
# Welcome to CMSC 240!

**CMSC 240 Software Systems Development**

Fall 2024

# Today

- Introductions
- Course logistics
- Motivation
- Hello C++
- Environment setup
- In-class coding exercise

RICHMOND


Introductions

# Dr. Doug Szajda

When I started here.  And had no gray hair

- Prof. Doug Szajda
  *"Shade-uh" or "Prof S"*
- He/Him

- BS in Math from Lafayette College
- MS, PhD in Math from UVA
- MCS from UVA, Postdoc at UMIACS (U Maryland)
- UR CS faculty since 2001
- CS Department Chair
- Research: Computer S&P, ML

# Dr. Doug Szajda

Monaco

- Things I like (in no particular order)
  - Travel
  - Learning Italian
  - Cooking
  - Godzilla movies
  - Lego video games
  - Programming
  - Dogs (and cats)

RICHMOND

# Dr. Doug Szajda

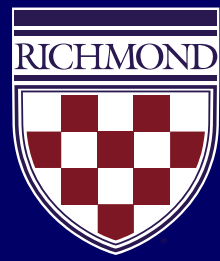
La Fontana di Trevi

- Things I like (in no particular order)
  - Travel
  - Learning Italian
  - Cooking
  - Godzilla movies
  - Lego video games
  - Programming
  - Dogs (and cats)

RICHMOND

# The Pets: Frosty, Ethan, and Indy

# Ask me anything



# Classroom Meet and Greet

1. Introduce yourself to a person near you

2. Introduce yourself to a different person near you

- Potential conversation topics:
  - What are some of the things that you like?
  - Who are your favorite pets?
  - Why do you want to take this class?

# Student Introductions

- Name
- Pronouns
- Major
- Class year
- Food you cannot live without

# Course Logistics

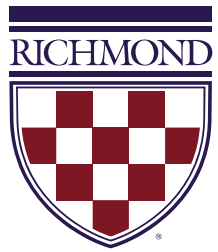
# Classroom Norms

- Questions are always welcome!!
  - Ask them at any time
- "I don't know" is okay
- Be curious
- Treat peers and instructors with kindness and respect
- Communication is key!
- Seek support when needed

# Where All Class Information Can Be Found

https://cmsc240-f24-dcs.github.io/cmsc240-f24-s2.github.io/

(Don’t ask.  Github classroom just seemed to want to make this URL like this, regardless of what Dr. Balash and I tried.)



# How to Communicate With Me

- Public or private post on Ed
  - https://edstem.org

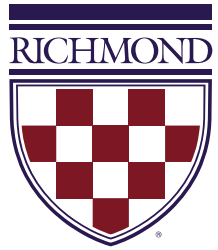
- After class or in office hours - 219 Jepson Hall
  - Tue 1:30 – 2:30
  - Fri 1:00 – 2:00
  - and by appointment

- Email
  - dszajda@richmond.edu

RICHMOND

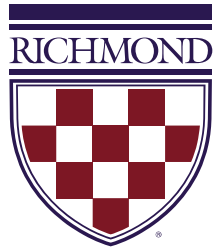
# Course Outline

- **Weeks 1-5** Introduction to C++ programming
  - Syntax, memory management, libraries, file IO
- **Weeks 6-10** Object-oriented programming
  - Abstraction, polymorphism, inheritance, encapsulation
- **Weeks 11-15** Software systems development
  - UML, design patterns, testing, debugging



# Learning Outcomes

- Experience modern* C++ programming
- Gain familiarity with Unix/Linux environments
- Understand the software development life cycle
- Practice object-oriented programming and design
- Understand design patterns, reuse, and usability
- Exposure to version control systems
- Demonstrate skill in software testing and debugging



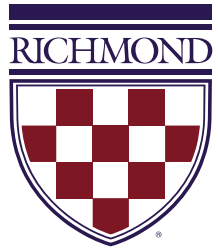
# Lecture

- Tuesdays and Thursdays Jepson 103A
- Will usually include in-class exercises
- In-class exercises will be due one week from when they are assigned (except during break)
- Regular attendance is expected
- Students who are sick should not attend class
- Notify me in advance of the absence, if possible

