

Welcome to CS 150

CS 150 – C++ Programming I
In-Person Lecture 1



Welcome to CS 150 - Canvas

- Find log-in link on OCC home page
- Syllabus & Schedule are first links
 - Read it when you need details
 - Contact information in syllabus
- How you'll be graded
 - 50% programming exams (10 @ 45 minutes each)
 - Three "written" midterms (15%) and 1 final (20%)
 - Homework, Reading and Lecture-Lab Exercises (20%)
 - Note 105%-"extra credit buffer" for emergencies

[Home](#) | [1](#) | [2](#) | [3](#) | [4](#) | [5](#) | [6](#) | [7](#) | [8](#)

CS 150 Summer 2023

Course Resources

Course [Syllabus](#) and [Schedule](#)
Office Hours: By Appointment

Course Tools & Links

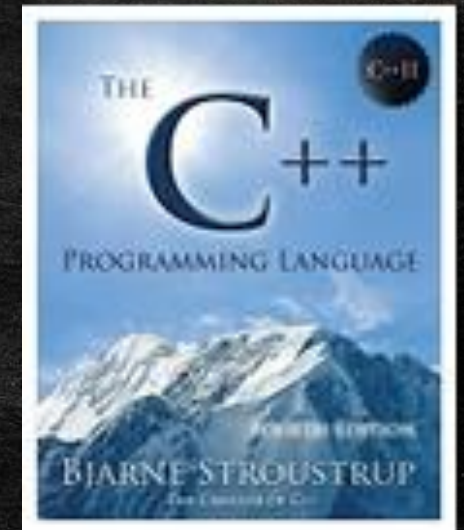
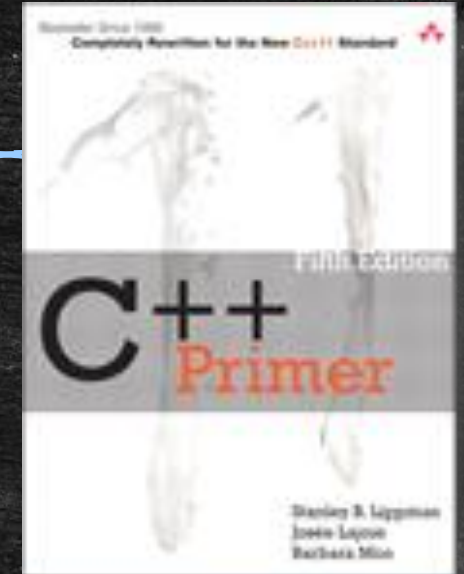
[CS 150 Homework Console](#)
[Course Reader](#), [Replit](#), [GitHub](#)
[C++ Shell](#), [GDB](#), [G++](#), [Clang++](#)

Practice & Reference

[Code Step by Step](#) (practice)
[Open Kattis](#) (challenge problems)
[C++ Reference](#) & [CPlusPlus](#) (ref)
[Fundamentals of C++](#) (alternate text)
[A Tour of C++](#) (1-5 supplemental text)
[C++ Core Guidelines](#) (style)

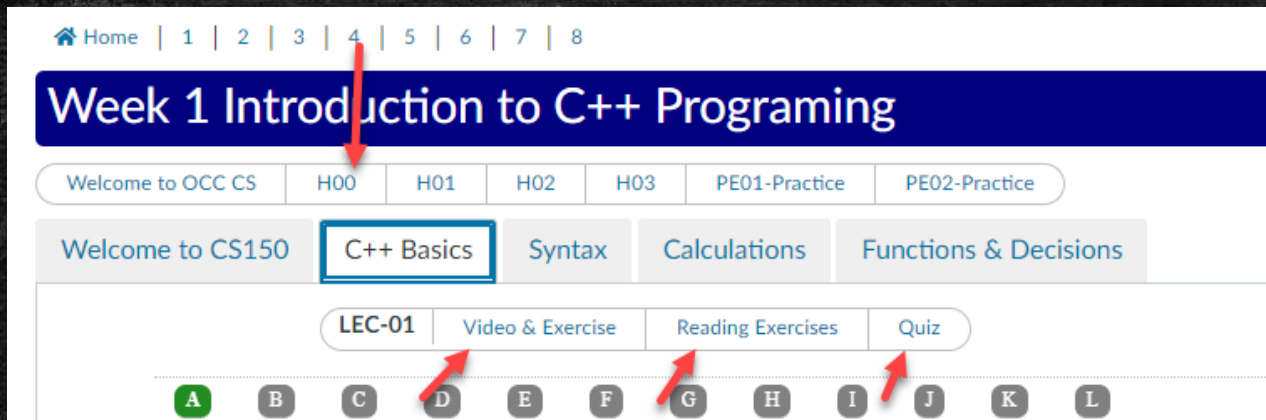
Reading

- Required CS 150 Course Reader (free, online)
 - Additional handouts also on home page
- Reference textbooks:
 - C++ Primer (Lippman)
 - A book you'll want to keep in your library after the course is over
 - The C++ Programming Language 4th Edition (Stroustrup)
 - The "official" reference guide
- Course will follow the Course Reader



Your Responsibility - Active Learning

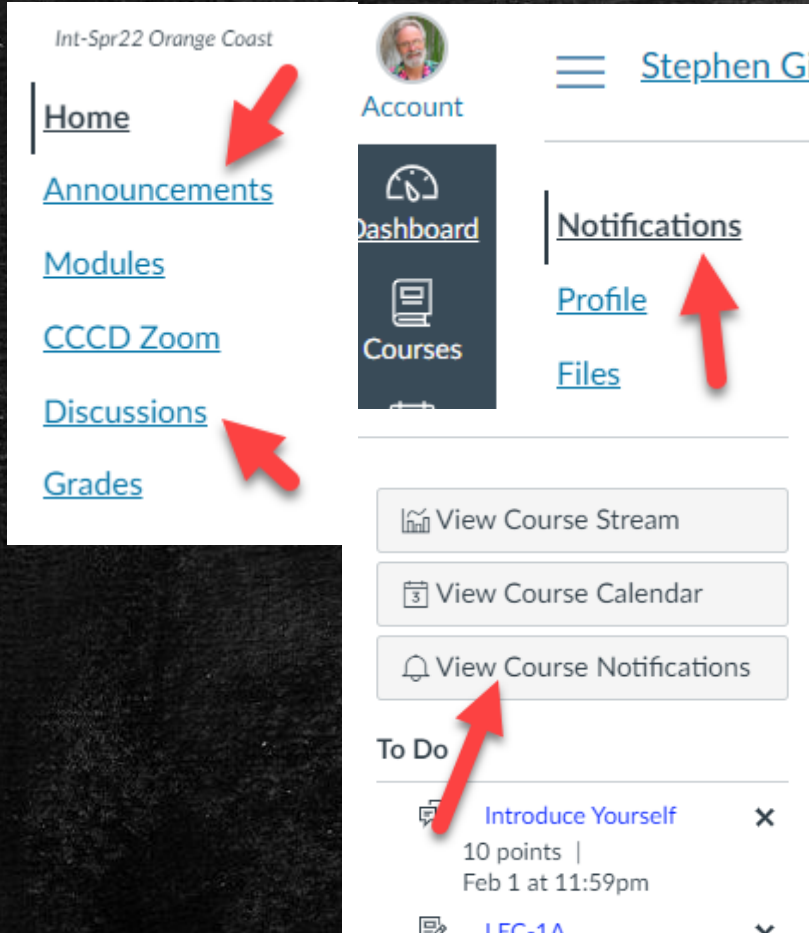
- Watch the lectures, read the book, do the reading exercises, take the quizzes and do the homework **before** the next class starts



