UNIVERSITY of WASHINGTON

LECTURE 1 · CSE 333 · Summer 2026

Intro to Systems Programming

Discussion: What are you excited for this summer?

Intro to the Course

COURSE DETAILS

Staff

Instructor:
Soham Pardeshi

Staff

Instructor:
Soham Pardeshi

Teaching Assistants:
Rishabh Jain
Tim Avilov
Keosha Chhajed
Jackson Tyler Lee Kent

Staff

Instructor:
Soham Pardeshi

Teaching Assistants:
Rishabh Jain
Tim Avilov
Keosha Chhajed
Jackson Tyler Lee Kent

Department Staff:
Undergraduate advisors
CSE Support
Facility managers
and many more…

Staff

Instructor:
Soham Pardeshi

Teaching Assistants:
Rishabh Jain
Tim Avilov
Keosha Chhajed
Jackson Tyler Lee Kent

Department Staff:
Undergraduate advisors
CSE Support
Facility managers
and many more…

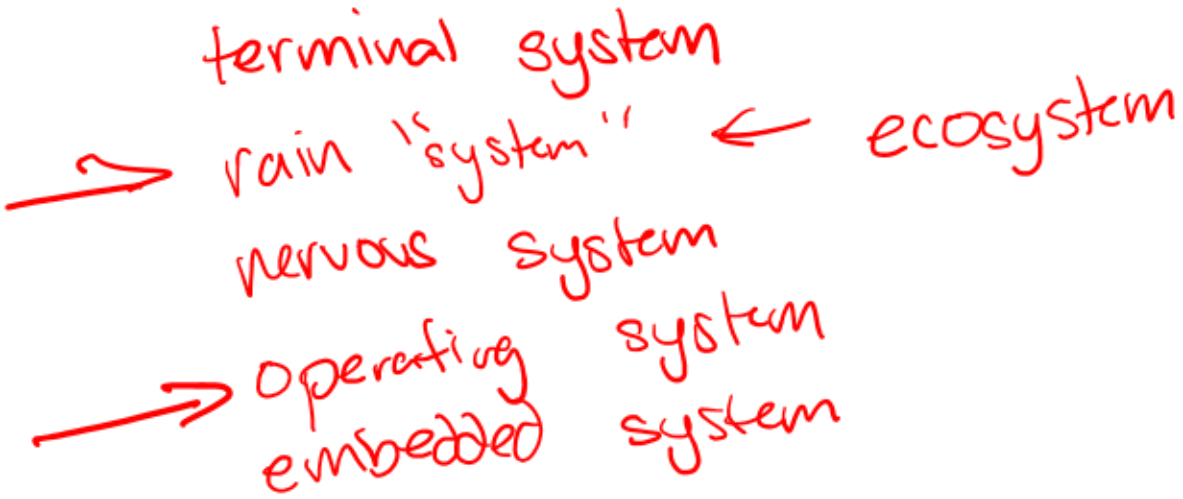
Our Goal:

Collectively, we want to ensure that you have a ***great*** experience in this course and that we prepare you to contribute to the global community of computer scientists and software people.

What is a system?

Discuss with the people around you.

- What is a system in general?
- What is a system in terms of computer science?
- What are some examples of systems?



What is a system?

Examples:

file system

operating system

version control system

database management system

build system

What is a system?

Examples:

file system

operating system

version control system

database management system

build system

mail system

What is a system?

- Has many components
- Has relationships between those components
- Provides an "abstraction" to users of the system

What is a system?

Examples:

file system

operating system

version control system

database management system

build system

What is a system?

- Has many components
- Has relationships between those components
- Provides an "abstraction" to users of the system

To build a system, we must be *very* thoughtful. Our abstraction must be simple, future-proof, and capable of handling anything a user may want to do. Typically, these systems are close to the hardware.

