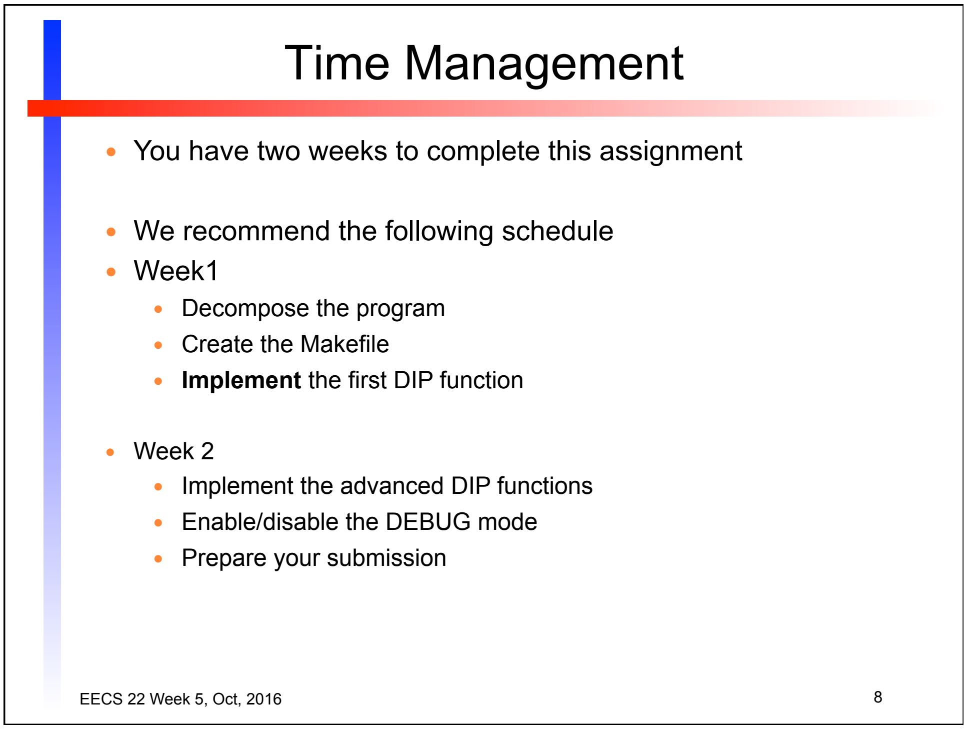
`void foo()`
`{`
`if(weather == good)`
`printf(„Awesome EECS22 Lab“)`
`else`
`printf(„Angry Birds“);`
`}`
	`void foo()`
	`{`
	`    if(weather == good) {`
	`        printf(„Awesome EECS22 Lab“)`
	`    } else {`
	`        printf(„Angry Birds“);`
	`    }`
	`}`



Assignment 3

- A menu driven digital image processing program [100 points]
- Bonus: Watermark [10 points]
- Deadline: 2016/11/03, Thursday, 6:00 pm

- Goals
  - Decomposing the PhotoLab in multiple source and header files
  - Adding new DIP operations
    - Add Noise
    - Posterization
    - Shuffle
    - Watermark (bonus)
  - Having support of a debug flag

