

CSC209: Software Tools and Systems Programming

Karen Reid and Paul He

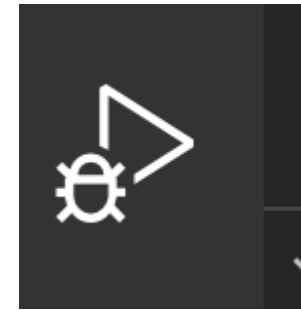
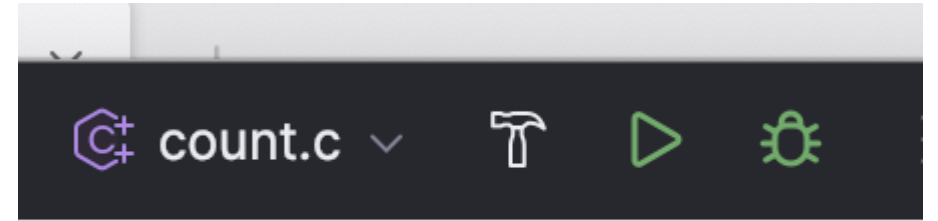
csc209-2025-01@cs.toronto.edu

Welcome to CSC209!

- First exercise: stand up and introduce yourself to at least 2 people you've never met
- Tell them your name
- Tell them one interesting thing you did recently
- Tell them what you think will be the most interesting or most scary/worrying thing about this course

Questions

- What does it mean to “run a program”?
- What is a program?
- What is a file?
- What is a running program?



```
$ ls
count.c  echo_arg.c  echo_stdin.c  hello.c
$ gcc -Wall -g -o hello hello.c
$ ./hello
Hello World
```

The Shell

```
$ gcc -Wall -g -o hello hello.c
```

- The \$ is a shell prompt
- Shells
 - Accept commands (programs) as input
 - Finds the executable
 - Passes the arguments
 - Starts executing the command
- Shells also have some “built-in” commands

Shell Demo

- Example of standard programs
- Key concepts:
 - Shell as an interpreter
 - Program arguments
 - Standard output and input
 - Redirection standard I/O to and from a file
 - Piping standard I/O to and from a process

Course Info

- Please read the syllabus!

Communication

- Piazza
 - Use for non-personal communication and technical questions
 - Try to make your subject lines informative
- Email
 - csc209-2025-01@cs.toronto.edu
 - Use for personal communication, like requests for special consideration
- Office hours
 - Use for questions about course material and help with coursework

Professionalism and Email

- Email is a formal method of communication
 - Use proper English
 - State your question clearly, with enough context
 - Sign it with your name (utorid or student number if you are asking about something logistical where it might help)