- No couch potatoes!
- Can't learn by passively listening
- Learn by doing



Announcements, Questions & Email



- I will post announcements on Canvas
 - Make sure you get notified
- Ask questions *right away* when stuck
 - Use the Canvas discussion group
 - Answer your peers' questions.
 - Teaching is the fastest way to learn
- Private or confidential questions—use email or office hours (or Zoom)
 - Canvas Messaging works well for this

Lecture-Lab Exercises

- Each lecture includes a video and a set of in-class exercises
 - Log into your (student) Gmail account & make a copy

The image shows a screenshot of the CS150 website interface. At the top, there is a navigation bar with links: Home, 1, 2, 3, 4, 5, 6, 7, 8. Below this is a blue banner that reads "Week 1 Introduction to C++ Programming". Under the banner, there are several tabs: "Welcome to CS150", "C++ Basics", "Syntax", "Calculations", and "Functions & Decision". Below these tabs, there is a row of buttons: "MONDAY", "Welcome to CS", "Introduce Yourself", "In-Class Exercise", and "Configu". The "In-Class Exercise" button is highlighted, and a mouse cursor is pointing at it. Below this row, there is a button labeled "LEC-00".

On the right side of the screenshot, there is a section titled "LEC 00 Lecture-Lab Exercise". It contains the following text: "Click the [link for the starter form](#). Save a copy of the form in **your own** G below, completing the exercises and shooting the requested screensh exercises, upload it here to get credit for this assignment." Below this text is a button labeled "Finish Up".

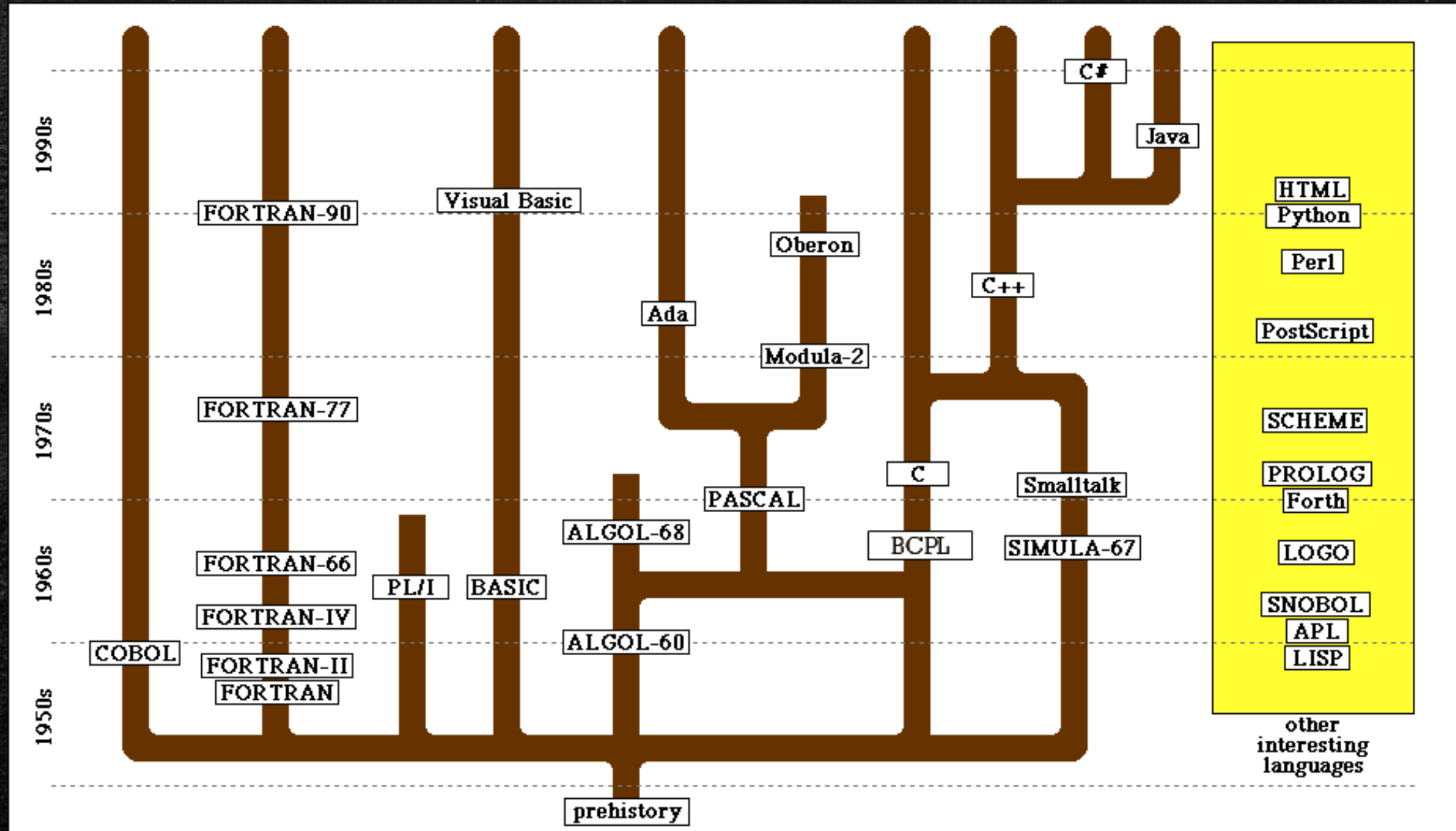
In the foreground, there is a Google Docs "Copy document" dialog. It shows the Google Docs logo and the text "Copy document". Below this, it asks "Would you like to make a copy of LEC-00?". There is a button labeled "Make a copy" with a mouse cursor pointing at it.

