Software Systems

Day 2 – Getting Started with C

Agenda

- Announcements
- Syllabus Questions
- Computational Setup
- GitHub Setup
- Getting Started with C

Announcements

- We've had a few drops, so you might get in if you're still on the waitlist.
  - If you just got in off the waitlist, I'll extend your quiz deadline - message me.
- If you're not on the Canvas, message me ASAP.
- Join the class Discord.

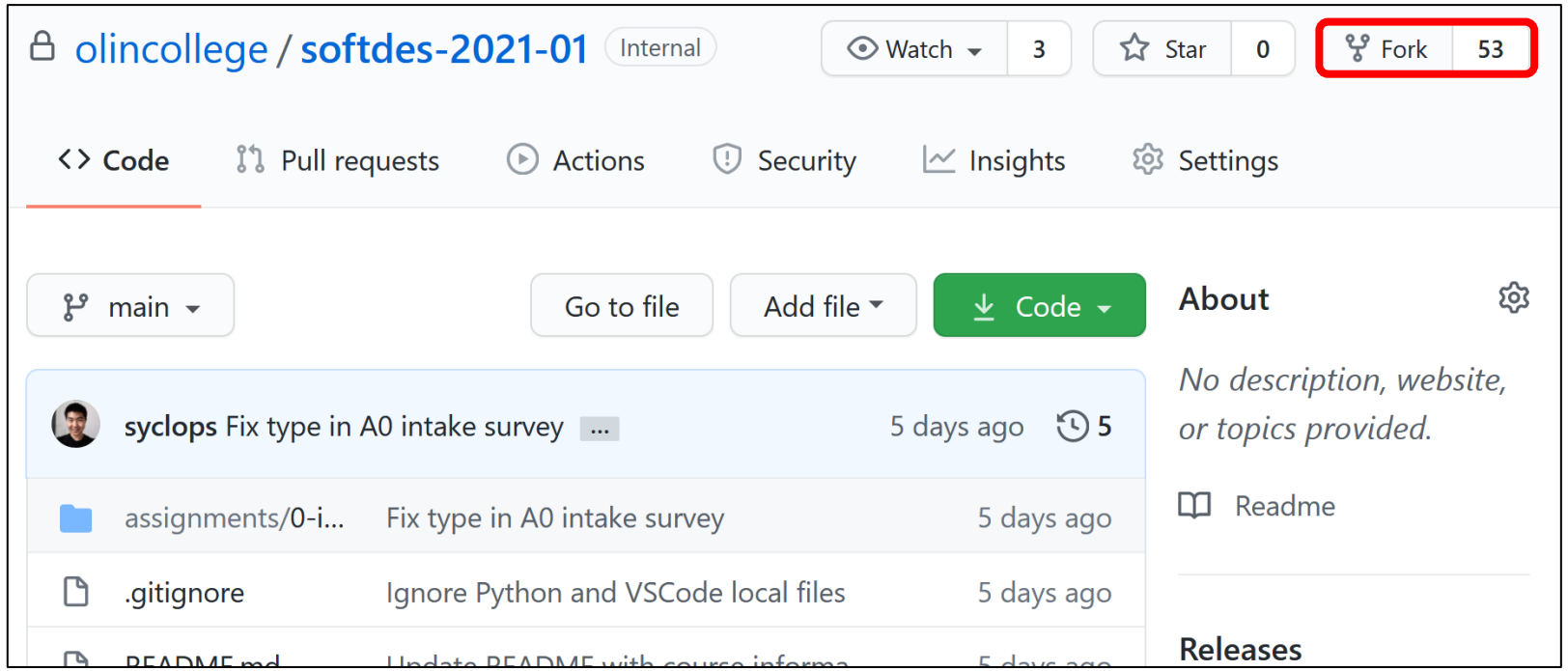
Syllabus Questions

Computational Setup

- If you are still working on the computational setup, now is a good time to get caught up.