RICHMOND

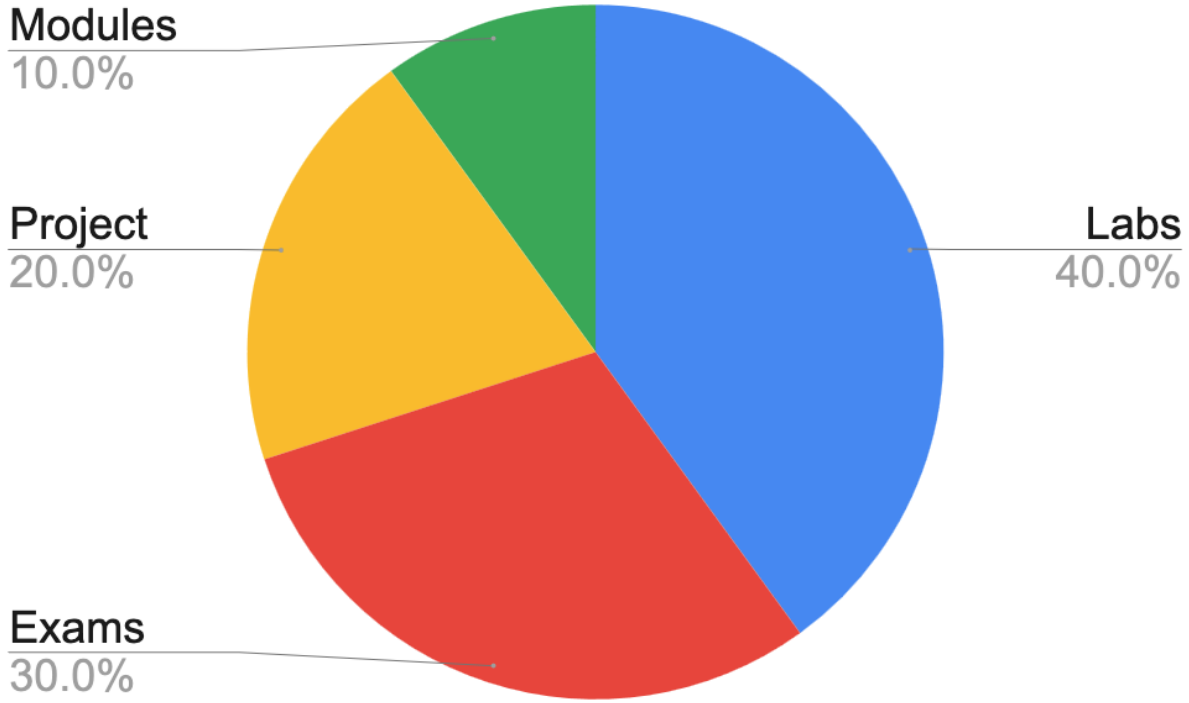
# Labs

- Fridays in Jepson 103A
- Lab assignments done individually and in groups
  - but will always be turned in individually
- Lab assignments are typically due at 5:00 pm on the night prior to the next lab (except during break)
- Please ask for help from me or the lab assistant (if we get one).