At the bottom left of the image, the text "CS 150 Lecture 00" is visible. At the bottom right, the date "1-Jun-23" is visible.

Shooting Screen Shots

- Each exercise is going to ask you to paste one or more **screenshots** into your Lecture-Lab document.
- In Windows 10
 - Hold down the **Windows + Shift + S** keys
 - **Drag** the cross-hairs through the portion you want to capture
 - **Paste** the picture into your document
- On the Mac use the keys **Cmd + Shift + 4**
- If you like you can use some other screen-capture software
 - However, just capture the portion of interest, **NOT the whole screen**

An Overview of Computer Languages




What is UNIX?



- Ken Thompson & Dennis Ritchie create UNIX operating system (and C programming language) at AT&T Bell Labs about 1970

Important UNIX Personalities

- **Bill Joy** – headed the Berkeley version of UNIX as a graduate student. One of the founders of Sun and developers of Java



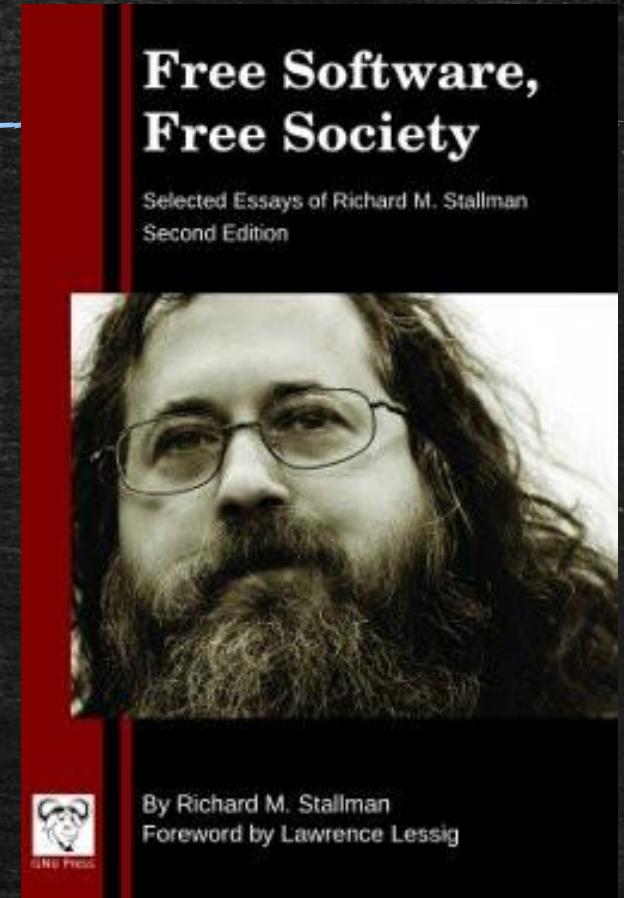
BILL JOY
American, 1954

"My method is to look at something that seems like a good idea and assume it's true"

Sun Microsystems
NFS Java SPARC
BSD csh Solaris vi
Berkeley UNIX +TCP/IP

Important UNIX Personalities

- **Richard Stallman** – M.I.T. research assistant in the AI Lab during the "Lisp" bubble of the late '80s. Founded the **Free Software Foundation** and is responsible for **GNU**, a free version of the standard UNIX utilities.
- The biggest of these is **GCC** – the GNU C Compiler Collection.
- Free Software and **GPL** license
 - Requires users to provide source code so users can modify code.

