

# TOOLS

# C++ DATA TYPES

---

Problem Solving with Computers-I  
Chapter 1 and Chapter 2

**C++**

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Hola, Facebook!";
    return 0;
}
```



CLICKERS OUT – FREQUENCY AB

# Programming in the unix environment

- File System
- File versus directory
- Path: Relative vs absolute
- Navigating the file system with unix commands:
  - ls (list)
  - mv (move)
  - cp (copy)
  - pwd (print working directory - show me the full path)
  - mkdir (make directory)
  - cd (change directory)
- Using `.`(dot) and `~` when specifying relative paths

# Writing a simple C++ program

```
// hello.cpp
#include <iostream>

using namespace std;

int main() {
    cout << "Hello CS 16!" << endl;
    return 0;
}
```

# Compiling and executing a C++ program

*name of executable*



*source file*



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

```
$ ./hello
```

```
Hello CS 16!
```

```
$
```

# Vim survival skills

- Learn the basic 8: [https://ucsb-cs16.github.io/topics/vim\\_basic\\_eight/](https://ucsb-cs16.github.io/topics/vim_basic_eight/)
- Open a new file: vim <filename>
  1. Quit without saving
  2. Enter code
  3. Save, Save and quit
  4. Copy paste, cut and paste
  5. Search, Search and replace
  6. Show line numbers
  7. Go to a line number
  8. Save as

# Program compilation

What does it mean to “compile” a C++ program?

- A. Write the implementation of the program in a .cpp file
- B. Convert the program into a form understandable by the processor
- C. Execute the program to get an output
- D. None of the above

## Kinds of errors

Which of the following types of errors is produced if our program divides a number by 0?

- A. Compile-time error
- B. Run-time error
- C. Both A and B
- D. Neither A or B

# Review: C++ Variables and Datatypes

- **Variables** are containers to store data
- **C++** variables must be “declared” before they are used by specifying a datatype
  - `int`: Integers
  - `double`: floating point numbers
  - `char`: characters
  - `string`: sequence of characters e.g. “apple”



# Naming variables

- **Variable names must:**
  - Start with an alphabet (a-z, A-Z) or underscore(\_)
  - Other characters can be alphanumeric and underscore characters
  - No spaces or other special characters.
- **C++ is case-sensitive**
  - 'x' and 'X' are considered different variables.

# C++ Uninitialized Variables

- Value of uninitialized variables is “undefined”
- Undefined means “anything goes”
- Can be a source of tricky bugs
- What is the output of the code below?

*undefined*  
*'a' and 'b' have junk values. Explicitly initialize variables in C++.*

```
int main() {  
    int a, b;  
    cout<<"The sum of "<< a << " and " << b<< " is:"<< a+b<<endl;  
    return 0;  
}
```

# Variable Assignment

- The values of variables can be initialized...

```
int myVariable = 0;
```

**-or-**

```
int myVariable;  
myVariable = 0;
```

- ...or changed on the fly...

```
int myVariable = 0;  
myVariable = 5 + 2;
```

# Variable Assignment

- ...or even be used to update the same variable!

```
int myVariable = 0;  
myVariable = 5 + 2;  
myVariable = 10 - myVariable;  
myVariable = myVariable==0;
```

## C++ types in expressions

```
int i = 10;
```

```
double sum = 1/i;
```

*Since both 1 and i are integers  
result is cast as an integer*

What is printed by the above code?

A. 0

B. 0.1

C. 1

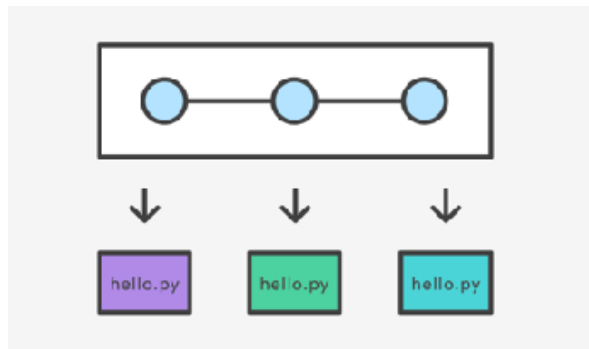
D. None of the above

# What is git?

Git is a version control system (VCS).

A VCS allows you to keep track of changes in a file (or groups of files) over time

Git allows you to store code on different computers and keep all these different copies in sync



# Git Concepts

**repo** (short for repository): a place where all your code and its history is stored

Remote repo: A repo that exists on the web (in our case [github.com](https://github.com))

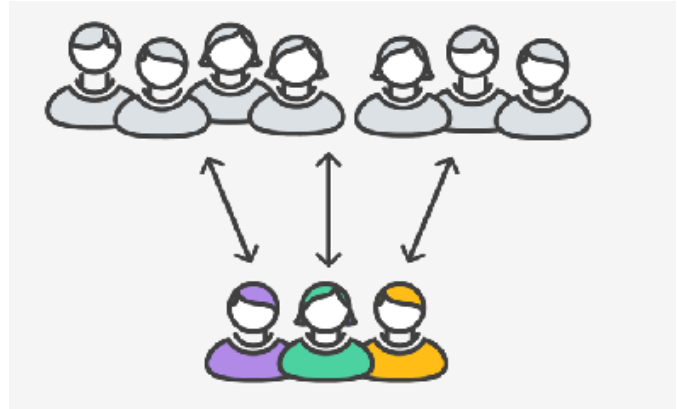
## In class demo

- **creating a repo on github.com**
- **adding collaborators to the repo**
- **adding files to the repo**
- **Updating files in a remote repo using a web browser**
- **Viewing the version history**



# Why are we learning git in this class?

- Collaborate
- Share code ownership
- Work on larger projects
- Provide feedback on work in progress
- Learn professional software development tools




# Creating a repo on the cloud ([www.github.com](http://www.github.com))

Navigate to [www.github.com](http://www.github.com) and create a repo on the internet

## Create a new repository

A repository contains all the files for your project, including the revision history.

Owner

 ucsb-cs24-s18 ▾

Repository name

lab00\_jgaucho\_aliy

Great repository names are short and memorable. Need inspiration? How about **potential-lamp**.

Description (optional)

Public

Anyone can see this repository. You choose who can commit.

Private

You choose who can see and commit to this repository.

Initialize this repository with a README

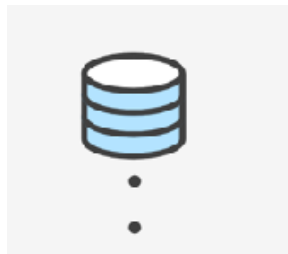
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: C++ ▾

Add a license: None ▾



Create repository



## Remote repo

# Next time

- Control Flow