Objectives - 1/11/2022 (Tue)

- Get acquainted with ECE 26400.
  - What you will learn
  - Programming environment: ecegrid
  - Homework
  - Resources: lab help, office hours, reference sheet
  - Exams & quizzes
  - Policies: grades, code quality, base requirements, attendance, communication, academic integrity

Note: Syllabus takes precedence over information in these notes in case of inadvertent inconsistency. Please read the syllabus carefully.

Learning objectives

- Recursion
- Files
- Structures
- Dynamic structures

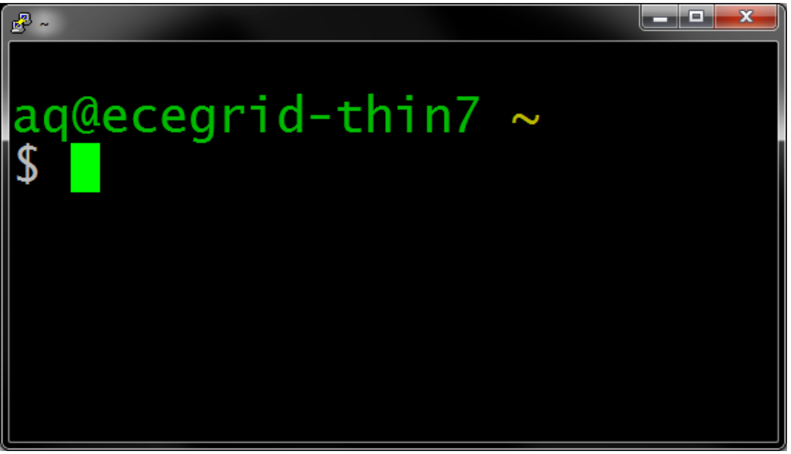
This is the bare minimum that the course must cover.

What you will learn

- C language
- Memory management
- Data structures (linked lists, trees, …)
- Software engineering
  - Build bigger programs → Development tools
  - Bug avoidance → Test driven development (TDD)
  - Testing → Code quality
  - Debugging → Debugging methods
- Data compression

Programming maturity

PuTTY or ssh



```
aq@ecegrid-thin7 ~
$ 
```

SSH

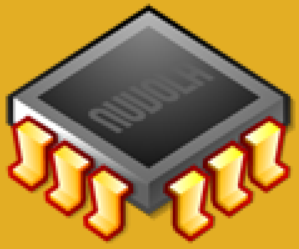
ecegrid

Get starter files

Tools: vim, gcc, gdb, valgrind

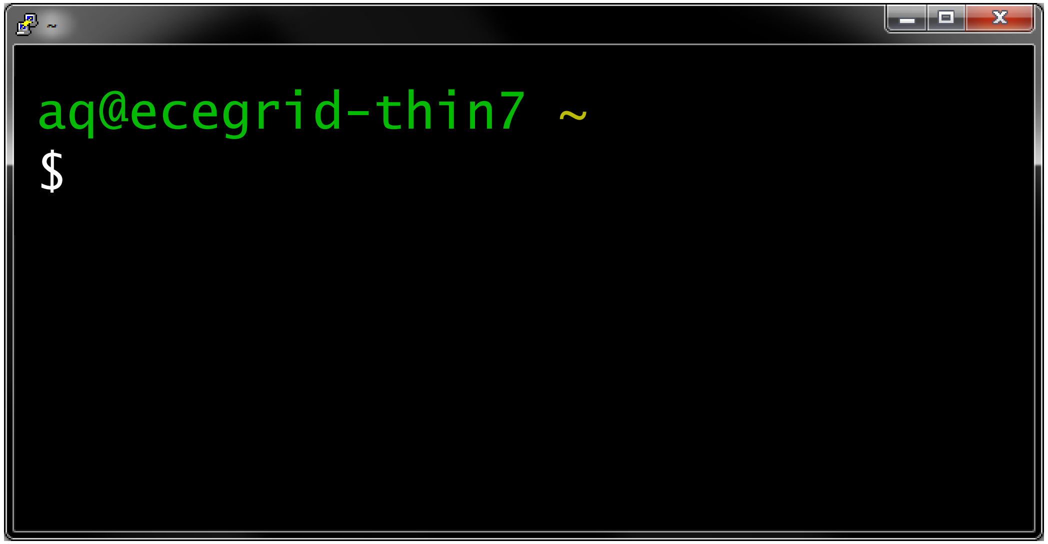
Test your code

Submit homework



Programming environment

- ecegrid via SSH (PuTTY or ssh in terminal)
- Linux
- ISO C11
- Tools: Vim 7.4 + GCC 8.3 + GDB 7.1 + Valgrind
- Compile with flags given in syllabus



```
aq@ecegrid-thin7 ~
$
```

Homework

□ Get an assignment

```
264get hw▒▒
```

□ Submit

```
264submit hw▒▒ ▒▒.c ▒▒.c
```

□ Pre-test (optional)

```
264test hw▒▒
```

Submission

```
264submit hw▒▒ ▒▒.c ▒▒.c
```

- Submit often and early
- You do not need to be anywhere near finished
- Submission creates a backup of your work
- To restore

```
264get --help
```