Pre-requisites

CSE 351:

- Basics of C
- Pointers
- Memory model
- System calls

CSE 123:

- Classes, inheritance, object-oriented programming
- Common data structures (stack, queue, hashmap)
- Good style

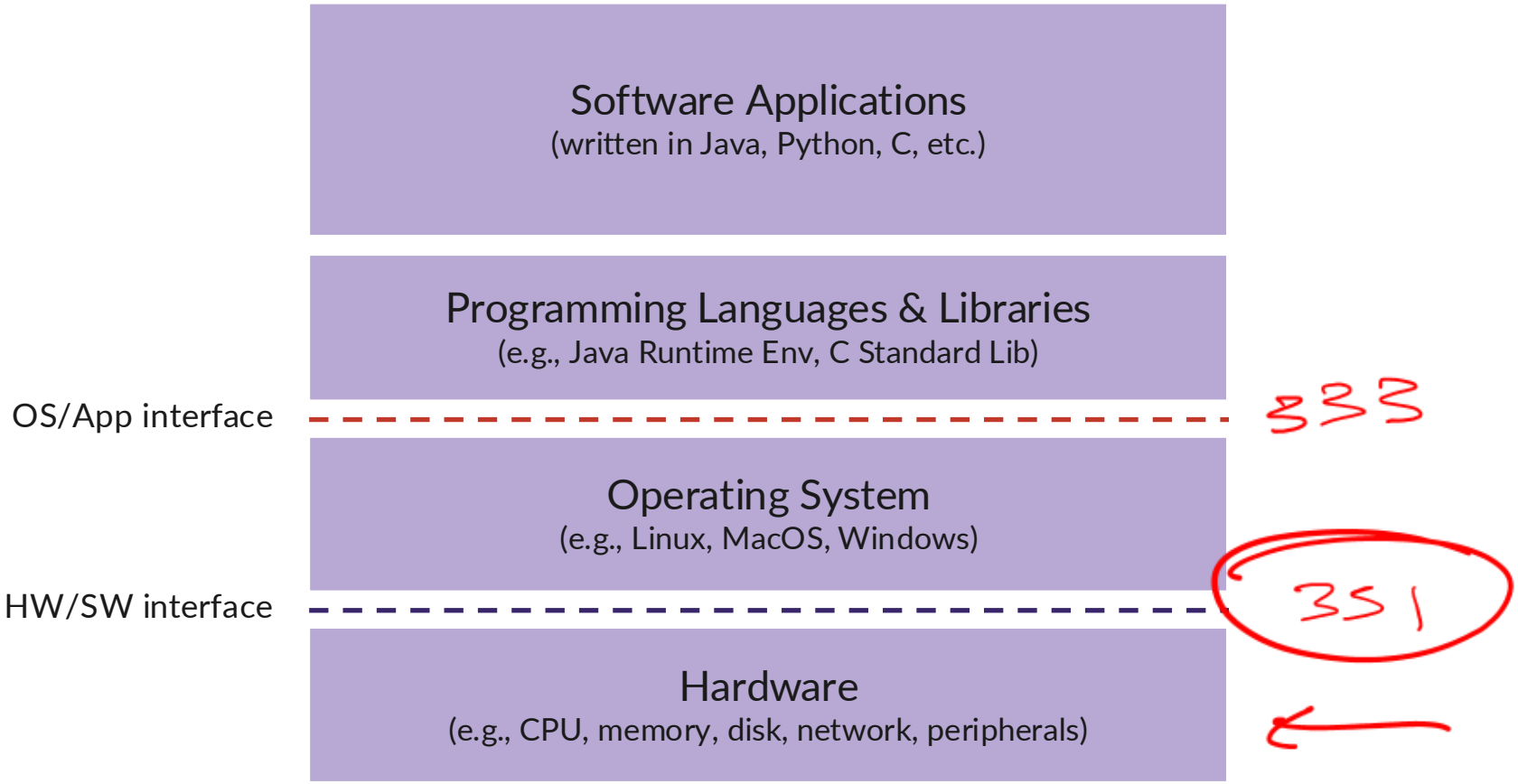
What are the goals of CSE 333?