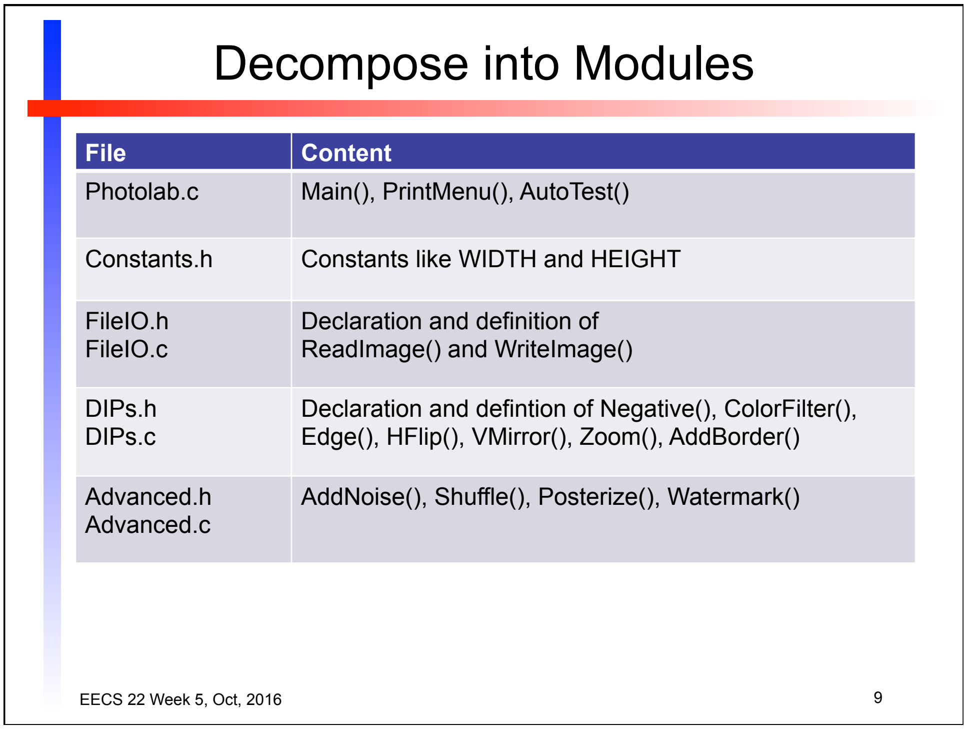


Time Management

- You have two weeks to complete this assignment

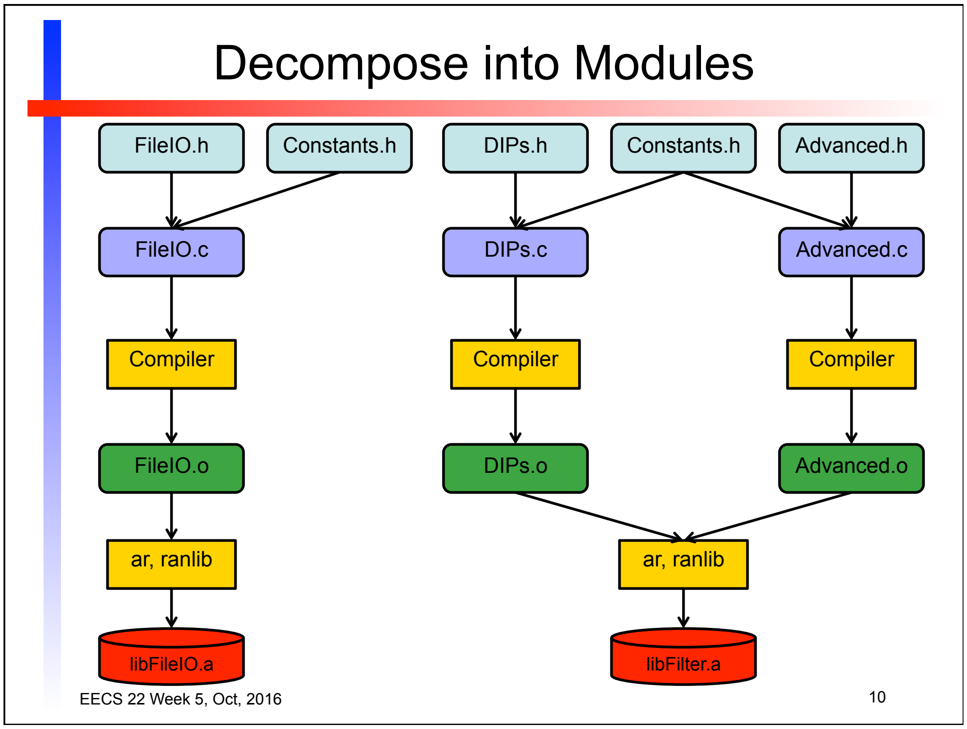
- We recommend the following schedule
- Week1
  - Decompose the program
  - Create the Makefile
  - **Implement** the first DIP function