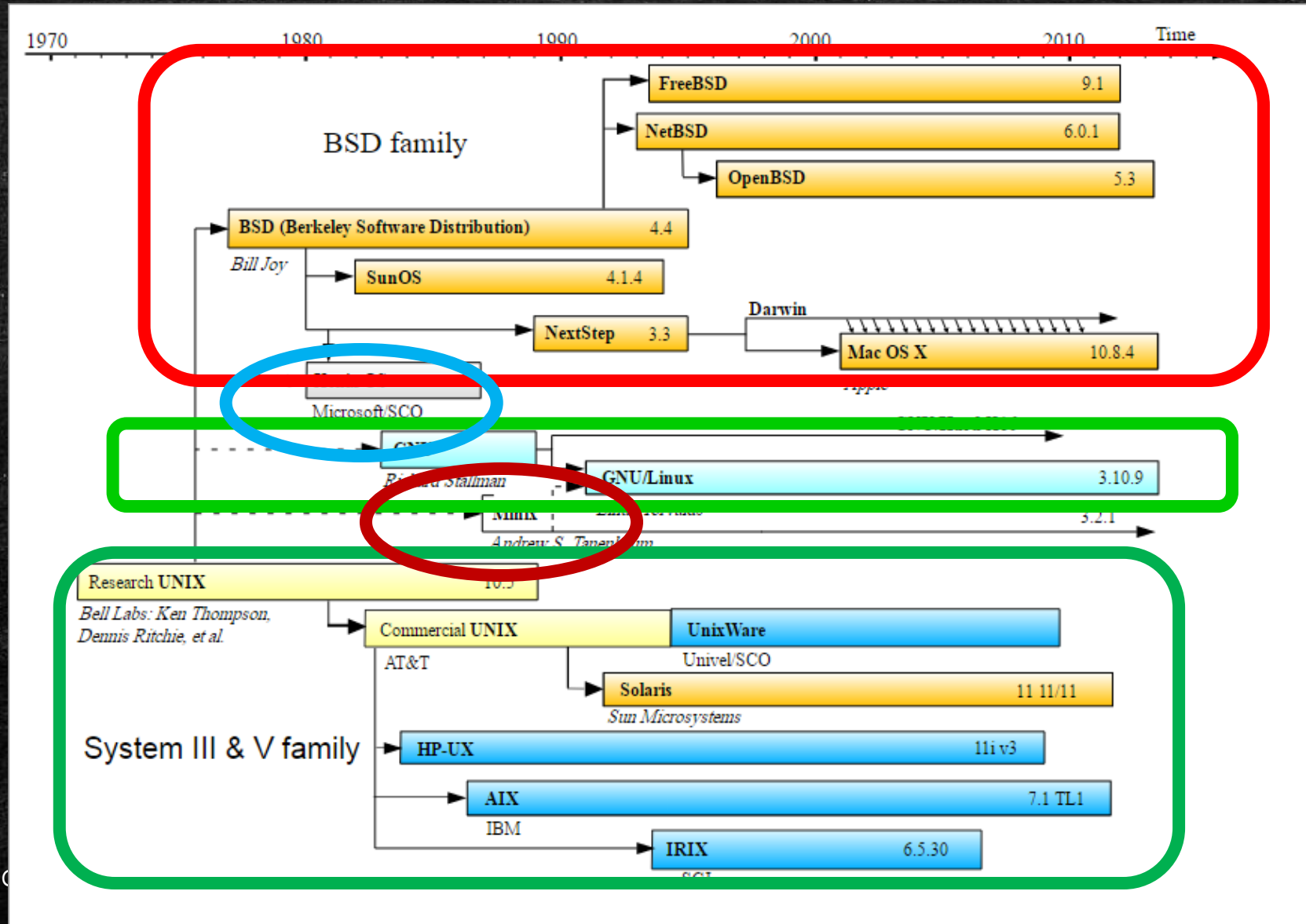


Important UNIX Personalities

- **Linus Torvalds** – Finnish graduate student who wanted to build a version of UNIX to run on i386 micro computers
- Advertised on USNET for collaborators and released first version of Kernel in 1994.
- Released under GPL
- Also developed GIT
- Currently works for Linux Foundation in Oregon

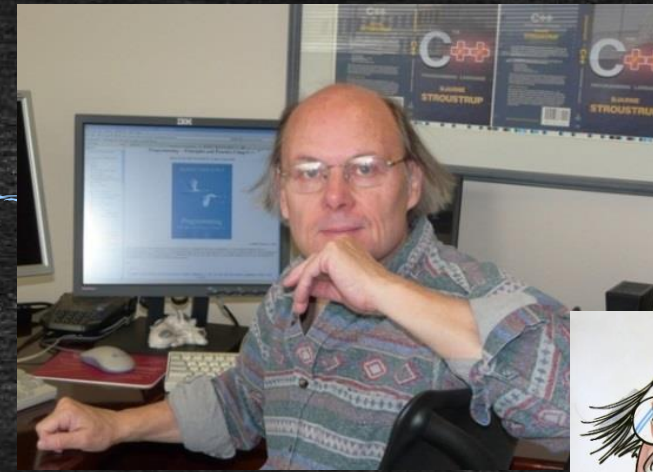


Versions of Unix

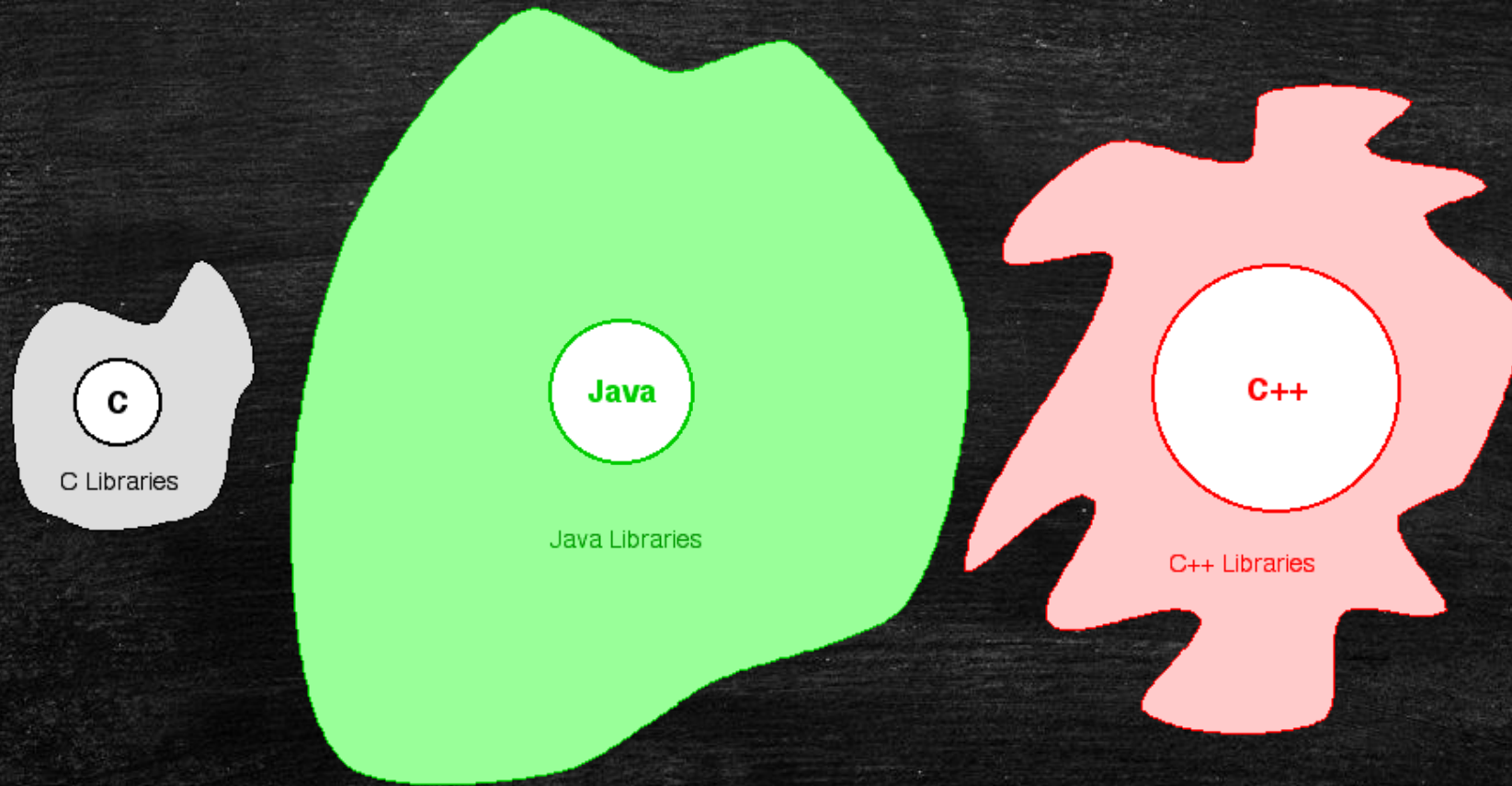


What about C++

- **Bjarne Stroustrup**
 - At Bell Labs in 1980
 - Now at Morgan Stanley & Columbia
- General-purpose, pragmatic rather than idealistic
- Support for different programming techniques:
 - **Procedural**: better C (type-safe, zero overhead)
 - **Data abstraction** (Object-based or Value-Oriented)
 - **Object-Oriented** programming (polymorphism)
 - **Generic** programming (Templates, parameterization)
 - **Functional** programming (Lambdas & functors)
- Designed as a language for professionals