- Think about system design

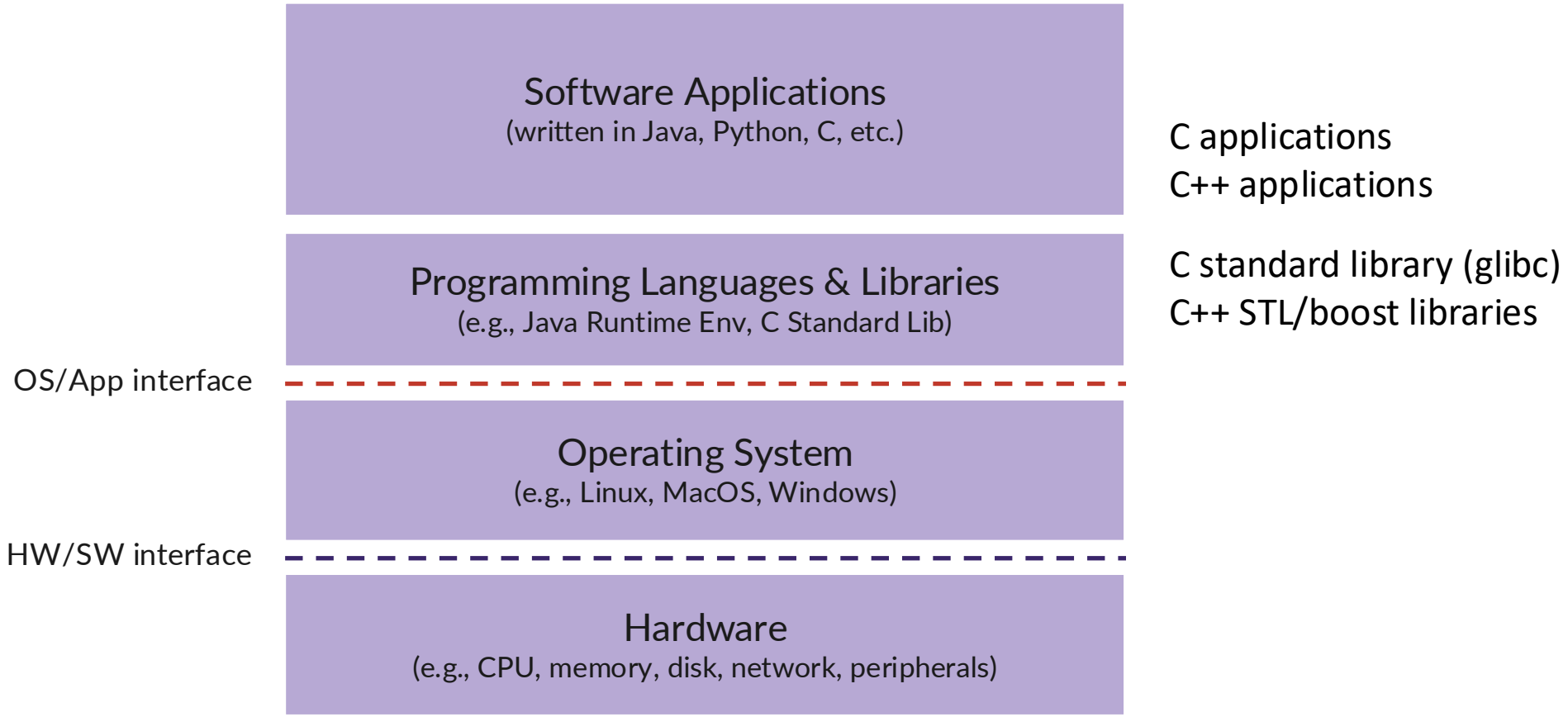
- Experience building low-level systems
  - Programming: C or C++
  - Concepts: Concurrency, OS interfaces, distributed systems algorithms
  - Practical: Build complex in-memory and on-disk data structures

- Build on top of low-level systems work, e.g.
  - operating system
  - file system
  - networking system

Goals of CSE 333



Goals of CSE 333



Assessments

- Homework (4)
  - project sequence that builds progressively more complex search engine
  - evaluated on correctness AND code quality
- Exercises (10)
  - stand-alone practice of lecture concepts
  - solutions released 24h after due date
- Midterm
  - during lecture, tentative. about halfway through the course
- Final
  - Tuesday, August 18

Grades

- Homework: 35%
- Exercises: 25%
- Midterm: 15%
- Final: 25%

Participation

You can earn up to 4 participation points each week by being engaged in the course