- Week 2
  - Implement the advanced DIP functions
  - Enable/disable the DEBUG mode
  - Prepare your submission



Decompose into Modules

File	Content
Photolab.c	Main(), PrintMenu(), AutoTest()
Constants.h	Constants like WIDTH and HEIGHT
FileIO.h FileIO.c	Declaration and definition of ReadImage() and WriteImage()
DIPs.h DIPs.c	Declaration and defintion of Negative(), ColorFilter(), Edge(), HFlip(), VMirror(), Zoom(), AddBorder()
Advanced.h Advanced.c	AddNoise(), Shuffle(), Posterize(), Watermark()



Decompose into Modules

FileIO.h

Constants.h

DIPs.h

Constants.h

Advanced.h

FileIO.c

DIPs.c

Advanced.c

Compiler

Compiler

Compiler

FileIO.o

DIPs.o

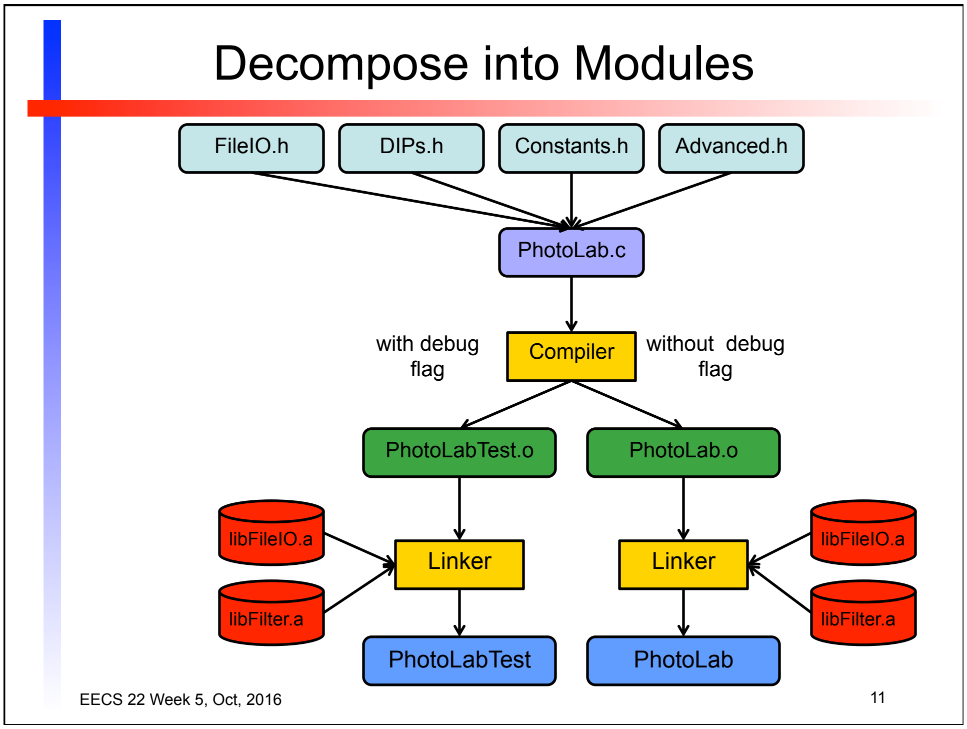
Advanced.o

ar, ranlib

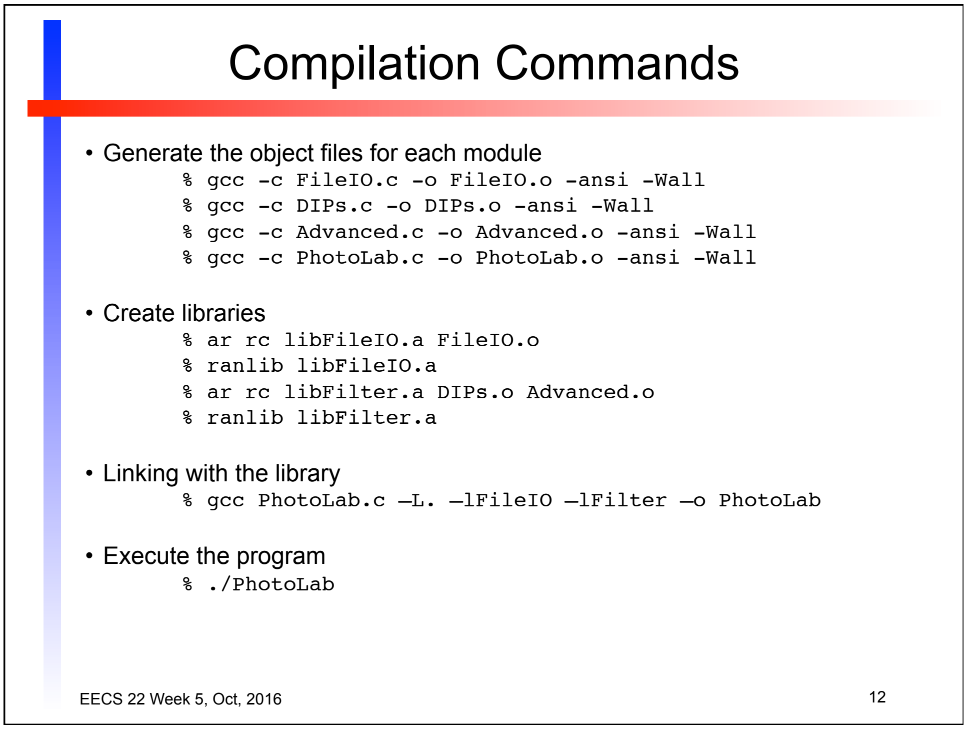
ar, ranlib

libFileIO.a

libFilter.a



Decompose into Modules



Compilation Commands

- Generate the object files for each module

```
% gcc -c FileIO.c -o FileIO.o -ansi -Wall
% gcc -c DIPs.c -o DIPs.o -ansi -Wall
% gcc -c Advanced.c -o Advanced.o -ansi -Wall
% gcc -c PhotoLab.c -o PhotoLab.o -ansi -Wall
```

- Create libraries

```
% ar rc libFileIO.a FileIO.o
% ranlib libFileIO.a
% ar rc libFilter.a DIPs.o Advanced.o
% ranlib libFilter.a
```

- Linking with the library

```
% gcc PhotoLab.c –L. –lFileIO –lFilter –o PhotoLab
```

- Execute the program

```
% ./PhotoLab
```

Makefile

Your Makefile must have the following targets:

all: the target to generate the executable programs

clean: the target to clean all the intermediate files,
e.g. object files, auto generated images,
and the executable programs

PhotoLabTest: the target to create and run PhotoLabTest

PhotoLab: the target to generate the executable program PhotoLab

Advanced DIP Operations

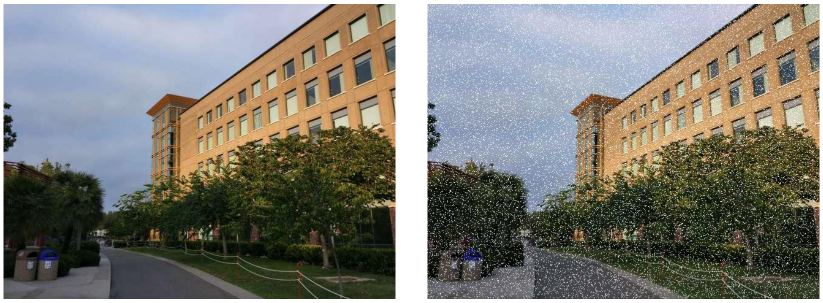
New Operations

- AddNoise
- Shuffle
- Posterize the image
- Watermark (bonus)