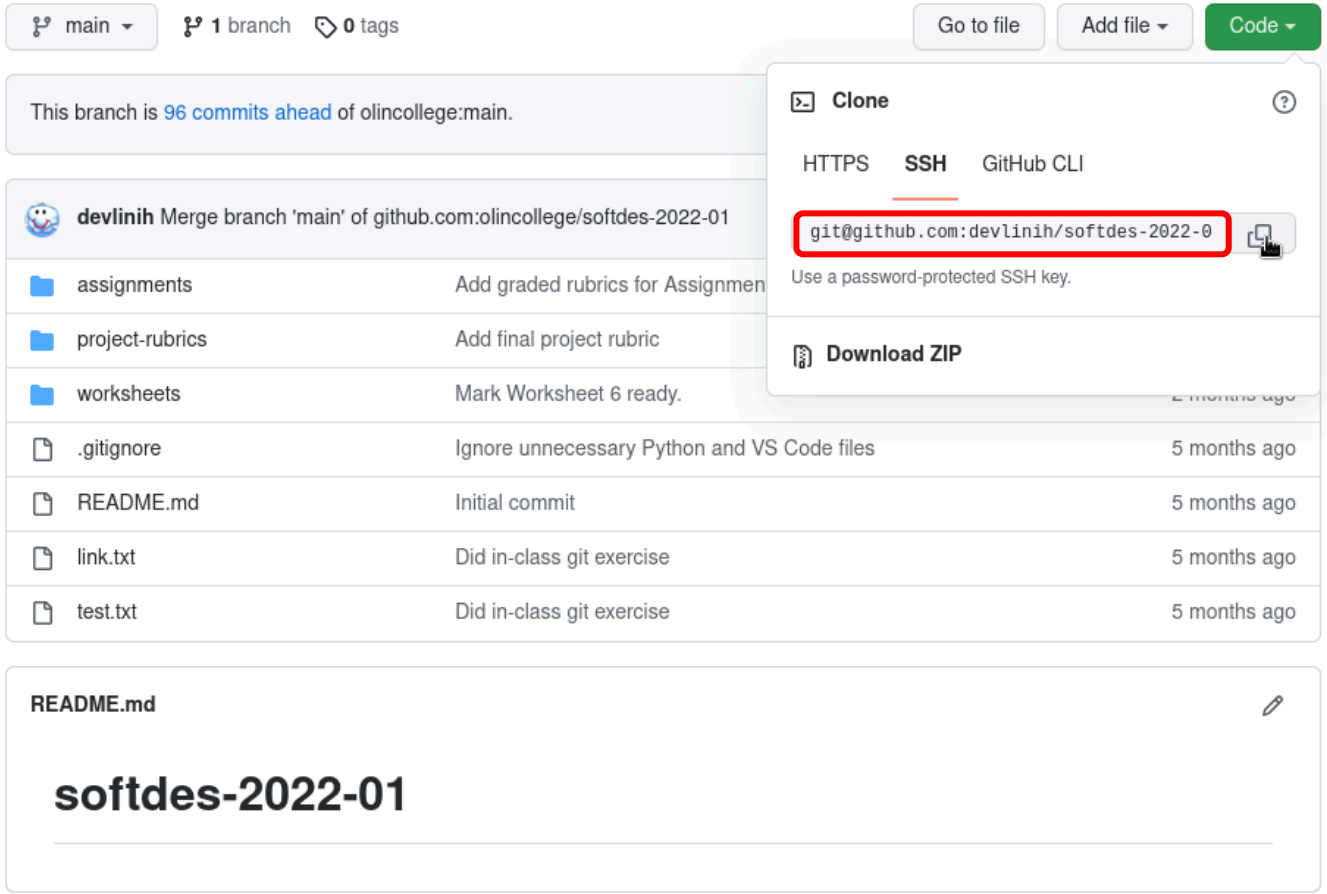
Git Setup

- Now, head to https://github.com/olincollege/softsys-20XX-YY (replacing XX and YY as appropriate) and click to fork the repository.



Git Setup

- Clone your fork:
  - git clone <URL>



Git Setup

- Set up remotes:
  - Copy the olincollege/softsys-20XX-YY URL.
  - Change to your clone directory.
  - git remote add upstream <URL>
- Check:
  - git remote –v should show:

```
origin  git@github.com:username/softsys-20XX-YY.git (fetch)
origin  git@github.com:username/softsys-20XX-YY.git (push)
upstreamgit@github.com:olincollege/softdes-202XX-YY.git (fetch)
upstreamgit@github.com:olincollege/softdes-20XX-YY.git (push)
```

Head First C, Chapter 1

- A whirlwind tour of C.
  - The main() function
  - Compiling and running C code
  - Character arrays/strings
  - Conditional logic (if/else, booleans)
  - Switch statements
  - Loops
  - Functions
- Remember, it’s a tour – you don’t have to master everything.

Exercise: Hello World

- Follow along to write a hello world function in a file called hello.c.
- Then compile and run it with gcc -o hello hello.c

Exercise: Printing

- Hello world is often written with printf instead of puts.
- What's the difference? Try it out.
- If you want, look it up on cppreference.com

Common C Gotchas: Strings (and Formatting)

- C doesn't really have strings.
- Use char* instead.
- Also, note that char* x and char *x are the same.

