

CS221  
C and Systems  
Programming



# What is C? What is Systems Programming?

C is the language that came after A and B

Systems programming is

- Programming to monitor, control, and use the lower-level parts of a computer system
- Provision of services/capabilities to software applications

**Networking**

**Memory management**

**Hardware management**

*Multiprocessing*

# What are the parts of a computer server?

Memory

"Persistent storage"

I/O Ports

Hardware devices (keyboard, mouse, monitor, camera, audio, ...)

Power and cooling

CPU

## Ok, what are the parts of a computer **system**?

### Network

- Network address manager ("DHCP") and address finder ("DNS")

### Distributed storage

- Distributed database

### Virtualization

- Task scheduler / load balancer
- Monitoring and management

COPYRIGHT 2024. PAUL HASKELL

4

Dynamic Host Config Protocol, Domain Name System

Virtualization: let users interact with a server without caring which HW server they are using

Monitoring & mgmt: centralized SW/FW updates, auto failover to backup, centralized configuration

# What do we cover in CS221?

Networking

Memory mangement

Operating system use

- CS326 covers how to write an operating system

## Administrative topics (1 of 3)

**Syllabus** online in Canvas. It is subject to change

**Office hours:**

- 11:30am – 12:30pm in person Mon/Wed, outside Harney 148
- 3:00 – 3:45pm Mon/Wed, outside Harney 148
- 1-2:30pm Thurs, via Zoom
- Or by appt
- I am an informal guy—just shoot me a note saying “Can we talk 5 minutes on Friday?”

**TA's:** Kabeer Thakur, Isaiah Walker

**Recommended Textbook:** *Head First C*, [oreilly.com](https://oreilly.com)

Course **communication tools:** email, Canvas, GitHub.

COPYRIGHT 2024. PAUL HASKELL

6

You will not do well unless you attend classes regularly.

Books: [learning.oreilly.com](https://learning.oreilly.com) LOTS of free books at oreilly

Show students Canvas modules, and README file!

## Administrative topics (2 of 3)

Course components:

- In-clas labs: 10% of final grade
- Take-home labs (homework): 10%
- Three big projects: 10% total
- Quizzes: 20%
- Midterm: 20%
- Final: 30%

I will **curve the course grade**

Grades for SW depend both on correctness of code and clean professional coding style

- I will give guidelines

Attendance basically mandatory—frequent in-class assignments

**Late-day policy** for late homeworks, projects. **No make-ups** for missed tests, quizzes, in-class labs

- Talk with me about exceptions i.e. with doctor's note

COPYRIGHT 2024. PAUL HASKELL

7

Curving: will not enforce Gaussian distribution. Will adjust grade thresholds so reasonable number of students get A's, B's, and C's. Will "publish the curve" approx. weekly after the first few assignments.

Quizzes each only about 1% of grade.

LATE homework/projects: tell policy! 8 "credit days".

## Administrative topics (3 of 3)

### Flipped Lectures...

#### **Honor Code** – zero tolerance for cheating

- Zero for assignment
- Every incident is reported to the University's Academic Integrity Committee
- Second incident causes automatic "F" for the course. Only appeal is to the Academic Integrity Committee

COPYRIGHT 2024. PAUL HASKELL

8

Scott and "C" in O-chem

Monica and "C-" in O-chem

If found guilty of cheating, your college transcript will say "Suspended for Honor Code violation" for the rest of your life

#1 issue among USF CS faculty. I think I am pretty successful at dissuading my students

- All exams and quizzes are paper and pencil
- I use an online system with every HW and proj that looks for similar code. Takes me 5 seconds. Pretty effective
- I give students a ton of help. Don't cheat—come to me for help! BUT DON'T START LATE!



