

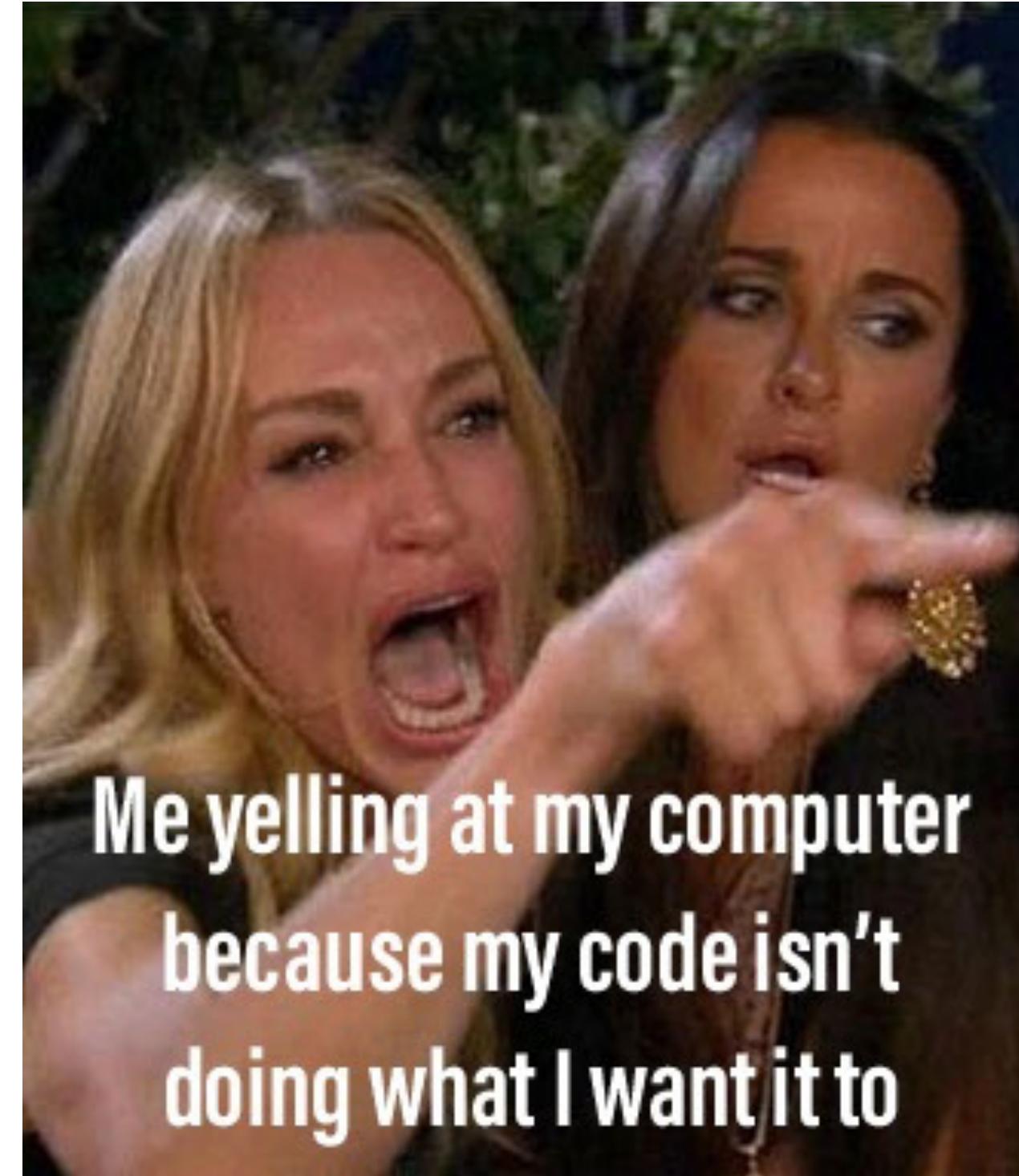
COMP1511 Week 1

Introduction

Joanna Lin

Self Introductions!