- up to 1pt from lecture participation
- up to 1pt from office hour attendance
- up to 1pt from using the Ed discussion board
- up to 1pt from other participation opportunities

Participation

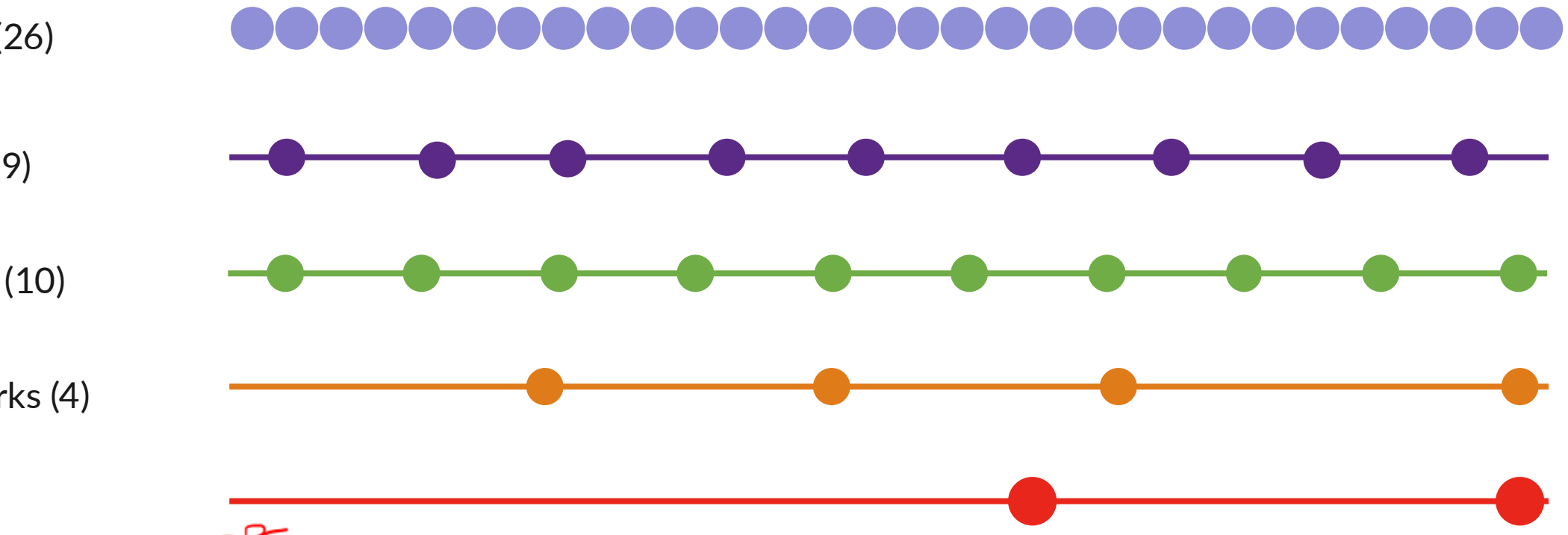
You can earn up to 4 participation points each week by being engaged in the course

- up to 1pt from lecture participation
- up to 1pt from office hour attendance
- up to 1pt from using the Ed discussion board
- up to 1pt from other participation opportunities

At the end of the quarter, if you have

- >= 5pts, three extra late days
- >= 10pts, drop your lowest exercise
- >= 15pts, earn a 10% bump on the midterm (or drop your worst questions)
- >= 20pts, earn a 10% bump on the final (or drop your worst questions)
- >= 25pts, earn a 10% bump on the homework

Overview



- **Midterm:** Mon, July ~~28~~ 27 in lecture
- **Final:** Tues, August 18

AI Usage

Don’t use it to cheat.