# Coursework and Grading

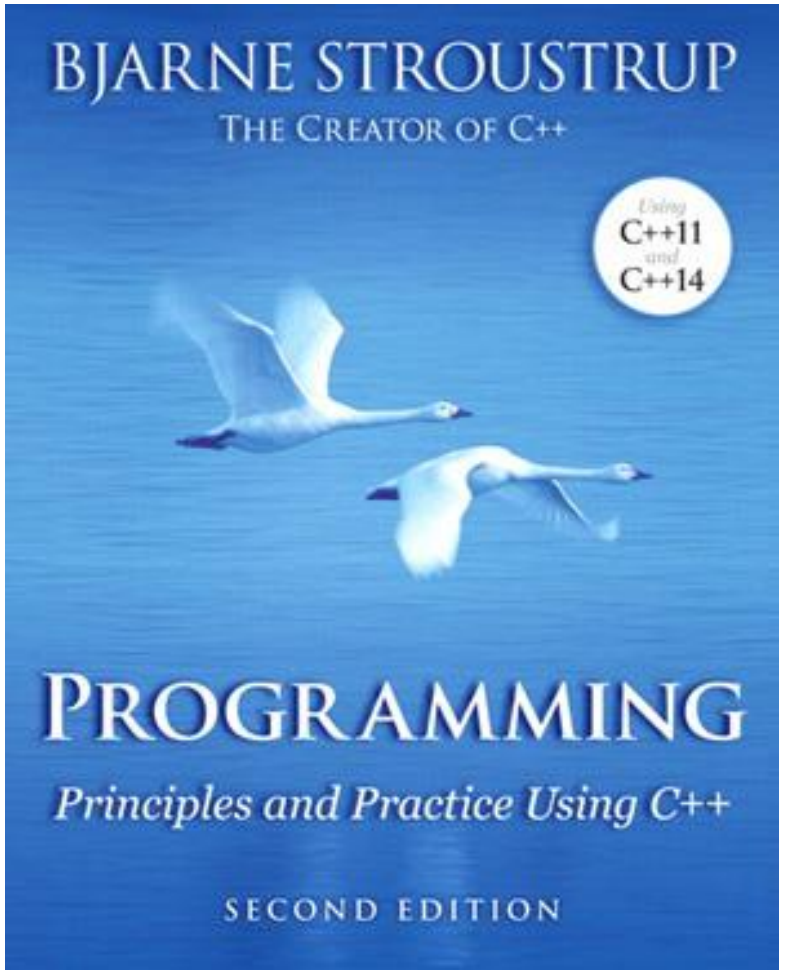
- Modules (In-class coding exercises)
- Lab assignments
- Programming project
- Midterm exam
- Final exam




RICHMOND

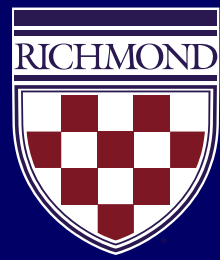
# Textbook

- Free electronically from the UR library
- Reading assignments




RICHMOND

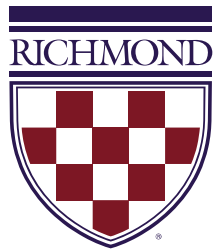
# Ask me questions




Motivation

# I have some questions for you!

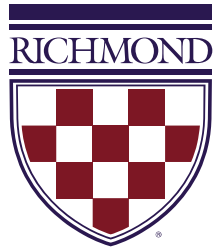
- If you are building large software (e.g., an operating system, a video game, a passenger reservation system), what problems might you have that you would not necessarily have with, say, a CMSC 150 or CMSC 221 project? Can you think of tools it might be nice to have to solve some of these problems? (By tools, I don't mean specific software applications, but rather "a program that does x…")



# I have some questions for you!

- Why C++?

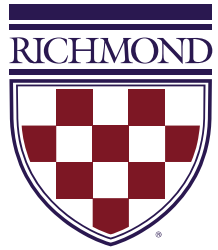
- Why so many programming languages?
  - Today, between 250 and 2500: https://en.wikipedia.org/wiki/List_of_programming_languages
  - Historically, almost 9000