Flipped/Blended Classroom

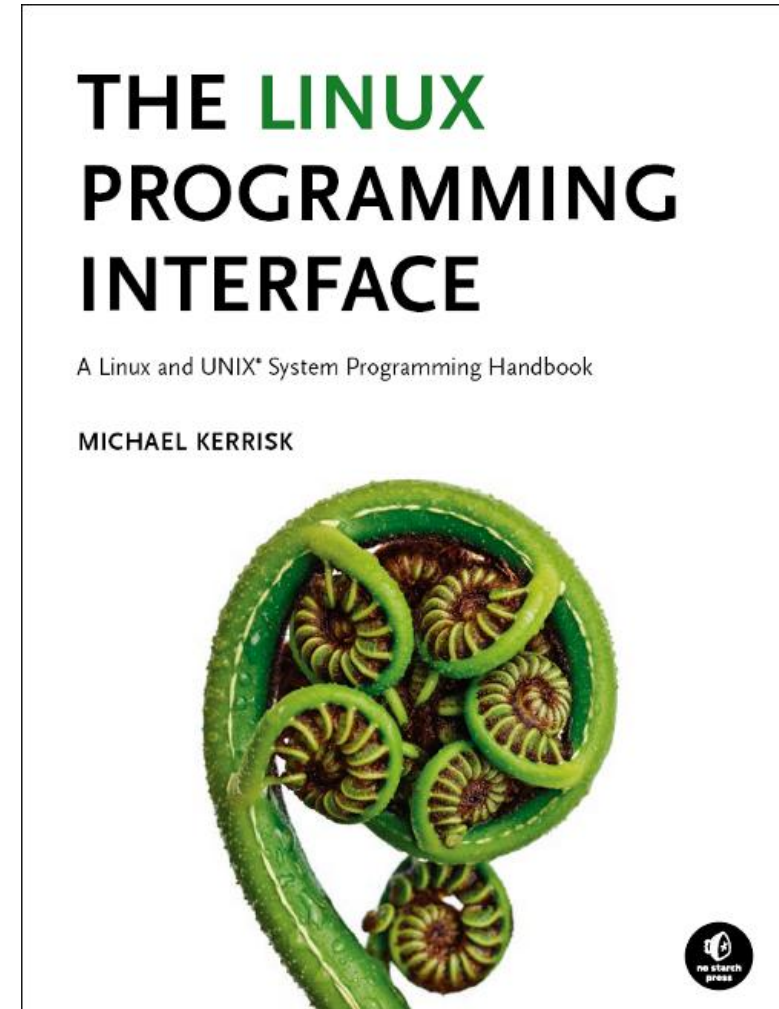
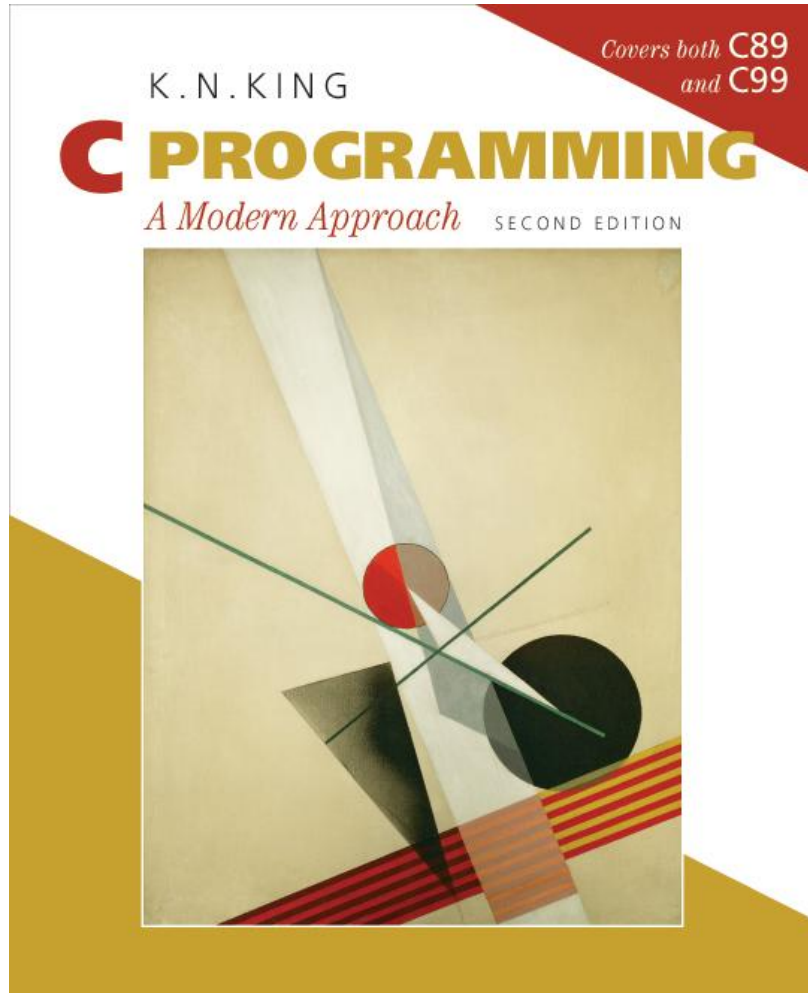
- Preparation before class (videos & exercises on PCRS)
- Hands-on activities in class
 - Need to be ready to do some programming during class




Course Information

- All important information will be on Quercus
- **Syllabus:** deadlines, policies
- **Home page** of Quercus: schedule, notes, worksheets, labs, etc.