Good rule of thumb:

- It should never have access to the assignment spec or your code
- You should never copy-paste (or blindly copy) its output

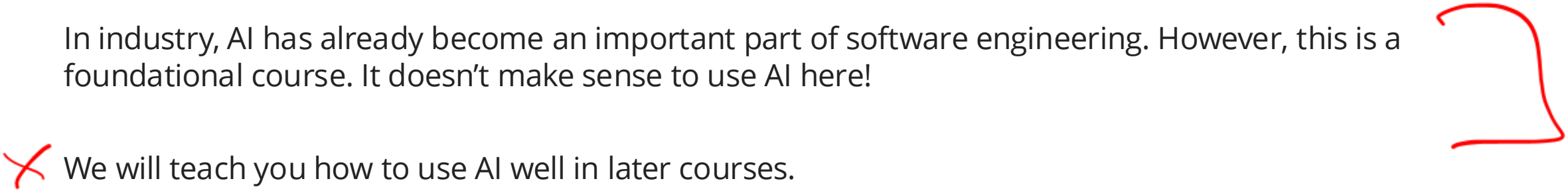
AI Usage

Just don't use it to cheat…

Good rule of thumb:

- It should never have access to the assignment spec or your code
- You should never copy-paste (or blindly copy) its output

In industry, AI has already become an important part of software engineering. However, this is a foundational course. It doesn't make sense to use AI here!



We will teach you how to use AI well in later courses.

Review of C

Differences from Java

- C doesn't support objects! But it does have … **structs**

Differences from Java (data structures)

- C doesn't support objects! But it does have … **structs**

```java
public class Point {
   private int x;
   private int y;

   public int getX() { return x; }
   public int getY() { return y; }
}
```

Java

```c
struct Point {
    int x;
    int y;
};
```

C

Differences from Java (data structures)

- C doesn't support objects! But it does have … **structs**

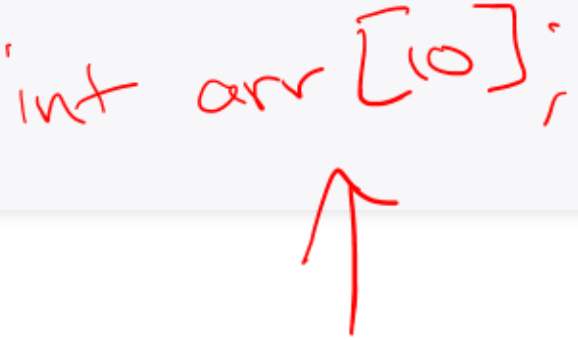
- Arrays are contiguous chunks of memory
  - **No implicit initialization** (not filled with zeros)
  - Don't know their own length, so **no bounds checking**

Differences from Java (data structures)

- C doesn't support objects! But it does have … **structs**

- Arrays are contiguous chunks of memory
  - **No implicit initialization** (not filled with zeros)
  - Don't know their own length, so **no bounds checking**


```
1  int arr[] = {25, 50, 75, 100};
2  arr[0] = 7;
3  arr[100] = 7; // no error
```



Differences from Java (strings)

- **C-strings** are just arrays of characters with a null-terminator

Differences from Java (strings)

- **C-strings** are just arrays of characters with a null-terminator

```
1  char greeting1[] = {'h', 'e', 'l', 'l', 'o', '\0'};
2  char greeting2[] = "hello";
```

Differences from Java (strings)

- **C-strings** are just arrays of characters with a null-terminator

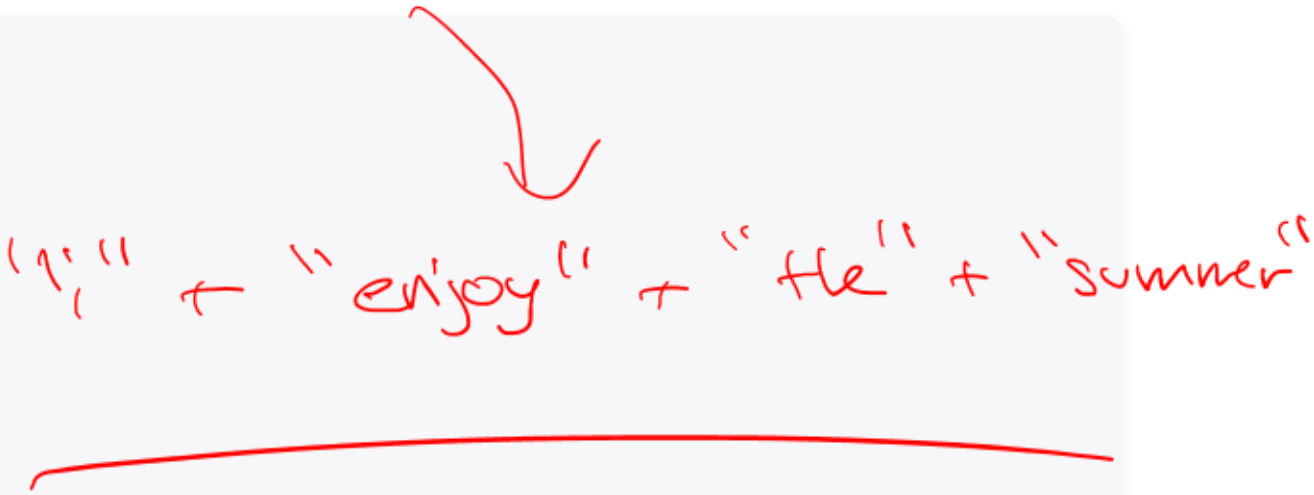
- Need to use standard libraries to "work with" strings
  - Documentation on these libraries is at https://cplusplus.com/reference/