Exercise: Format Strings

- In your main function, define a separate string with your name.
- Then use printf to print "Hello, <your name>!" with that string.

Common C Gotchas: void

- If a function takes no arguments, you need to write void in the parentheses.
- The function will work if you leave it off, but it can lead to some baffling results.

Common C Gotchas: scanf

- scanf is how you read input from standard in.
- But you need to read into the address:

```
int x, y;
scanf("%d %d", &x, &y);
```

Exercise: scanf

- Modify your earlier greeting function to read a name as input from the user.
- Use scanf to read a string, and then use it in printf

Common C Gotchas: Conditionals

- Python

```
if delete_request:
    send_confirmation(user.email)
    delete(user)
```

- C

```
if (delete_request)
  send_confirmation(email);
  delete(user);
```

Indentation doesn't matter in C.

So delete will **always** run here.

By default, `if` only "binds" to the next line.

Common C Gotchas: Conditionals

- Python

```
if delete_request:
    send_confirmation(user.email)
    delete(user)
```

- C

```
if (delete_request)
  send_confirmation(email);
  delete(user);
```

Common C Gotchas: Assignment

- Python

```
x = 0
if x = 1:
    print("x is 1.")
else:
    print("x isn't 1.")
```

Syntax error!

- C

```
int x = 0;
if (x = 1) {
  printf("x is 1.\n");
} else {
  printf("x isn't 1.\n");
}
```

x = 1 sets x to 1 **and** returns 1

Any nonzero integer is true.

Always prints x is 1.

Exercise: Conditionals

- Write a program that reads in an integer using scanf.
- Then, if the integer is odd, print "<int> is odd."
- Otherwise, print "<int> is even."

Common C Gotchas: Switches

- Python

```
i = 0
if i == 0:
  i += 1
elif i == 1:
  i *= 2
else:
  i = 10
print(i)
```

Prints 1

- C

```
int i = 0;
switch(i) {
  case 0:
    i += 1;
  case 1:
    i *= 2;
  default:
    i = 10;
}
printf("%d", i);
```

Switch statements jump to the matching case

Cases "fall through": execute **all** lines to the end of the block

Prints **10**

Common C Gotchas: Switches

- Python

```
i = 0
if i == 0:
  i += 1
elif i == 1:
  i *= 2
else:
  i = 10
print(i)
```

- C

```
int i = 0;
switch(i) {
  case 0:
    i += 1;
    break;
  case 1:
    i *= 2;
    break;
  default:
    i = 10;
}
printf(“%d”, i);
```

Exercise: Switch

- Use scanf to read in a string of the form "x + y" or "x * y", where x and y are integers.
  - So read in something like 6 * 7.
- Then use a switch statement to evaluate the result.
  - If the operator is not + or *, then return –1.
- Finally, print the result using printf.