… and the follow directions

Pre-test

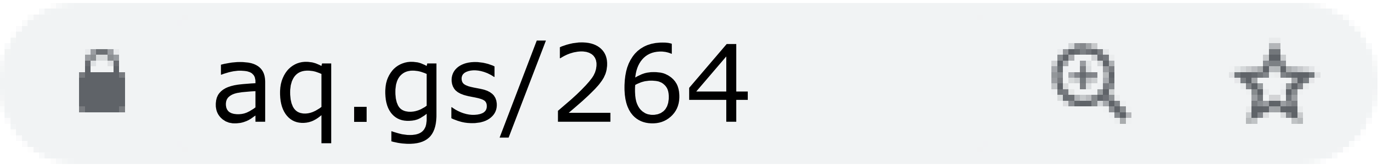
```
264test hw▓▓
```

- Starting with HW02
  - HW01 is special; see description
- Use only after your own testing determines you are done
- Pretester is optional and never necessary
- Typically available ≈2 days before deadline
- “Best effort” → not guaranteed
- Works on your most recent submission
- May not catch everything; expect some “false negatives”

Resources

- Reference sheet
- Lab hours: Discord
- Office hours: Zoom
- Piazza
- Web site: aq.gs/264

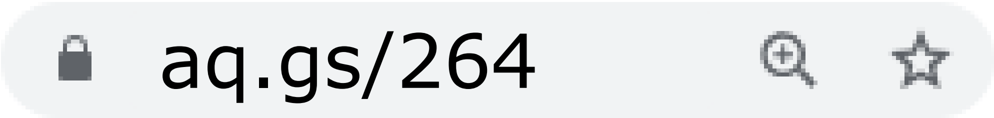
There may be changes. Watch your email.



aq.gs/264

Grading

- □ Homework: 60%
  - Weighted by the number of calendar days given, except where otherwise noted
  - Extending a deadline will increase its weight, except where otherwise noted
  - HW01 will be weighted as "4-day assignment"
- □ Quizzes: 20%
- □ Final exam: 20%
- □ + Bonus
- □ + Participation



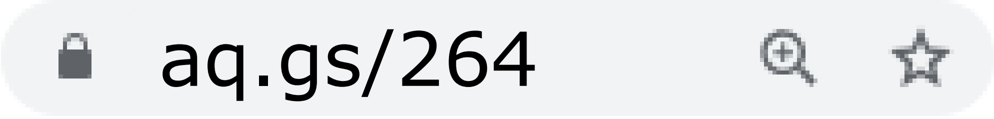
aq.gs/264

There will be no exams or quizzes because I am

Code quality

- Write clean code from the start
  - Prevents some types of bugs
  - Makes other bugs easier to find
  - Helps you understand code in the morning
- Enables others to help you
  - Course staff will not assist with sloppy code
- Read Code Quality Standards
  - −2% per rule violated (to the extent that we can detect)

Writing clean code is an acquired skill.



aq.gs/264

Code quality

- Write clean code from the start
  - Prevents some types of bugs
  - Makes other bugs easier to find
  - Helps you understand code in the morning
- Enables others to help you
  - Course staff will not assist with sloppy code
- Read Code Quality Standards
  - −2% per rule violated

Writing clean code is an acquired skill.

aq.gs/264

⚠

⚠ Base requirements

- All required files must be included in a **single submission.**
- All required files must be **named exactly as specified** in the assignment description.
- **Code can be compiled (as is)** on ecegrid with gcc v7.1.0 (64-bit Linux) and the following parameters: -g -std=c11 -Wall -Wshadow -Wvla -Werror -pedantic -DNDEBUG -Wno-unused-function.
- **Code can run** on ecegrid well enough to be tested.
- Code **finishes in a reasonable amount of time** (e.g., 2.0 seconds for most assignments)
- Function signatures and data types **match the specification** in the assignment description.
- Any main(…) function must **return EXIT_SUCCESS**.

⚠ **ZERO credit if you fail to meet any of the base requirements.**

aq.gs/264

⚠ Cheating

Cheating includes:

- Copying code from written by other people
  - other students
  - the web
  - the instructor (unless explicitly authorized in writing)
- Dishonest conduct
  - e.g., attempting to access exam contents, etc.
- Helping others to cheat
  - Posting your code publicly online (e.g., GitHub)
  - Sharing your code with others
- *Attempting* to do any of the above
- Doing any of this during or after the semester

⚠ Cheating

Penalties:

- Very minor (e.g., copying 1-3 lines from the web on a homework)

  0 on assignment

- Others:

  F in the course

All instances will be referred to the Office of Student Rights and Responsibilities

Full penalties will be applied for the first offense.

Penalties may be applied any time cheating is discovered.

/ ⚠

⚠ Read the syllabus carefully.

We will assume you have read the syllabus.

These slides are not comprehensive.