Differences from Java (strings)

- **C-strings** are just arrays of characters with a null-terminator

- Need to use standard libraries to "work with" strings
  - Documentation on these libraries is at https://cplusplus.com/reference/

```
1 #include <string.h>
2
3 char str[80];
4 strcpy (str,"i");
5 strcat (str,"enjoy");
6 strcat (str,"the");
7 strcat (str,"summer");
```

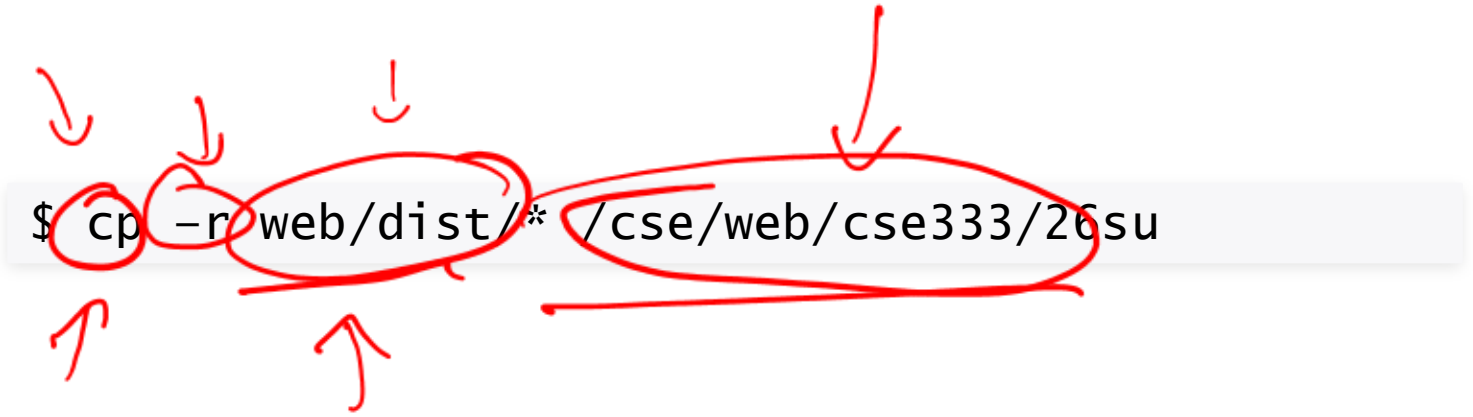


A typical program in C

```
#include <system_libs.h>
#include "local_files.h"

#define macro_name macro_expr

/* declare functions */
/* declare external variables & structs */

int main(int argc, char* argv[]) {
        /* core logic */
}

/* define functions that were declared above */
```

Command-line arguments

Command-line arguments



argc = 4

Reminders

General:

- Read the welcome email
- Check your access to Ed, Gradescope, Canvas, and GitLab repository (!)
- Setup your lab environment
  - CSE lab machine
  - SSH into attu (requires Husky OnNet VPN when you're off campus)
- CSE course websites are down (you can use this backup website)
- Notify us ASAP if you have conflicts with the midterm (7/27) or final exam (8/18)

Assessments:

- Pre-quarter survey due Wednesday at 11:59pm
- