

ECPS 203

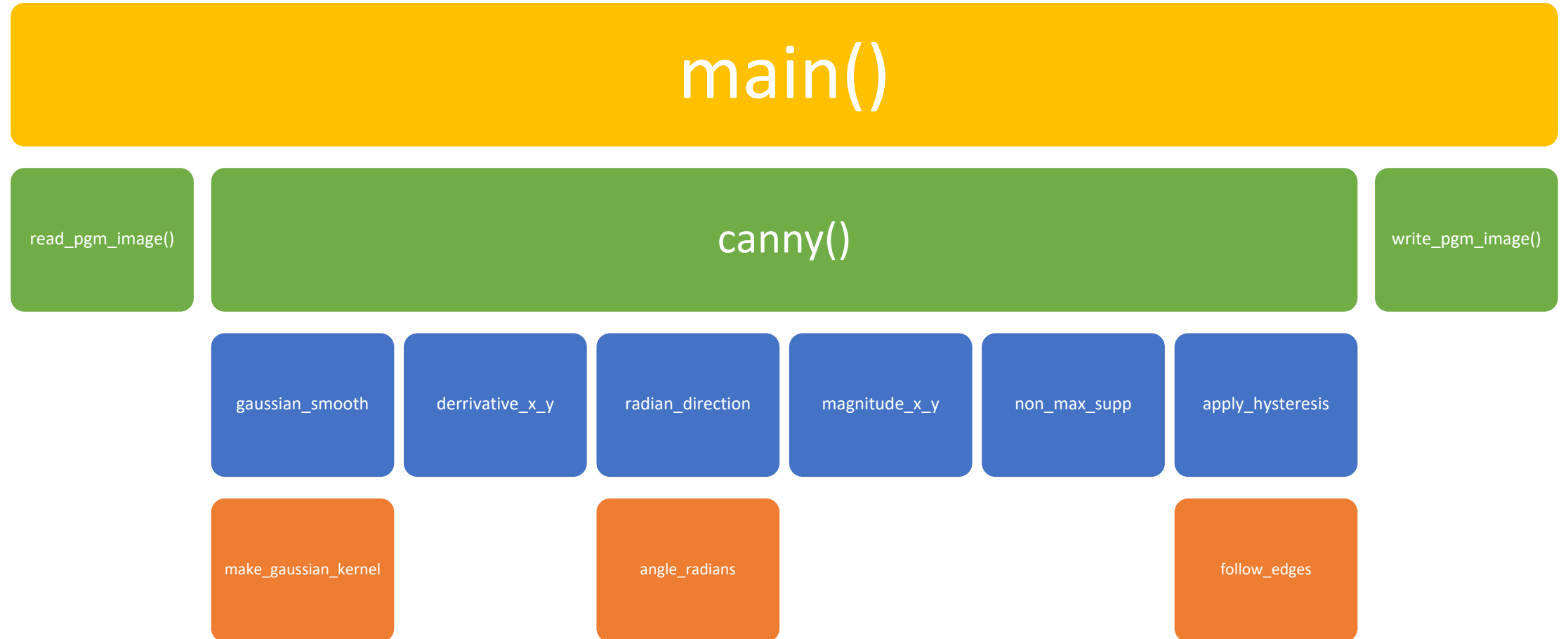
Discussion

TA: Zhongqi Cheng

Agenda

1. Review of Assignment 1
2. Assignment 2

Function call hierarchy



Reference code for assignment 1

- can be found in `~ecps203/public/canny.c`
- use this code as the foundation of assignment 2

Reminders

- Name your homework directory in lower-cased fashion
 - .../hw/hw1
 - otherwise the submission script cant find your homework
- Run your code on the server, not your local machine



Assignment 2

due: Wednesday 6:00 p.m. next week

Assignment 2

Tasks:

1. Rewrite C code into C++
2. Clean up codes
3. Hardcode parameters
4. Remove dynamic memory allocation

C and C++

- C function definition can have no return type
- C++ function definition must have return type

C style

```
foo(){  
    print("hello world\n");  
}
```

C++ style

```
void foo(){  
    print("hello world\n");  
}
```

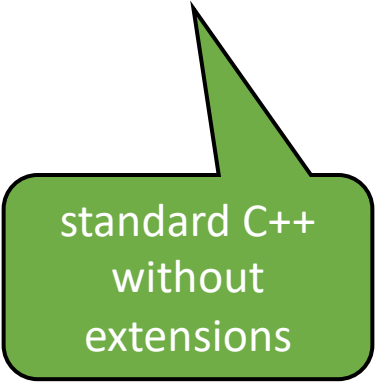

Clean Up Code

- Compile with

```
g++ -Wall -pedantic -O2
```



detect all
warnings



standard C++
without
extensions

Hardcode parameters

- In assignment 1, certain parameters are adjustable and passed as arguments to function calls
 - `canny (image, rows, cols, ...)`
- Now, hardcode them in all functions
 1. `rows = 240`
 2. `cols = 320`
 3. ...

Remove dynamic memory allocations

- In assignment 1, we use **calloc(rows*cols, ...)**
- Now, since all the parameters are hardcoded, we can statically create arrays of size = $240*320$

Submission

- run the script in your command line: `~ecps203/bin/turnin.sh`
 1. `canny.cpp`: source code
 2. `canny.txt`: describe your efforts or troubles