Language Size and Complexity

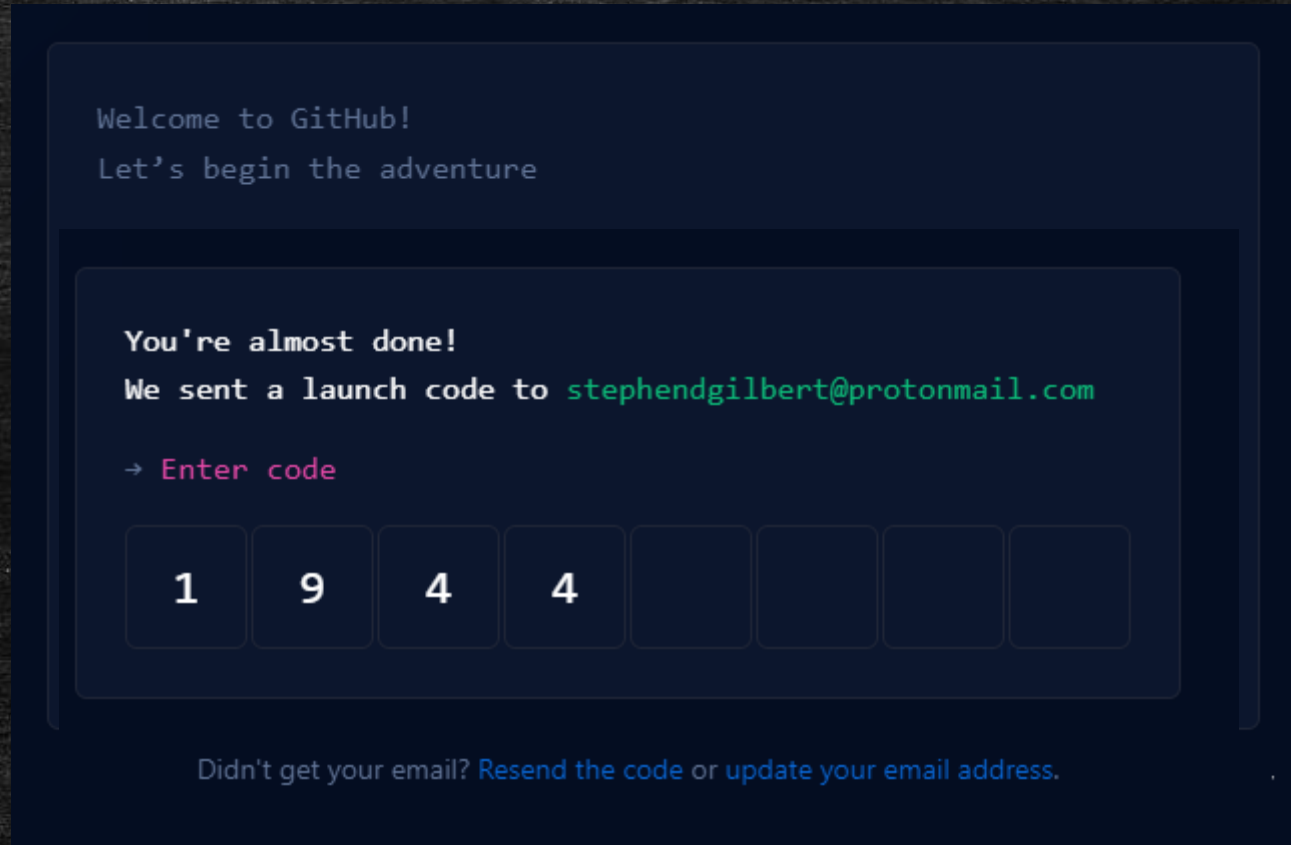
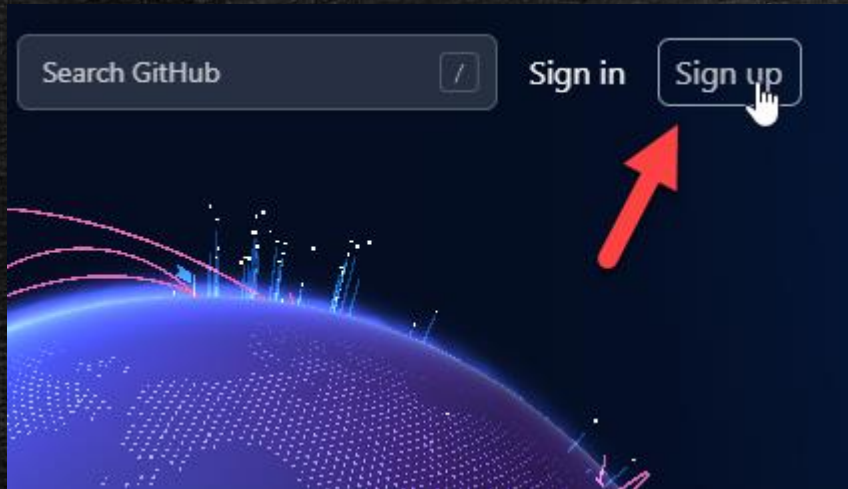


Configuring your CS 150 IDE

- Before programming, you need to **set up your tools**:
 - 1. Create a GitHub account (if you don't have one)
 - 2. Join the CS 150 Classroom and create your repo
 - 3. Got to your repo and create your CodeSpace
 - 4. Run 150config.py to configure your IDE
 - 5. Log into the CS 150 Homework Console
- Instructions are found in the Lecture-Lab document
 - Follow along with your instructor
 - Shoot screenshots as you go

Step 1 – Create a GitHub Account

- Where you'll store your code as you work on it



Step 2 – Join the CS 150 Classroom

- Click the link in the instru

Step 2 - Connect to the CS 150 Classroom

When you submit your homework assignment in CS 150, they'll be s
repository that only you and T

- Click this link to [join the](#)
- Join the CS 150 Classroom
list then **uou mau skin**

Join the classroom:

CS150 - Summer '23

To join the GitHub Classroom for
from the list below to associate y
school's identifier (i.e., your nam

Identifiers

kgilbert@student.cccd.edu

kmivamoto5@student.cccd.edu

CS150 - Summer '23

Accept the assignment —
CS 150 Starter Code

Once you accept this assignment, you will be granted access to the
itory in the [occ-cs150](#) organization

[occ-cs150 / cs150-su23-steveandkathyg](#) Private

Unwatch 1

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

main

1 branch

0 tags

Go to file

Add file

<> Code



github-classroom[bot] Initial commit

ee96a24 1 minute ago

1 commit



.vscode

Initial commit

1 minute ago



bin

Initial commit

1 minute ago



cs150

Initial commit

1 minute ago



.devcontainer.json

Initial commit

1 minute ago



150config.py

Initial commit

1 minute ago



Dockerfile

Initial commit

1 minute ago



README.md

Initial commit

1 minute ago

About

cs15

GitH

F

C

1

CS150 Starter Code.

been created:

Rele

No rel

Create

this assignment (update).