# Automated Code-checking tool

```
total = total - 1.0/counter;
counter++; }
else{
    total = total + 1.0/counter;
}
}
System.out.println("Taylor(10)=" + total);
}
}
>>>> file: limits.java
class limits {
    public static void main(String[] args) {
        // to find maximum byte value
        long var1 = 0;

        long var2 = (byte)var1;
        while (var2 == var1) {
            byte byt1 = (byte)++var1;
            var2 = byt1;
            if (var1 != var2){
                System.out.println("Maximum byte value is " + (var1 - 1));
            }
        }

        // to find minimum byte value
        long varI = 0;

        long varII = (byte)varI;
        while (varII == varI) {
            byte bytI = (byte)--varI;
            varII = bytI;
            if (varI != varII){
                System.out.println("Minimum byte value is " + (varI + 1));
            }
        }
    }
}

}
// print the smallest possible saved float value
System.out.println("Smallest possible float is: " + var2) ;

// initialize double variables
double smallestDouble = 1.0 ;
double doubleValue = 0 ;

// locate the smallest possible double
while (smallestDouble != 0) {
    doubleValue = smallestDouble ;
    smallestDouble = smallestDouble / 2.0 ;
}

// print the smallest possible saved double value

System.out.println("Smallest possible double is: " + doubleValue) ;
}
>>>> file: limits.java
class limits {
    public static void main(String[] args) {
        // long to byte maximum value
        long var1 = 0 ;
        long var2 = (byte)var1 ;

        while (var1 == var2) {
            byte maxByte = (byte) ++var1 ;
            var2 = maxByte ;
            if (var2 != var1) {
                System.out.println("Maximum byte value is " +(var1 - 1)) ;
            }
        }

        // long to byte minimum value

        long minByte = 0 ;
        long compareMinByte = (byte)minByte ;

        // find minimum value for byte
        while (minByte == compareMinByte) {
            byte minByte2 = (byte) --minByte ;
            compareMinByte = minByte2 ;
            if (compareMinByte != minByte) {

```

I run this automatically as part of auto grading. I look at results

## Tools for the class



# Text editor

## Graphical editors:

- Notepad++, Sublime, etc
- VisualStudio is more than a text editor...

## Command line editors:

- nano
- vim

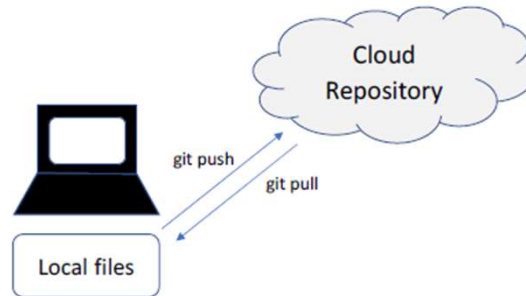
IDE's: we will start using in ~4 weeks. I ask everyone to stick with the command line until then

VisualStudio: everyone can use whatever IDE you want. But if you don't use my choice, the TA's and I will not help you with it.

# Git, GitHub, and Git Bash

REVIEW: **git** is a collection of SW tools to save and retrieve versions of software from a **repository**

REVIEW: **GitHub** is a repository-hosting website



COPYRIGHT 2024. PAUL HASKELL

12

Two repositories: shared "CourseInfo" and individual repos for each student

## Git use

You can look at files on GitHub via browser...

...or can "pull" to your computer and look there

When using **git** in a terminal window, you must check error messages

Good to double-check via browser also

COPYRIGHT 2024. PAUL HASKELL

13

I will happily help you fix problems with git setup (talk about merge probs).  
But if work is not COMMITTED on time, it is late. Git problems are not an excuse.  
You always can COMMIT even if you can't PUSH.

# Git Bash

A **terminal program** ("shell") available on Mac and Windows

Everyone is required to use Git Bash **development environment** in CS221

Windows vs MacOS vs Linux...

Terminal commands...

"Home directory"

COPYRIGHT 2024. PAUL HASKELL

14

"Bash" = "Bourne-again shell"

"Bourne shell" = one of the first shell programs

Even Microsoft uses Linux in their enterprise environments

# Git Bash

## Basic commands:

- ls, pwd, cd, rm, mv, cp, mkdir, rmdir, echo

## Tips and tricks:

- Filename substitution with '\*'
- Command substitution with '!'

## Shell variables:

- Set
- PATH variable

# Git Bash

Shell configuration:

- .profile

Shell **scripts...**

COPYRIGHT 2024. PAUL HASKELL

16

Write some shell scripts, with "if" and "while"  
See 'basicScript'

Let's look at a script that removes a file by moving it to a wastebasket



Lab01