⭐ Meme of the Week ⭐



Me yelling at my computer
because my code isn't
doing what I want it to



My code doing exactly
what it was programmed
to do

Links + Where To Get Help

- Course website: <https://cgi.cse.unsw.edu.au/~cs1511/21T3/>
- If you're stuck,
 - Ask on the **forum**: <https://discourse.cse.unsw.edu.au/21t3/comp1511/>
 - Look out for **help sessions** on BBCollab (timetable to be released)
 - **Email** us (this includes if you need an extension!)
 - joanna.lin@student.unsw.edu.au
 - muhammad_waleed.shahid@student.unsw.edu.au
 - for more serious matters, we may direct you to the course email: cs1511@cse.unsw.edu.au
- If you're struggling to cope with stress, reach out to:
 - Student support: <https://student.unsw.edu.au/advisors>
 - CAPS: <https://student.unsw.edu.au/counselling>

Assessment Structure

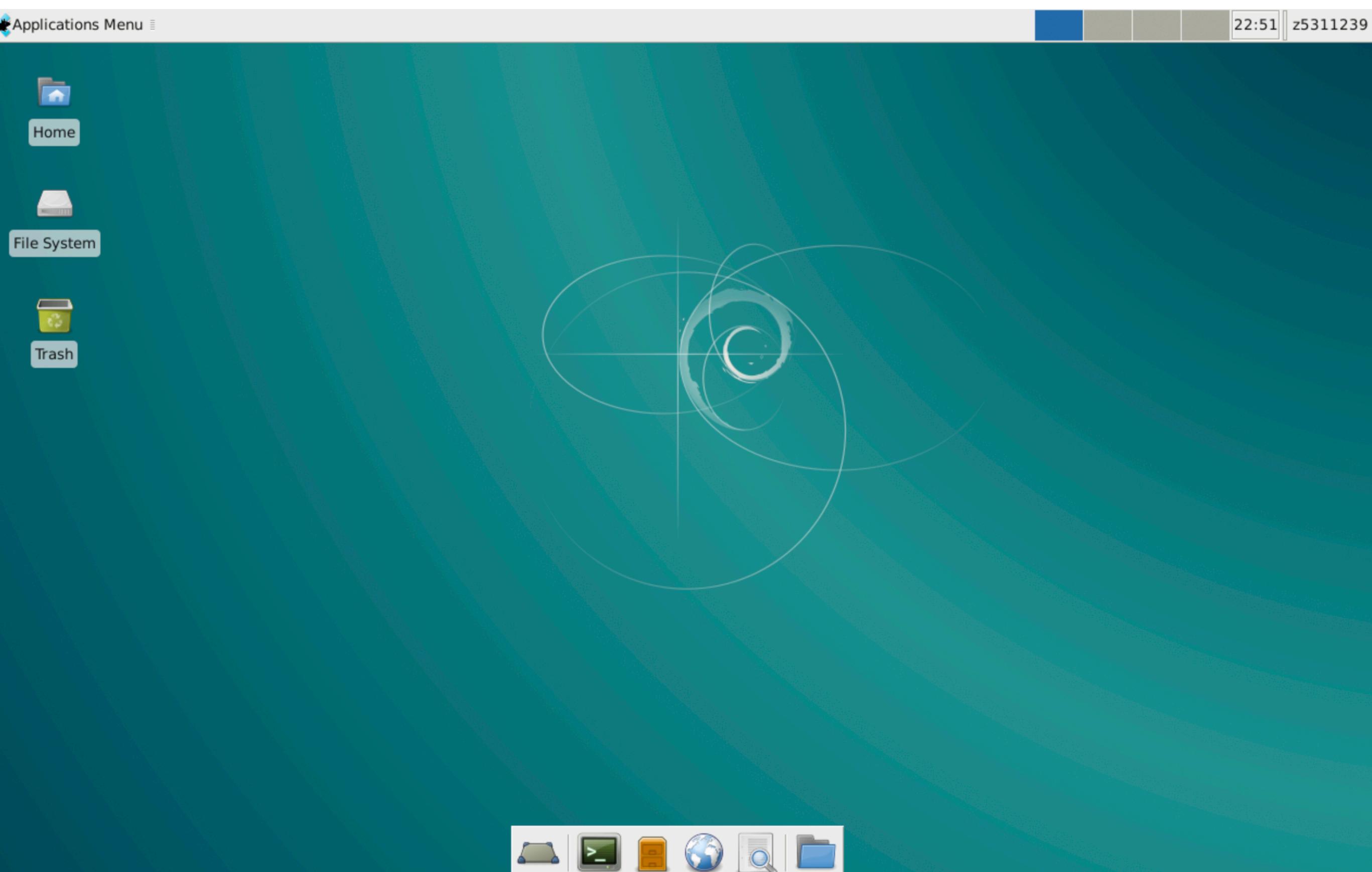
- **Lab Exercises (10%).** Coding problems to be done during weekly labs (after tutorial). Split into 1-dot, 2-dot, 3-dot and skull (based on difficulty). You can get 100% in this component completing only the 1 and 2-dot exercises, but 3-dot exercises can make up for lost marks. Skull exercises are not worth marks.
 - Labs in **weeks 2 – 5, 7 – 10** are assessed, week 1 has a lab but it's not assessable. They are due at **8pm on Monday** the week after release. Only the top **7 out of 8** labs count towards your final mark.
- **Weekly tests (10%).** Three coding questions to be done under self-enforced exam conditions in 1 hour. After the hour is up, you may continue working on the problems with no penalty as long as you submit before the due date.
 - Tests start from **week 3 – 5, 7 – 10**. They are due at **8pm on Thursday** the week after release (watch out for exact due date). Only the top **6 out of 7** are count towards your final mark.
- **Assignments (15% + 25%).** You will be writing larger programs than in labs according to provided specifications. Look out for announcements to see when they are released and when they are due.
- **Final exam (40%).** Will occur during exam block. Has 2 'hurdles' you need to pass in order to pass the course. Look out for more details closer to week 10.