Step 3-Create Your Codespace

The screenshot shows the GitHub repository page for 'cs150-f22-stev' with the 'Code' dropdown menu open, showing 'Local' and 'Codespaces' options. A red arrow points to the 'Code' button. Another red arrow points to the 'Create codespace on master' button. A third red arrow points to the 'Update Now' button in the 'Updates Available' dialog. A fourth red arrow points to the 'Hide logs' link in the terminal output. The terminal output shows the process of finding an image, building a container, and starting configuration. The 'Open this codespace in VS Code Desktop' button is also visible.

Setting up your codespace

Go to file Add file <> Code About

Local Codespaces

Welcome to cloud editing

Edit, debug, and run your repository without local cloning and setup

cs150-f22-stev
GitHub Classroom

Readme
0 stars

Updates Available
Source: CS50 (Extension)
Update Now Remind Me Later

Create codespace on master

Codespace usage for this repository is paid for by occ-cs150
Need even more power? [Contact our team](#) to enable 32-core or GPU machines.

No releases published
[Create a new release](#)

Packages
No packages published
[Publish your first package](#)

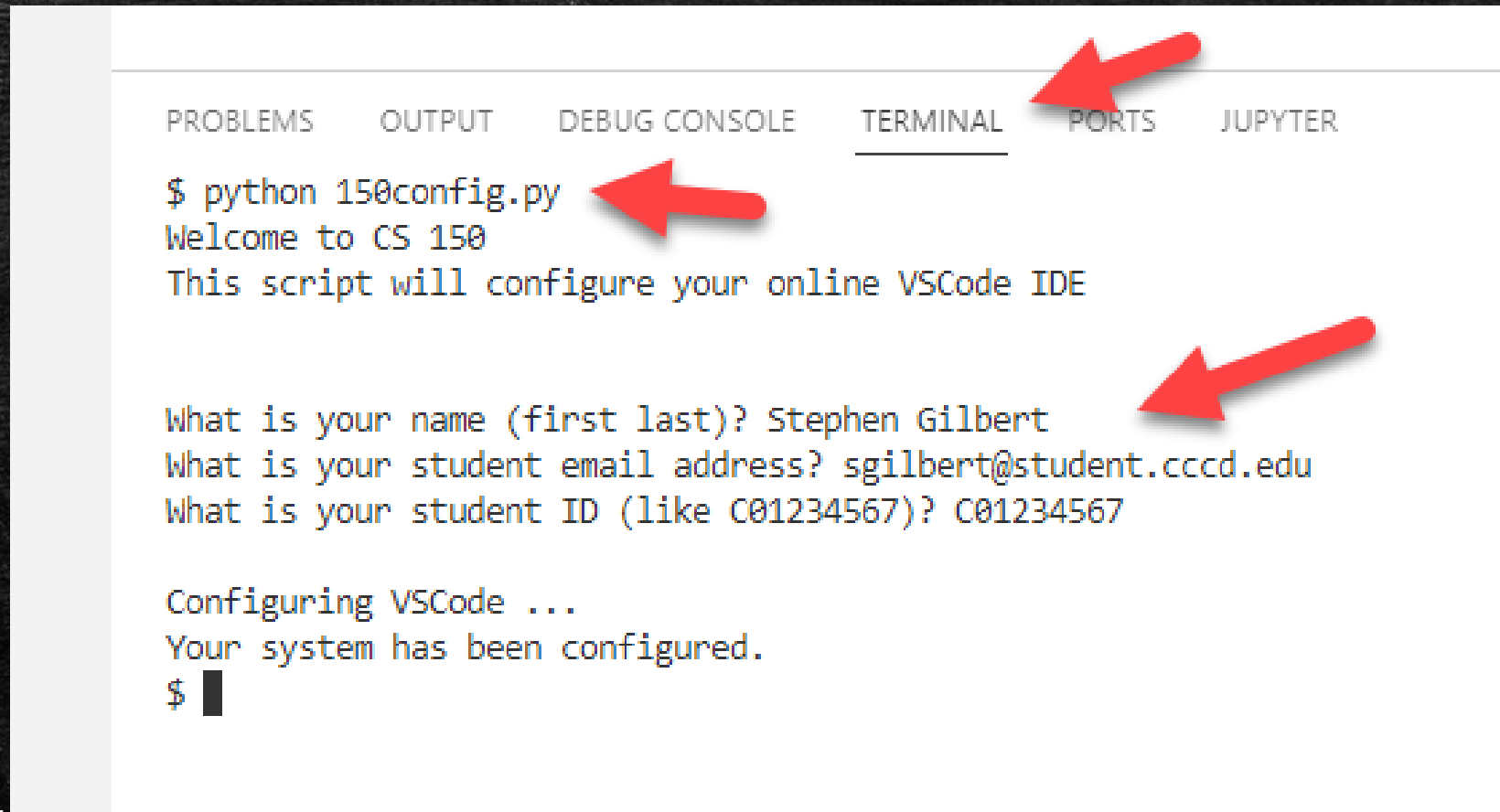
Open this codespace in VS Code Desktop

Tip See your application running with port forwarding. [Learn more](#)

```
✓ Image found.  
Building container...  
Hide logs  
Configuration starting...  
git process exited with exit code 0  
Creating container...  
[13 ms] @microsoft/vscode-dev-containers-cli 0.63.0.  
[148 ms] @microsoft/vscode-dev-containers-cli 0.63.0.  
codespacemount/workspace" clone --branch master --depth 1  
cs150-f22-steveandkathyg "/var/lib/docker/codespacemoun  
kathyg"  
codespacemount/workspace/cs150-f22-steveandkathyg'...  
ode 0  
codespacemount/workspace/cs150-f22-steveandkathyg" config  
fs/heads/*:refs/remotes/origin/*
```


