



UNIVERSITY OF  
RICHMOND

Welcome to CMSC 240!

CMSC 240 Software Systems Development

# Today

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





# Introductions





# Dr. David Balash



Professor Balash

*“Ba-lish”*

He/Him

- BS in computer engineering Iowa State
- Two-decade career as a software engineer
- MS and PhD in computer science from GW
- Research: Computer S&P

Faculty page: <https://cs.richmond.edu/faculty/dbalash>

Homepage: <https://davidbalash.github.io>

# Dr. David Balash



## Things I like

- 🎓 Education/Learning
- 🧑 Hiking
- 🚴 Cycling
- 🎸 Guitars
- ♟ Board games
- 💻 Programming
- 🐱 Cats

Ask me anything



# Assignment 1

**Task:** Create a personal introduction slide and post it to the **introductions** channel on the course Slack workspace

**Due:** Friday

**Points:** 5

Be Creative

Name

Dr. David Balash

Photo



Faculty page: <https://cs.richmond.edu/faculty/dbalash>

Homepage: <https://davidbalash.github.io>



Professor Balash

*"Ba-lish"*

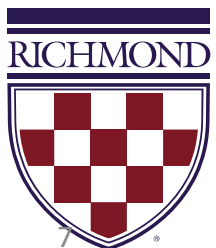
He/Him

- BS in computer engineering Iowa State
- Two-decade career as a software engineer
- MS and PhD in computer science from GW
- Research: Computer S&P

Pronunciation

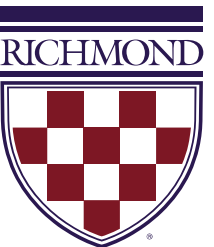
Pronouns

Personal Introduction



# Classroom Meet and Greet

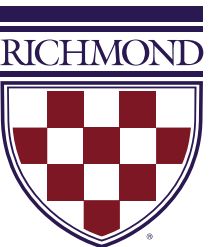
1. Introduce yourself to a person near you
  2. Introduce yourself to a different person near you
- Potential conversation topics:
    - What are some of the things that you like?
    - Who are your favorite pets?
    - Why do you want to take this class?





# Student Introductions

- Name
- Pronouns (optional)
- Major
- Class year
- Favorite snack food





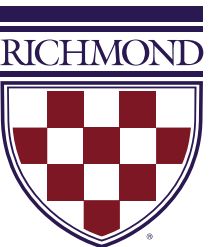
# Course Logistics





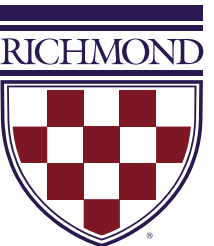
# Classroom Norms

- 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

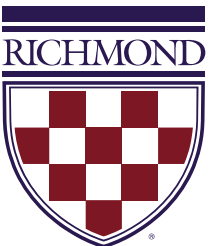
<https://cmssc240-f24.github.io>





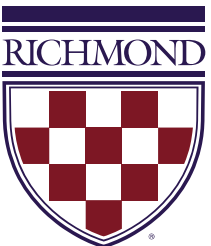
# How to Communicate With Me

- Slack workspace
  - <https://cmssc240-f24.slack.com>
- After class or in office hours - 223 Jepson Hall
  - Tue 4:30PM - 6:00PM
  - Thr 4:30PM - 6:00PM
  - and by appointment
- Email
  - [david.balash@richmond.edu](mailto:david.balash@richmond.edu)



# Course Outline

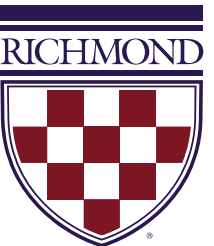
- **Weeks 1-5** Introduction to C++ programming
  - Syntax, memory management, libraries, file IO
- **Weeks 6-10** Object-oriented programming
  - Abstraction, polymorphism, inheritance, encapsulation
- **Weeks 11-15** Software systems development
  - UML, design patterns, testing, debugging





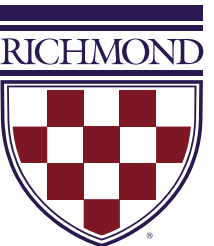
# Learning Outcomes

- 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



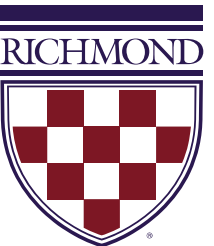
# Lecture

- Tuesdays and Thursdays Jepson G04
- 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



# Labs

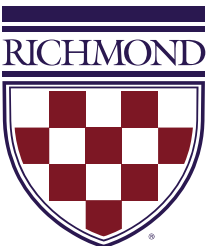
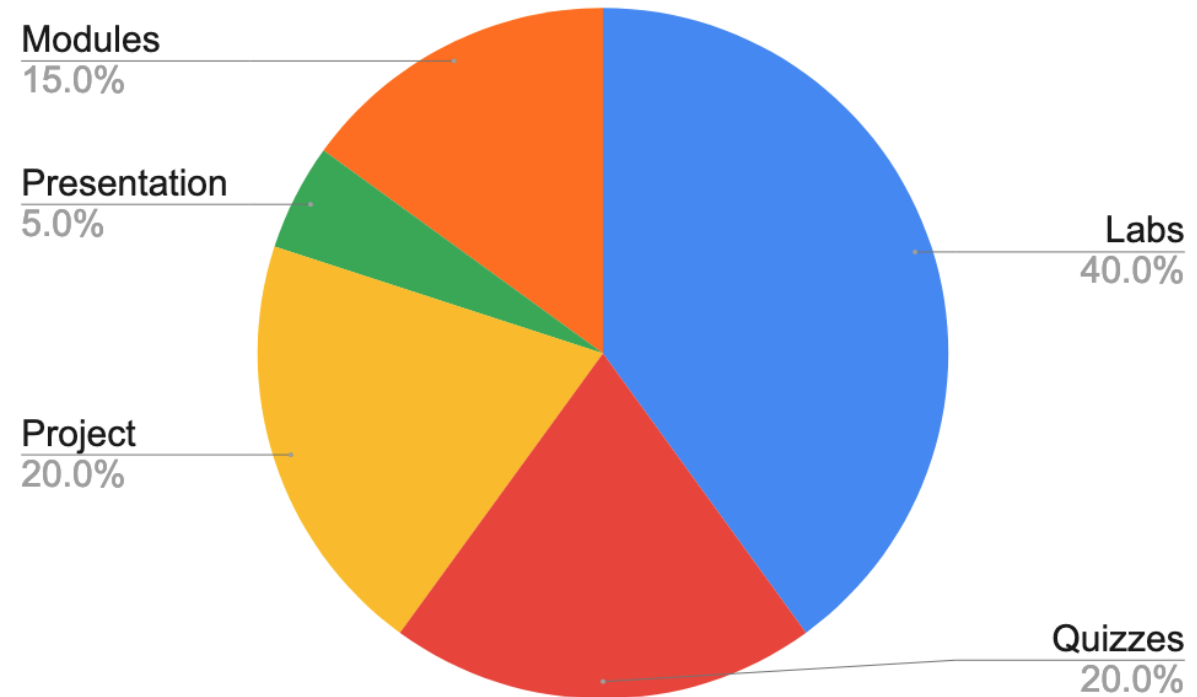
- Fridays in Jepson G03
- Lab assignments done individually and in groups
  - but will always be turned in individually
- Lab assignments are typically due at 11:59 pm on the night prior to the next lab (except during break)
- Please ask for help from me or the lab assistant