The menu looks like:

```
--------------------------------
1:  Load a PPM image
2:  Save an image in PPM and JPEG format
3:  Make a negative of an image
4:  Color filter an image
5:  Sketch the edge of an image
6:  Flip an image horizontally
7:  Mirror an image vertically
8:  Add Border to an image
9:  Zoom an image
10: Add noise to an image
11: Shuffle an image
12: Posterize an image
13: Watermark
14: Test all functions
15: Exit
please make your choice:
```

AddNoise



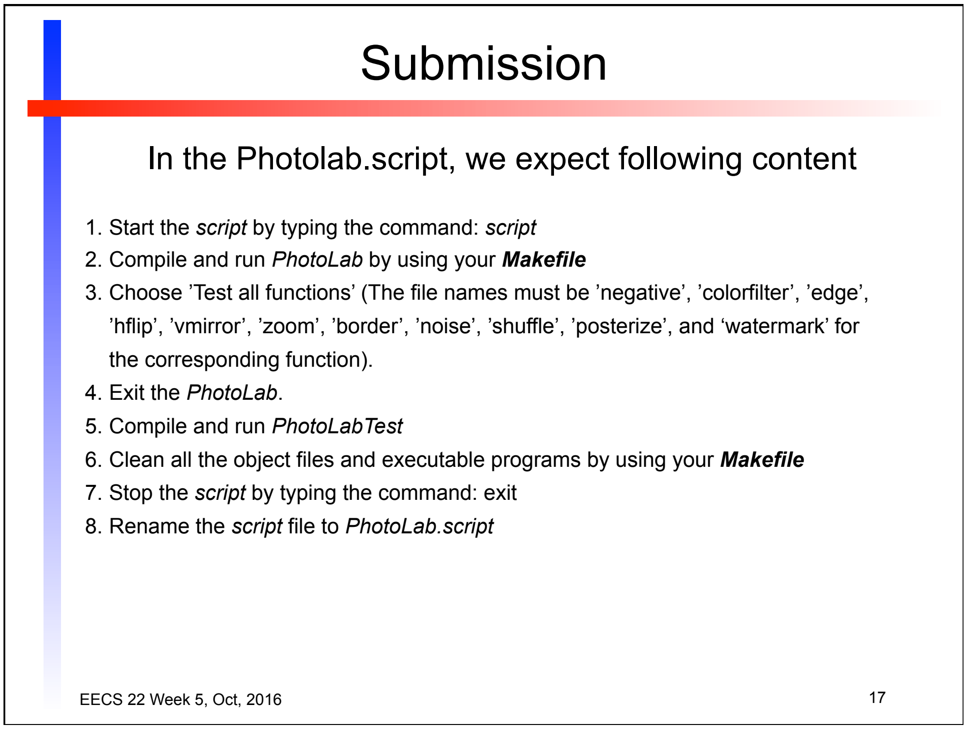
```
void AddNoise(
          int n,
          unsigned char R[WIDTH][HEIGHT],
          unsigned char G[WIDTH][HEIGHT],
          unsigned char B[WIDTH][HEIGHT]);
```

- n: defines the percentage of white pixels
- Functions rand() and srand() should be used to generate random white spots
- Each execution creates a different result

Submission

The submission should include these files

- PhotoLab.c
- PhotoLab.script
- PhotoLab.txt
- FileIO.c
- FileIO.h
- Constants.h
- DIPs.c
- DIPs.h
- Advanced.c
- Advanced.h
- Makefile



Submission

In the Photolab.script, we expect following content

1. Start the *script* by typing the command: *script*
2. Compile and run *PhotoLab* by using your ***Makefile***
3. Choose ’Test all functions’ (The file names must be ’negative’, ’colorfilter’, ’edge’, ’hflip’, ’vmirror’, ’zoom’, ’border’, ’noise’, ’shuffle’, ’posterize’, and ‘watermark’ for the corresponding function).
4. Exit the *PhotoLab*.
5. Compile and run *PhotoLabTest*
6. Clean all the object files and executable programs by using your ***Makefile***
7. Stop the *script* by typing the command: exit
8. Rename the *script* file to *PhotoLab.script*