Configure Your IDE

- In the terminal type `python 150config.py`
 - **Repeat** this every time you rebuild your code space



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  JUPYTER

$ python 150config.py
Welcome to CS 150
This script will configure your online VSCode IDE

What is your name (first last)? Stephen Gilbert
What is your student email address? sgilbert@student.cccd.edu
What is your student ID (like C01234567)? C01234567

Configuring VSCode ...
Your system has been configured.
$
```


Step 4 – The CS 150 Homework Console

- Go to <https://cs170-console.appspot.com>
 - Login ID:** your email name w/o student.cccd.edu
 - Password:** your student ID **including** the capital C

CS 150 Homework Console

Please Log In

Login ID:

Your login ID is your OCC email name, with

Password:

Stephen Gilbert

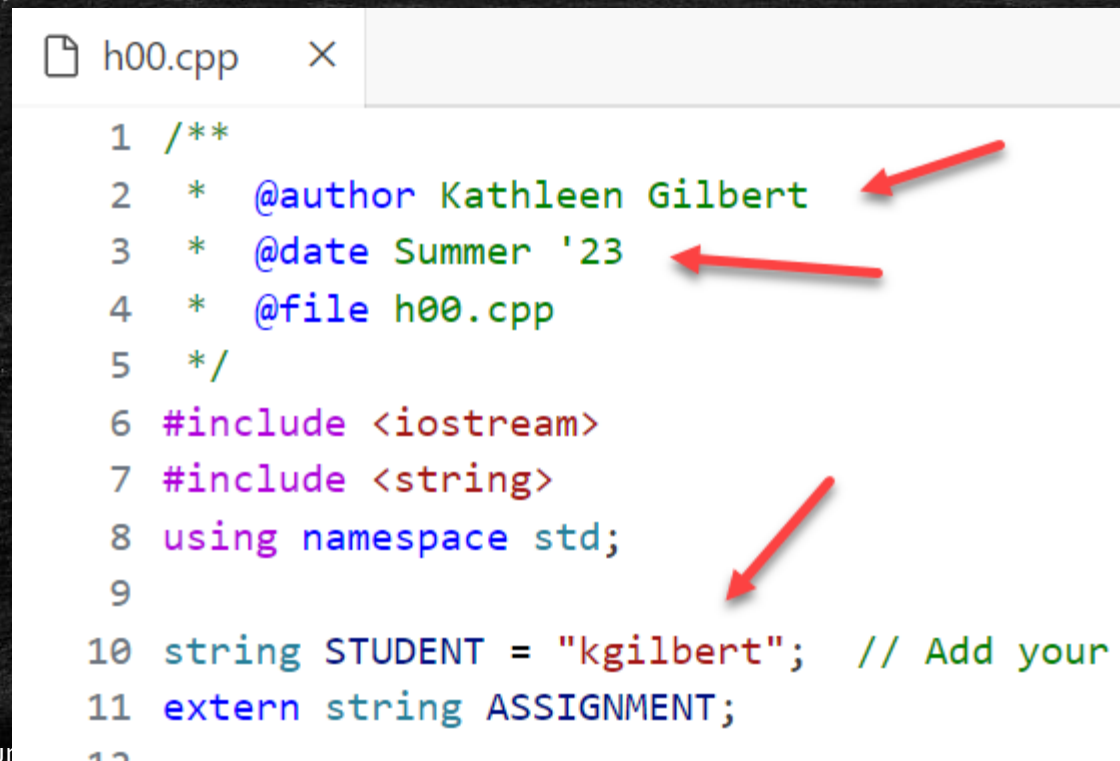
Welcome to 150!

Homework Grade: 0%

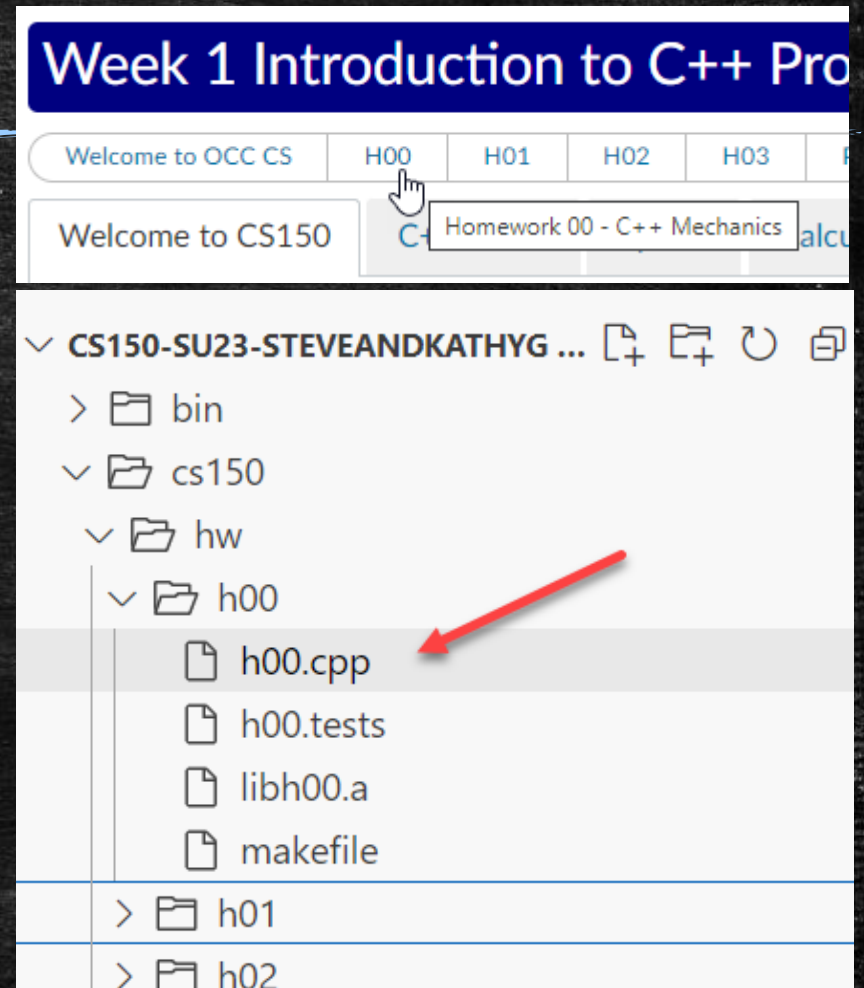
Assignment	Deadline	Score
H00	Feb 07, 09:35 AM	0
H01	Feb 14, 09:35 AM	4

Homework 00 - C++ Mechanics

- Open the instructions on Canvas
- Open your IDE and open `h00.cpp`
- Add your name, section and id