# C++ is a Very Popular Language

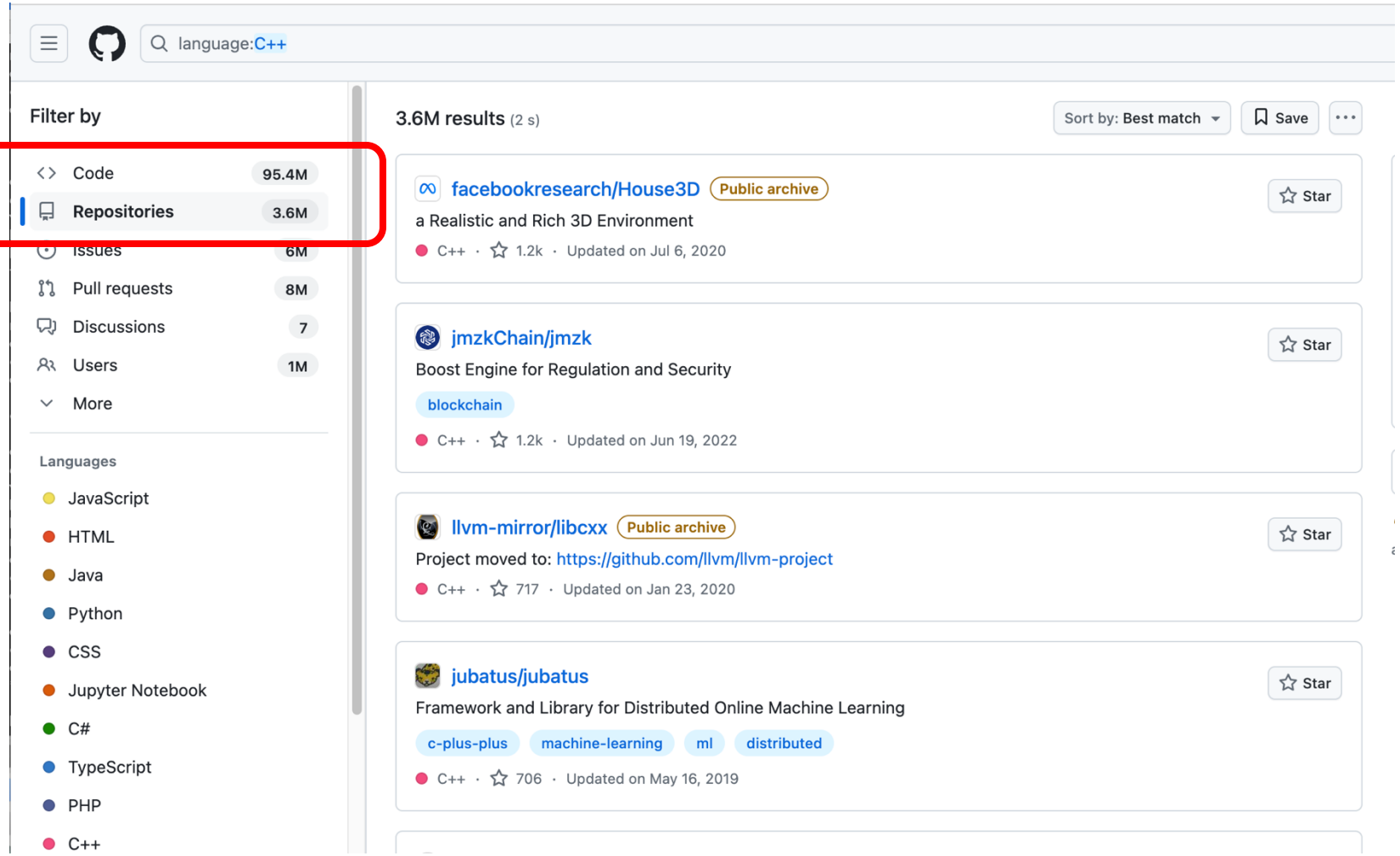
| Aug 2024 | Aug 2023 | Change | Programming Language | Ratings | Change |
|---|---|---|---|---|---|
| 1 | 1 | | Python | 18.04% | +4.71% |
| 2 | 3 | ↑ | C++ | 10.04% | -0.59% |
| 3 | 2 | ↓ | C | 9.17% | -2.24% |
| 4 | 4 | | Java | 9.16% | -1.16% |
| 5 | 5 | | C# | 6.39% | -0.65% |
| 6 | 6 | | JavaScript | 3.91% | +0.62% |
| 7 | 8 | ↑ | SQL | 2.21% | +0.68% |