What we'll cover today

- What is VLAD and how do we access it?
- What is a **terminal**?
 - How is it different from a graphical user interface?
- What is a **code editor**?
- What is code **compilation**?
- Writing our first C program(s).
 - What is **code style**?

VLAB

- CSE school's linux-based operating system (as opposed to Windows or MacOS).
- VNC viewers (such as TigerVNC) give you access to the VLAB on the school's computer from your own computer.
- You'll be using this to complete your lab exercises and weekly tests
- Instructions on how to log into VLAB:
https://cgi.cse.unsw.edu.au/~cs1511/21T3/home_computing/vlab.html



Terminal

- The terminal is our replacement for File Explorer on Windows and Finder on Mac. We use it to navigate through directories (folders), create files and access existing files.
 - File Explorer and Finder use a graphical user interface (GUI) which has user-friendly buttons for us to click on with our mice.
 - The terminal uses a command line interface (CLI). We interact the terminal mainly through our keyboard.

- Some common commands are:

ls list files and directories in current directory

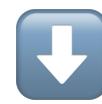
mkdir make a directory

cd change to another directory

mv rename or move a file

- Some useful abbreviations/actions are:

. current directory

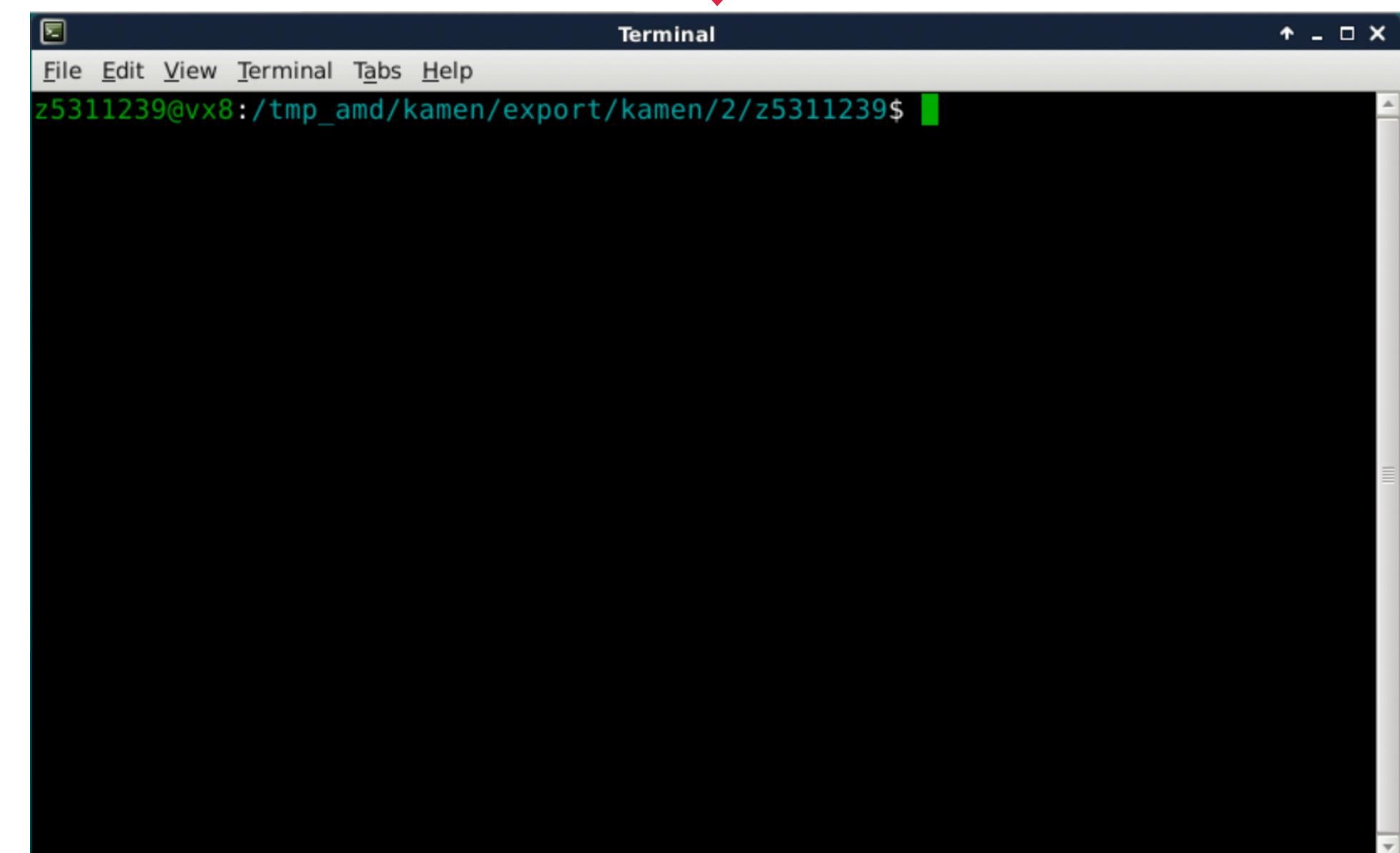
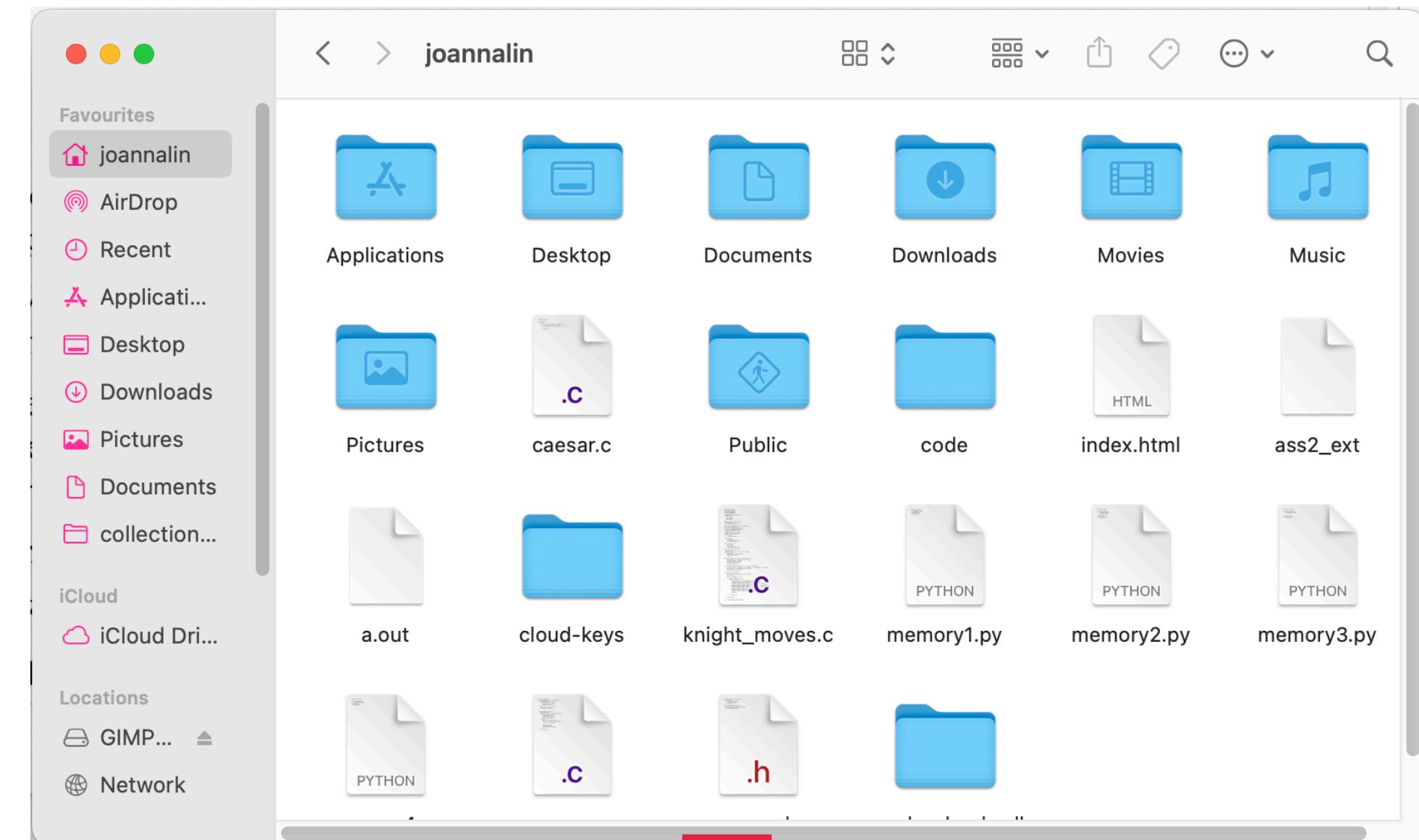
 next command