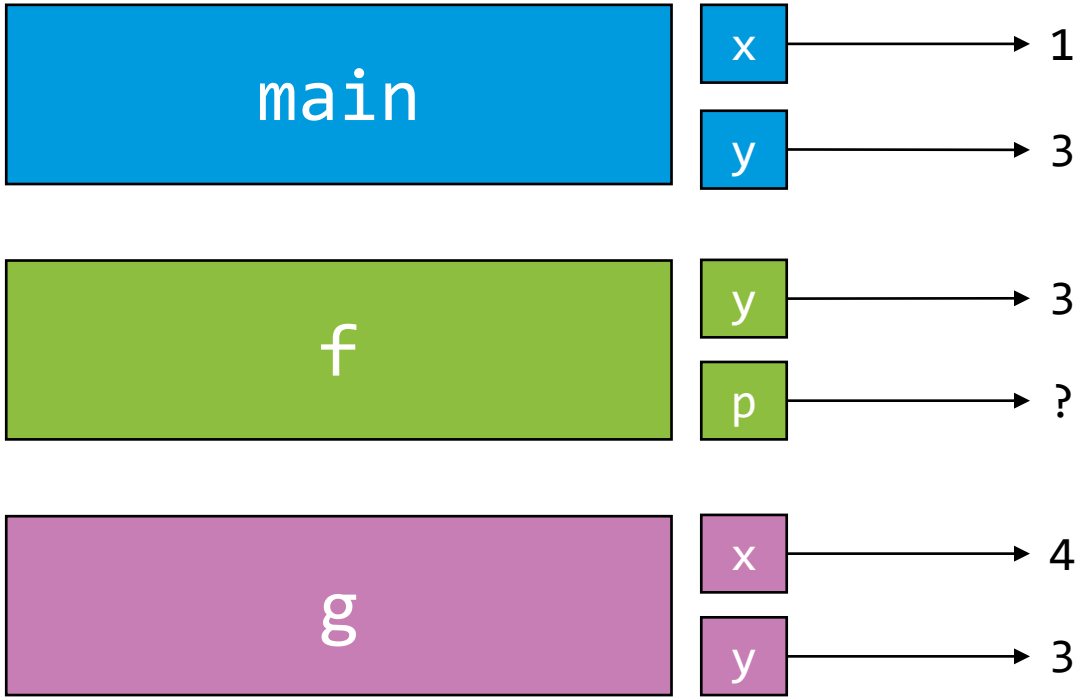
Advice for Learning C

- You **will** be slower learning C than Python – don’t be hard on yourself.
- Error messages can be confusing – focus on the first one.
- It’s easy to make subtle mistakes – write a bit of code, then test.
- Project-based IDEs (Eclipse, CLion, etc.) have an overwhelming amount of configuration – start with a simple editor.

Stack Diagrams

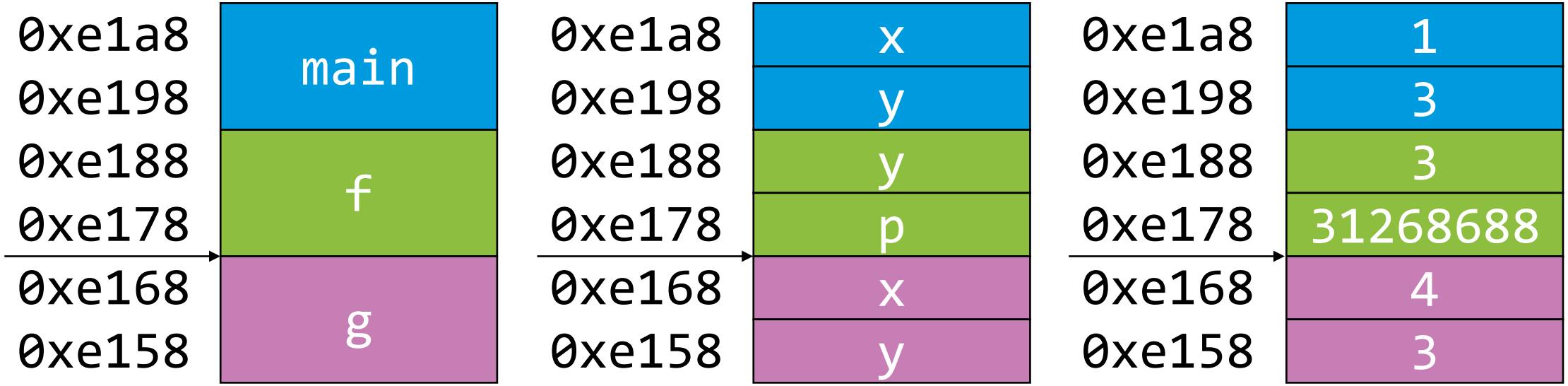
- You may remember stack diagrams in Python.
- They show you what functions are running and what each variable is.

```
def f(y):
    p = g(y, y)
    return p

def g(x, y):
    x += 1
    return x * y

def main():
    x = 1
    y = x + 2
    f(y)
```



Stack Diagrams

- In C, things look similar, but differ from Python in a few ways.



Stack Diagrams: Exercise

- Draw a stack diagram for the following code at the point indicated.

```
int add(int x, int y) {
  int z = x + y;
  return z;
}

void test_add() {
  int sum = add(3, 4);
  printf("%d\n", sum);
}
```

```
char *check_parity(int n) {
  switch (n % 2) {
    case 0:
      return "n is even";
    case 1:
      return "n is odd";
  }
}

void main() {
  test_add();
  char *s = check_parity(3);
  printf("%s\n", s);
}
```

Stack Diagrams: Exercise

- Draw a stack diagram for the following code at the point indicated.

```
int add(int x, int y) {
  int z = x + y;
  return z;
}
```

```
void test_add() {
  int sum = add(3, 4);
  printf("%d\n", sum);
}
```

```
char *check_parity(int n) {
  switch (n % 2) {
    case 0:
      return "n is even";
    case 1:
      return "n is odd";
  }
}
```

```
void main() {
  test_add();
  char *s = check_parity(3);
  printf("%s\n", s);
}
```

Stack Diagrams: Exercise

- Draw a stack diagram for the following code at the point indicated.