TIOBE Index for August 2024




Source: https://www.tiobe.com

# Many Open-Source Projects




https://github.com

RICHMOND

# Cool Things Were Built With C++

amazon.com

Google

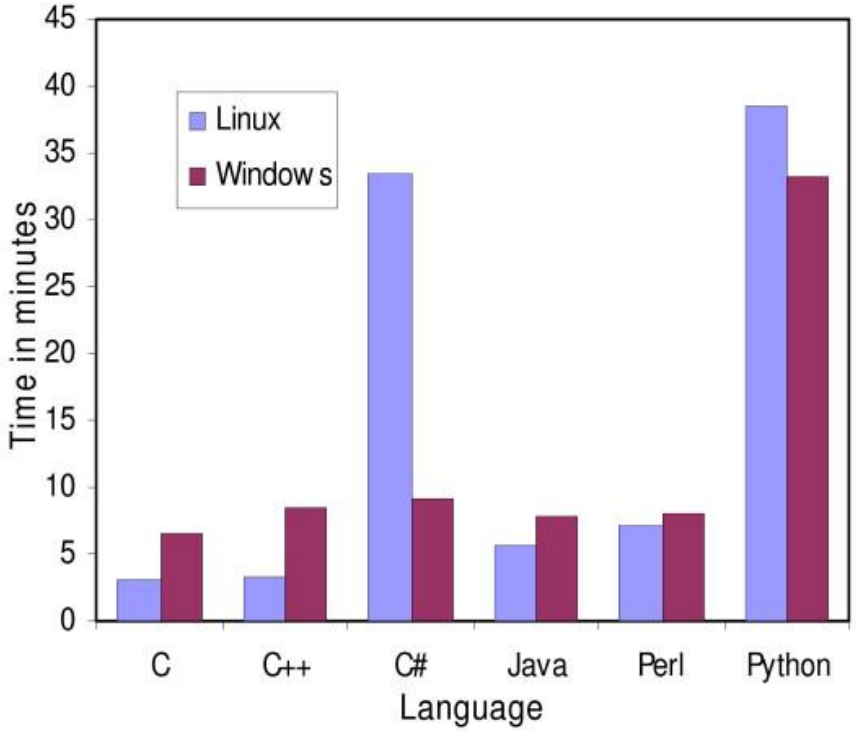
Windows

RICHMOND

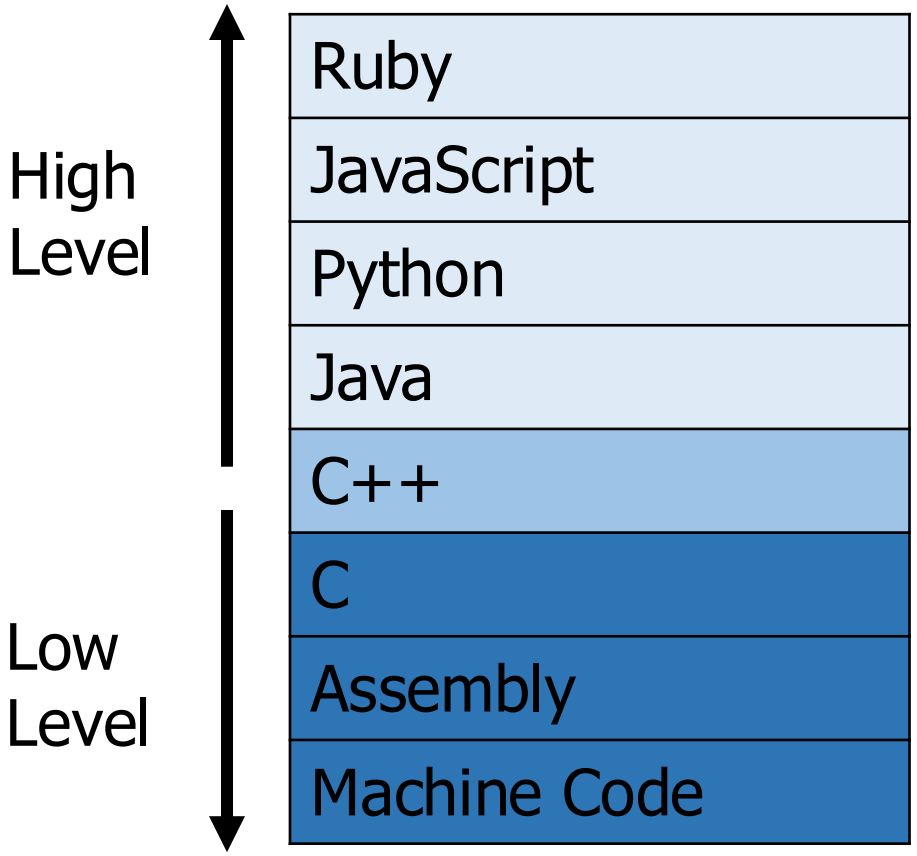
Images source: https://commons.wikimedia.org