```
1 /**
2  * @author Kathleen Gilbert
3  * @date Summer '23
4  * @file h00.cpp
5  */
6 #include <iostream>
7 #include <string>
8 using namespace std;
9
10 string STUDENT = "kgilbert"; // Add your
11 extern string ASSIGNMENT;
```



Designing a Solution

- Here is the program we are going to solve
 - *A metric ton is 35,273.92 ounces. Write a program that will read the weight of a package of breakfast cereal in ounces and output the weight in metric tons as well as the number of boxes needed to yield one metric ton of cereal.*
- **Design your program** before you start coding
 - "The sooner you start coding, the longer it will take"
 - What are **inputs**, **outputs**, **algorithms** and **assumptions**?
 - Add these as **internal comments** in your code

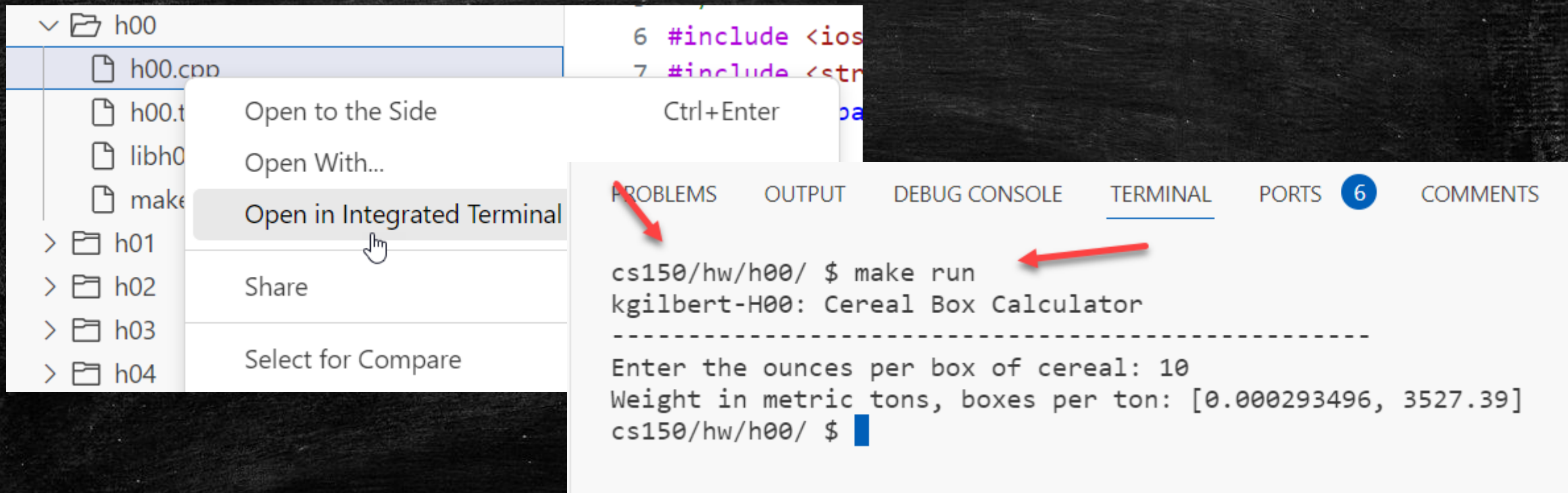
Planning the Interaction

- **Mocking up the interaction** is determining what the program will look like when it runs
 - Use the **cout** object and the **<<** operator to display output

```
25
26 cout << STUDENT << "-" << ASSIGNMENT << ": "; 1
27 cout << "Cereal Box Calculator" << endl;
28 cout << string(50, '-') << endl; 2
29 // Input
30 cout << "Enter ounces per box of cereal: " << 10 << endl; 3
31 cout << "Weight in metric tons, boxes per ton: ["
32     << 0.000283496 << ", " << 3527.39 << "]" << endl; 4
33
```


Compile, Link & Run

- First, switch the current working directory
 - Right click `h00.cpp` and choose "Open in integrated terminal"
 - Type `make run` at the shell prompt, and press `ENTER`



Reading Input

- Follow these steps to read the input:
 - Remove the "mockup" data at the end of the prompt
 - Create a variable to hold the input
 - Use the `cin` (see-in) object with the `>>` operator to read

```
29 cout << string(50, ' ') << endl;
30
31 // Input
32 cout << "Enter ounces per box of cereal: "; // prompt
33 double ouncesPerBox;                        // store the input
34 cin >> ouncesPerBox;                        // read the input
35
36 // Processing section
```


Processing and Output

- Create and initialize variables for the output
 - Use variables in place of the literals in the output section

```
31 // Input
32 cout << "Enter ounces per box of cereal: "; // prompt
33 double ouncesPerBox; // store the input
34 cin >> ouncesPerBox; // read the input
35
36 // Processing section
37 double weightInTons = ouncesPerBox / 35273.92;
38 double boxesPerTon = 1.0 / weightInTons;
39
40 // Output section
41 cout << "Weight in metric tons, boxes per ton: ["
42     << weightInTons << ", " << boxesPerTon << "]" << endl;
43
44 return 0;
45 }
```


Constants, Testing and Submitting

- For the given or assumed values, use constants

```
// Processing section
const double OUNCES_PER_TON = 35273.92;
double weightInTons = ouncesPerBox / OUNCES_PER_TON;
double boxesPerTon = 1.0 / weightInTons;
```

- To run the instructor tests, type `make test` and then `ENTER`
- After tests pass, type `make submit` and then `ENTER`
- Check the Homework Console that your grade is updated

Coming Up Next

- **Week 1** – Deadlines are 1pm on Monday or Tuesday
 - LEC-01—C++ Mechanics & Hello World
 - LEC-02—History, Syntax & Variables, Ho1
 - LEC-03—Numbers, Expressions & Calculations, Ho2
 - LEC-04—Functions & Decisions, Ho4
- **Week 2** – Monday Holiday/Tuesday **PE01** & **PE02**
 - LEC-05—Characters, Strings & References, Ho4
 - LEC-06—Programming with Loops, Ho5
 - LEC-07—More on Loops, Ho6
 - LEC-08—Function Libraries, Ho7 & Ho8