.. parent directory

 previous command

tab (x 2) list possible ways to complete current word (if any)

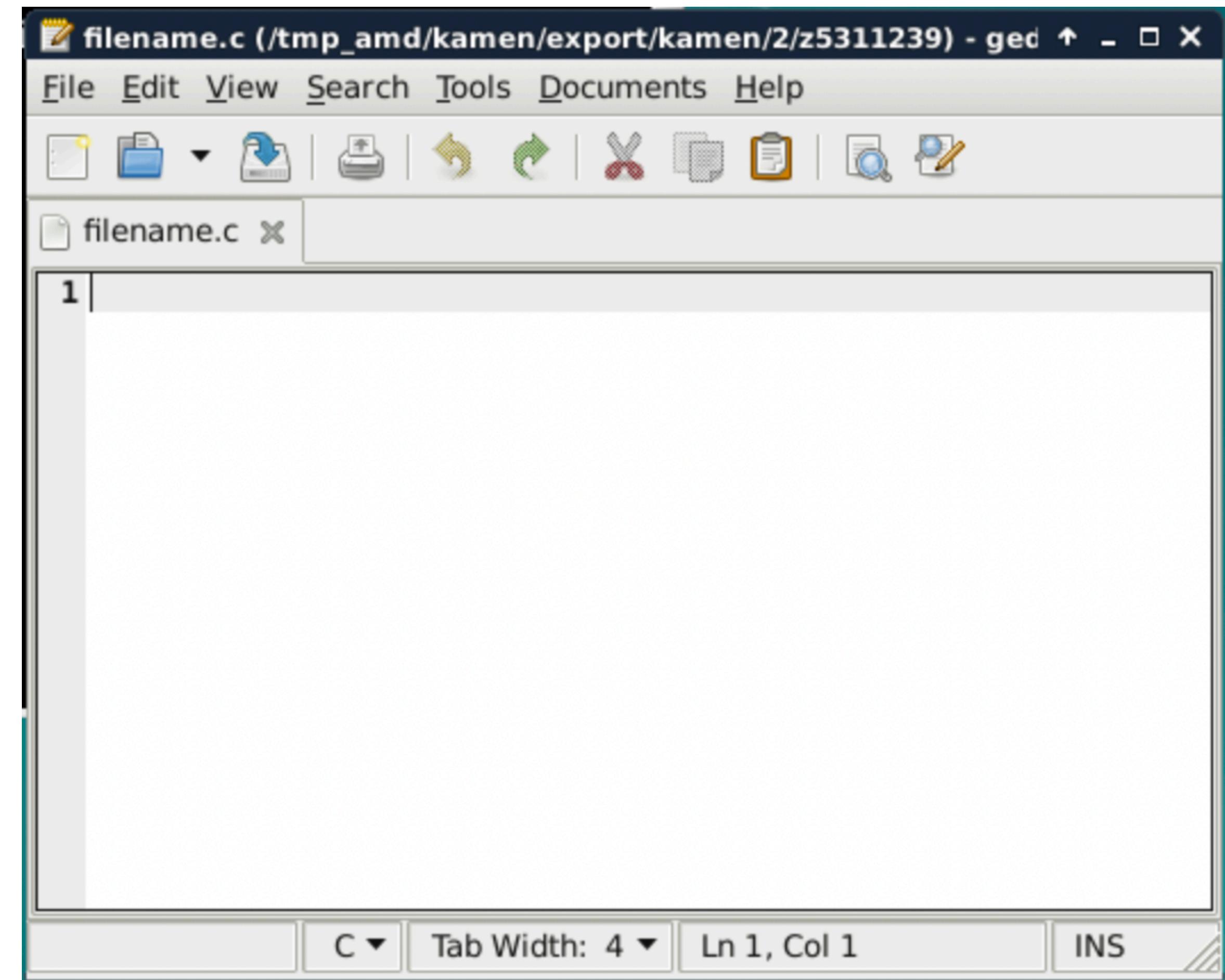
tab complete current word (if only 1 possible way)



Code Editor

Gedit

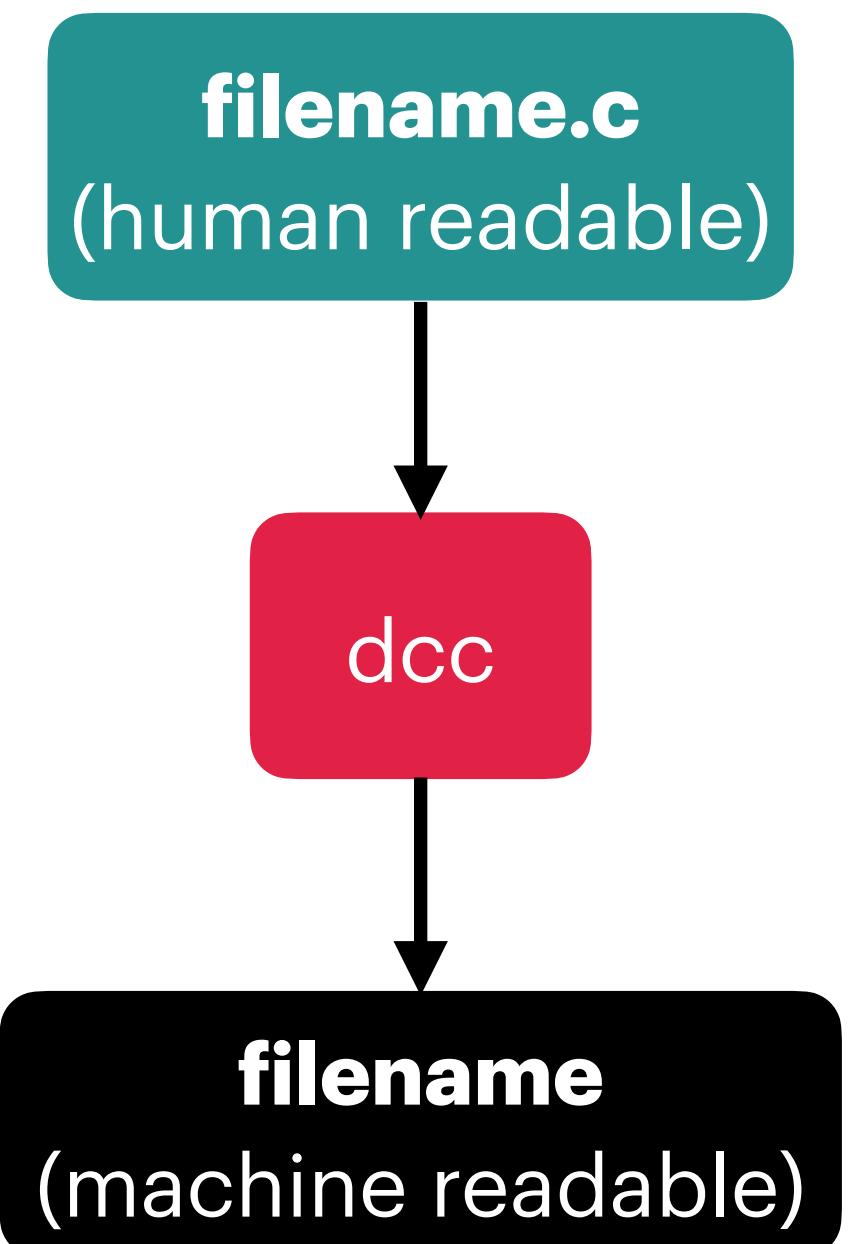
- We will use the C programming language in this course. Files containing C code have a `.c` extension.
- We write all of our C code into a code editor. We use gedit in this course.
- To edit or create a C file in gedit, we type `gedit filename.c&` into the terminal. The `&` allows us to continue using the terminal while we edit the file.
- Don't forget to save (`ctrl + s`) once you're done editing the file! An asterisk (*) next to the file name at the top indicates you have unsaved changes.



Code Compilation

What is it and why?

- The computer cannot read C code. It only understands binary.
- Compilation is the process of translating the C code we've written into machine code which the computer can read.
 - We compile code by running `gcc filename.c -o filename`
 - `gcc` is the compiler – the piece of software that does the translating
 - `filename.c` is the file where our C code is written
 - `filename` is the name of the output file – the file in which the machine code is written to. Note this does not have the `.c` extension.
 - Note that `filename` and `filename.c` are separate files. If we update `filename.c`, we'll need save and compile the code again to see changes reflected in the programs behaviour when we run it.
- We run the output program using the command `./filename`



Our First C Program

Breaking it Down

1. Header comment describing the program and stating the author and date. In general, comments (lines starting with `//`) are ignored by the compiler. They are used to explain code to humans. You can also use `/* */` for multi-line comments

2. `#include` gives our code functionality from other files. `stdio.h` was written by other programmers, allowing us to print to the terminal via `printf`

3. Computer will execute instructions line-by-line between the curly braces `{ }` after `int main(void)`.

The indentation between the curly brackets improves readability.

4. Everything inside the double quotes after `printf` and a set of brackets will be printed to the terminal.

7. `return` indicates the end of the program, where 0 indicates no errors occurred (by convention).

```
1 // Basic Hello World program
2 // Joanna Lin (z5311239), September 2021
3
4 #include <stdio.h>
5
6 int main(void) {
7     printf("Hello World\n");
8 }
9
10 }
```

5. The words 'Hello World' and a new line (represented by the escape sequence `\n`) will be printed to the terminal. The backslash (\) is known as the escape character

Question: How do we actually print \ to the terminal?

6. The semi-colon, indicates the end of one instruction. We must have one after every instruction.

First Steps

Before Starting Your Lab...

- Make sure you are on the course forum!
- Log into VLAB. Make sure you choose ‘**Use default config**’ on the pop-up when you first log in.
 - If you didn’t do that (you can tell if you don’t have anything at the top and bottom of your screen), right click the screen, click ‘**Open terminal here**’, type **1511 reset_panel** into the terminal, press enter and restart VLAB
- Open up the terminal.
- Run **1511 setup**
- Run **1511 colours**
 - close then reopen the terminal to get a better-looking terminal with colours!!!
 - You only need to run these commands **ONCE**.