# What Makes C++ Great?

## Speed: It's Fast!



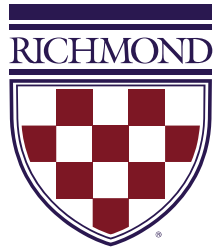

## Low-level control



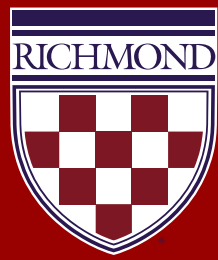

Source: https://www.researchgate.net

# Foundational Software Development Skills

- Object-oriented design
- Software development life cycle
- Design patterns and code reuse
- Version control systems
- Testing and debugging

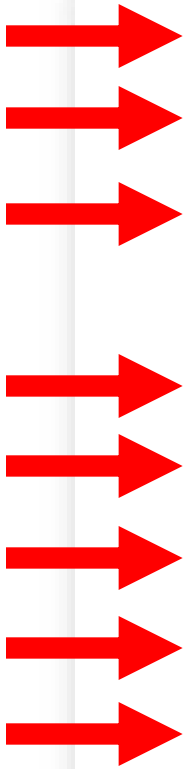


# What motivates you?




Hello C++
RICHMOND

# Writing Your First C++ Program



```
// This program outputs the message "Hello, World!"
#include <iostream>
using namespace std;

int main()
{
    cout << "Hello, World!" << endl;
    return 0;
}
```

RICHMOND

# Writing Your First C++ Program

```
// This program outputs the message “Hello, World!”
#include <iostream>
// Without using namespace std

int main()
{
    std::cout << "Hello, World!" << std::endl;
    return 0;
}
```

RICHMOND

# Writing Your First C++ Program

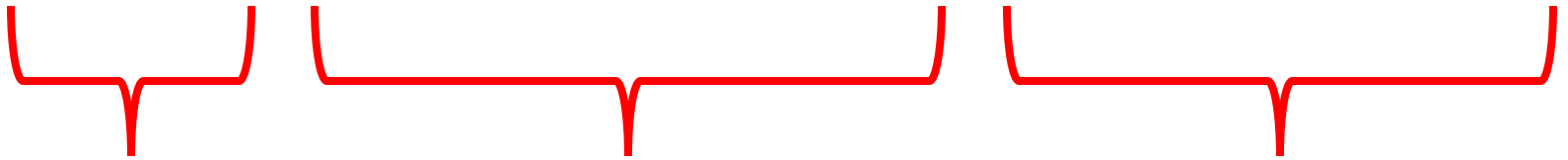
```
// This program outputs the message "Hello, World!"
#include <cstdio>

int main()
{
    printf("Hello, World!\n.");
    // ^ a C function
    return 0;
}
```

RICHMOND

# Compile & Execute Your Program

```
g++ hello.cpp -o hello
```



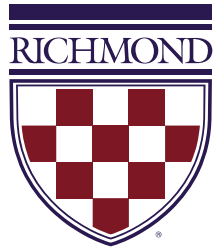
The C++ compiler

The source code file name

Using the `-o` option allows you to name the executable file

```
./hello
```

indicates that the executable resides in the current directory




Environment Setup
RICHMOND

# Development Environment

- All work will be compiled, run, tested and graded on the computer science Linux machines:
  - cs01 – cs06.richmond.edu

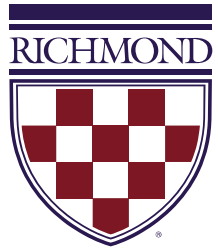

- GitHub classroom for all assignments
  1. Accept the assignment
  2. Clone repository using VSCode with remote-ssh
  3. Make updates to the code and README.md file
  4. Add (Stage), Commit, and Sync changes

RICHMOND

# Development Environment

1. Open a terminal
2. ssh your_UR_netid@cs01.richmond.edu
   For example: for me, ssh dszajda@cs01.richmond.edu

3. Follow instructions:
   - https://cmsc240-f24-dcs.github.io/cmsc240-f24-s2.github.io/guides/vscode-ssh




In-Class Coding Exercise
RICHMOND