Textbooks



Marking Scheme and Schedule

Work	Weight	Deadline
Lecture Preparation (PCRS)	5% (best 9 of 11)	Tuesdays before 9am (weeks 2 - 12)
Lab Exercises	7% (best 8 of 10)	Fridays before 10pm (weeks 1 - 11, one week will not have a lab)
A1	7%	Thu 30 Jan, before 10pm
Term Test 1	10%	Tue 4 Feb, during your lecture section
A2	10%	Thu 6 Mar, before 10pm
Term Test 2	10%	Tue 11 Mar, during your lecture section
A3	10%	Thu 3 Apr, before 10 pm
Research Surveys	1% (0.5% x 2 surveys)	Survey 1: link  , Survey 2: TBA
Final exam	40%	Minimum grade of 40% required to pass this course

Surveys

- First survey due Thursday 10am
- Research about the embedded ethics module that we will see next class

Assignments

- You will use git to manage and submit your assignments
- The git repositories are set up on MarkUs
- Submit your assignment to your repo well before the due date, and submit iteratively as you make progress
- All code must work on teach.cs to receive full marks
 - **Code that does not compile on teach.cs will get 0**
- See syllabus for late penalties and remark policies

Weekly Lab Exercises

- You are signed up for a tutorial section in a computer lab each week
- No in-person lab this week, but there is an exercise to do
- Submit your work on MarkUs using git
- One lab will be in-person only (date to be announced)
- Due Fridays by 10pm (no late submissions)

Software Installation

- Follow software setup instructions on Quercus
 - With help from friends or Piazza if needed
- Begin learning to use a good code editor (e.g. vim, VS code, emacs, nano, atom, sublime)
- Learn to connect remotely to teach.cs and compile, edit, and run programs:
 - Learning to use ssh is really important
 - <https://www.teach.cs.toronto.edu/using-labs/remote-access/>

Do your own work

- The work you submit must be your own, done without participation by others. It is an academic offence to hand in anything written by someone else without acknowledgement.
- You are hurting your friend when you **give** them a copy of your assignment
- You are hurting your friend when you **ask** them to give you a copy of their assignment

Academic Integrity

- Academic Integrity: Plagiarism and cheating
 - Very serious academic offences
 - All potential cases will be investigated fully
 - All assignments and exercises are to be completed individually
- Do not submit code for grading that is not your own
 - If you reuse any code, document the source
 - E.g., hash function from KN King book
- Do not look at others' code, and do not share your code
- Do not search for solutions on the web
- Use Generative AI, ChatGPT, CoPilot, etc., sparingly when learning the material (must cite if reusing, like other code!)
- Ask (and answer) questions on Piazza, but don't add details about your solution

Most importantly, be honest with
yourself

Unix Principles

- Do one basic thing well
 - with some basic variations
- Simple input formats
 - plain text
 - don't require interactive input
 - stdin to stdout/stderr
- Simple output format
 - expected to be input to another tool

Basic Tools to Learn

- `cd` - change directory
 - `mkdir` - make a new directory
 - `ls` - list files in a directory
 - `cp` - copy files
 - `mv` - move a file or files
 - `rm` - remove a file
 - `chmod` - change permission of file
 - `diff` - compare two files
 - `cut` - extract a column
 - `cat` - see entire file
 - `wc` - word count
 - `grep` - find text in file
- Get in the habit of checking the man page.
E.g. `man ls`**