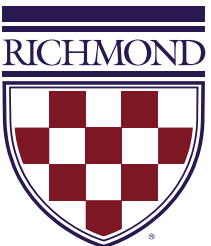
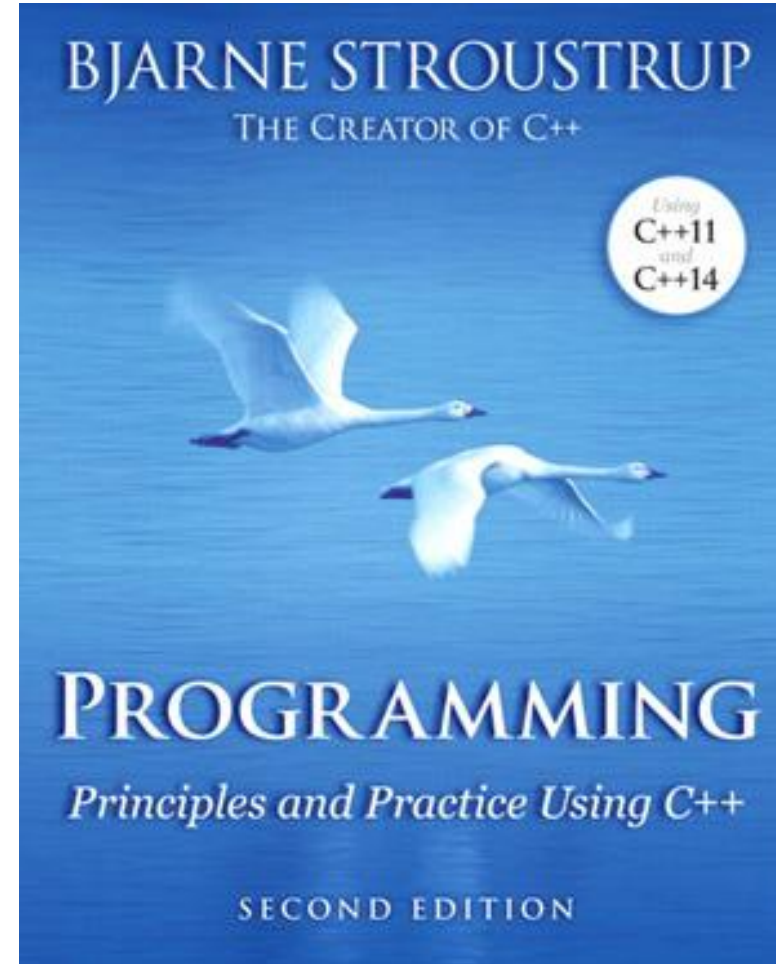
# Coursework and Grading

- Modules (In-class coding exercises)
- Lab assignments
- Programming project
- Project Presentation
- 4 Quizzes (5% each)



# Textbook

- Free electronically from the UR library
- Reading assignments



Ask me a question





# Motivation

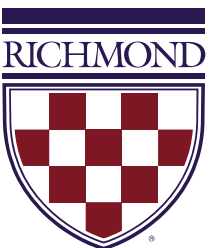




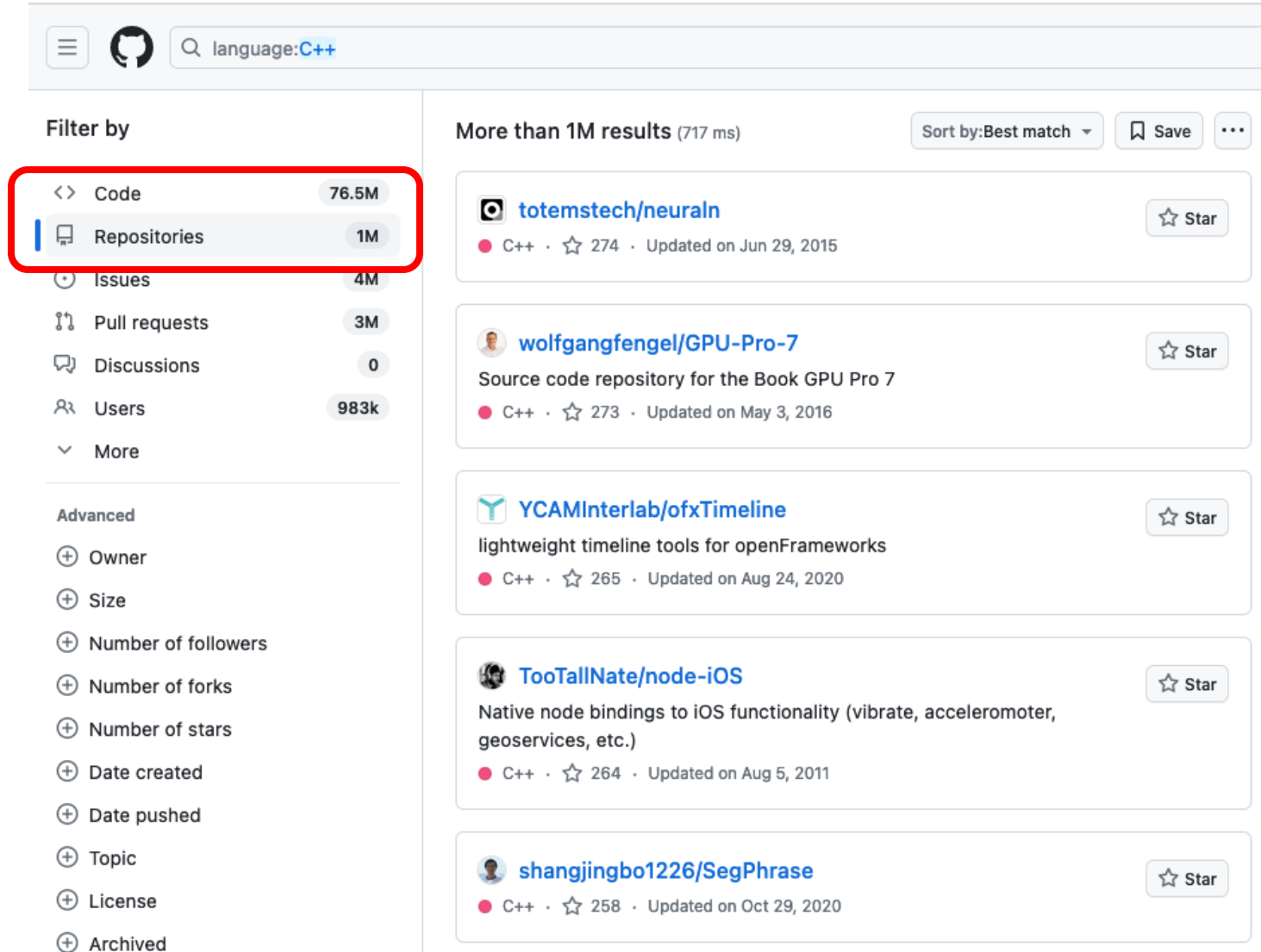
# C++ is a Very Popular Language

Aug 2023	Aug 2022	Change	Programming Language		Ratings	Change
1	1			Python	13.33%	-2.30%
2	2			C	11.41%	-3.35%
3	4	▲		C++	10.63%	+0.49%
4	3	▼		Java	10.33%	-2.14%
5	5			C#	7.04%	+1.64%
6	8	▲		JavaScript	3.29%	+0.89%
7	6	▼		Visual Basic	2.63%	-2.26%

TIOBE Index for August 2023



# Many Open-Source Projects



The screenshot shows the GitHub search interface with the search bar set to "language:C++". The left sidebar shows the "Filter by" section with "Repositories" highlighted in a red box, indicating 1M results. Other filters include Code (76.5M), Issues (4M), Pull requests (3M), Discussions (0), Users (983k), and More. The main content area displays "More than 1M results (717 ms)" and lists several repositories. The first repository is "totemstech/neuraln" with 274 stars and updated on Jun 29, 2015. The second is "wolfgangfengel/GPU-Pro-7" with 273 stars and updated on May 3, 2016. The third is "YCAMInterlab/ofxTimeline" with 265 stars and updated on Aug 24, 2020. The fourth is "TooTallNate/node-iOS" with 264 stars and updated on Aug 5, 2011. The fifth is "shangjingbo1226/SegPhrase" with 258 stars and updated on Oct 29, 2020.

language:C++

Filter by

- <> Code 76.5M
- Repositories 1M**
- Issues 4M
- Pull requests 3M
- Discussions 0
- Users 983k
- More

Advanced

- + Owner
- + Size
- + Number of followers
- + Number of forks
- + Number of stars
- + Date created
- + Date pushed
- + Topic
- + License
- + Archived

More than 1M results (717 ms) Sort by: Best match Save ...

**totemstech/neuraln** Star  
C++ · ☆ 274 · Updated on Jun 29, 2015

**wolfgangfengel/GPU-Pro-7** Star  
Source code repository for the Book GPU Pro 7  
C++ · ☆ 273 · Updated on May 3, 2016

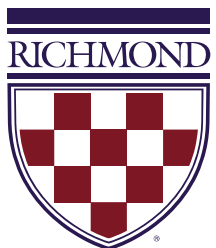
**YCAMInterlab/ofxTimeline** Star  
lightweight timeline tools for openFrameworks  
C++ · ☆ 265 · Updated on Aug 24, 2020

**TooTallNate/node-iOS** Star  
Native node bindings to iOS functionality (vibrate, accelerometer, geoservices, etc.)  
C++ · ☆ 264 · Updated on Aug 5, 2011

**shangjingbo1226/SegPhrase** Star  
C++ · ☆ 258 · Updated on Oct 29, 2020

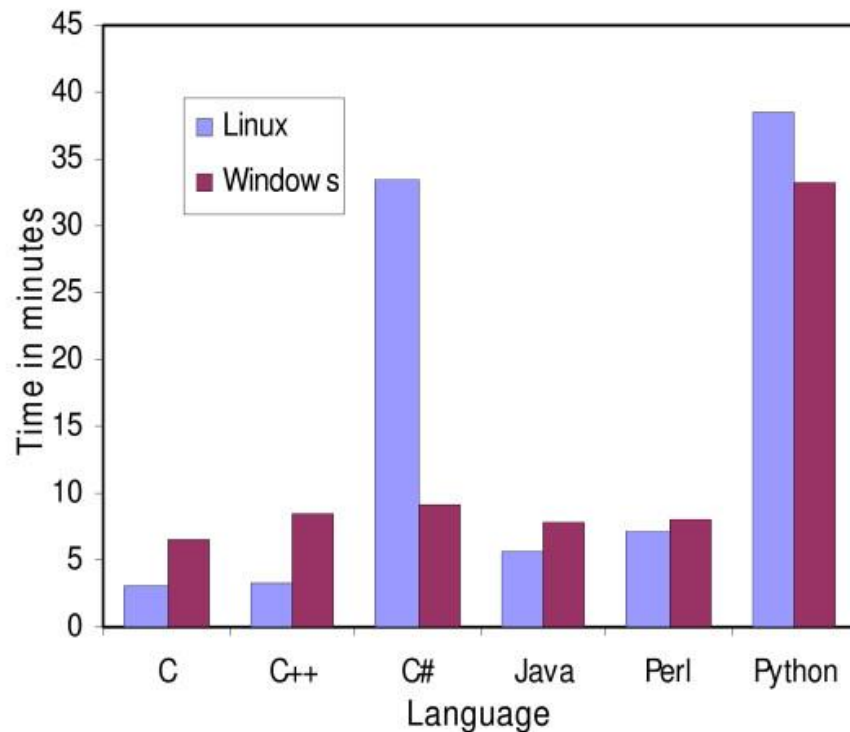


# Cool Things Were Built With C++



# What Makes C++ Great?

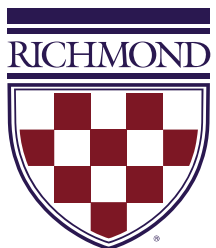
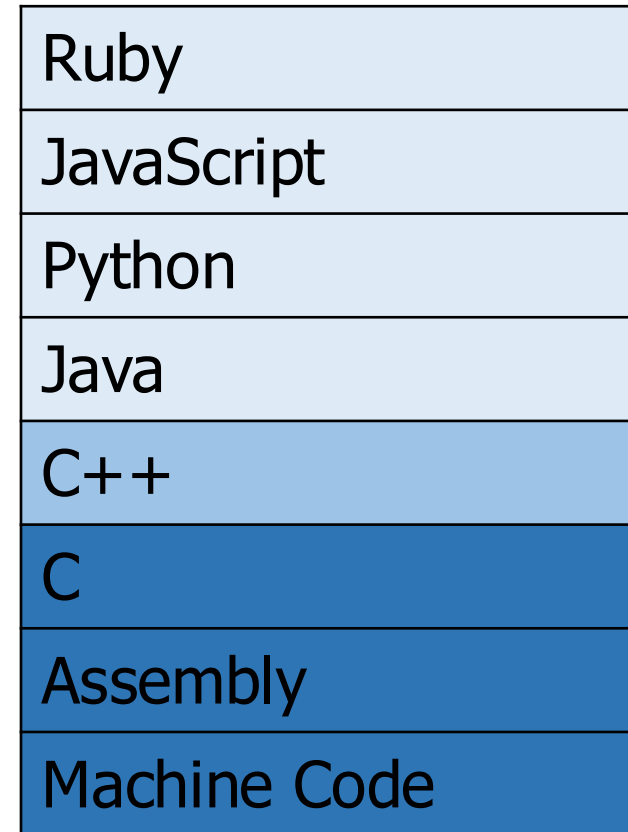
Speed: It's Fast!



Low-level control

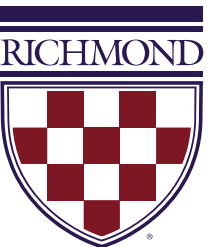
High Level

Low Level



# Foundational Software Development Skills

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





An aerial photograph of a university campus. In the center, a tall, ornate brick tower with Gothic-style architecture rises above the surrounding trees. The campus is lush with greenery, including large trees with yellow and green foliage in the foreground and dense evergreens in the background. Several paved walkways are visible, with a few people walking on them. The sky is clear and blue.

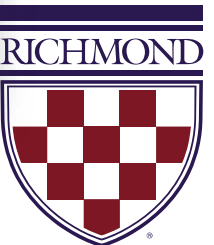
# Hello C++





# 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;  
→ }
```



# Writing Your First C++ Program

*// This program outputs the message "Hello, World!"*

`#include <iostream>`

*// Without using namespace std*

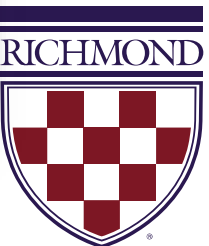
`int main()`

`{`

`std::cout << "Hello, World!" << std::endl;`

`return 0;`

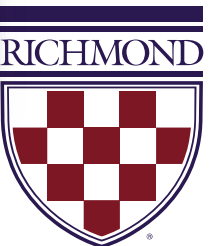
`}`






# 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;  
}
```



# Compile & Execute Your Program

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




The C++  
compiler

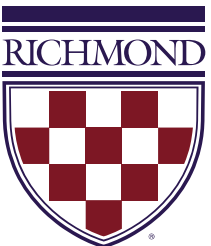
The source  
code file name

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

```
./hello
```



indicates that the executable  
resides in the current directory





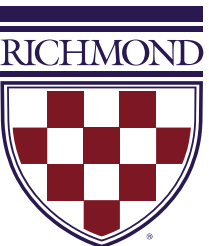
# Environment Setup





# 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



# Development Environment

1. Open a terminal
2. `ssh your_UR_netid@cs01.richmond.edu`  
For example: my netid is **dbalash**@cs01.richmond.edu
3. Follow instructions:
  - <https://cmssc240-f24.github.io/guides/vscode-ssh>





# In-Class Coding Exercise

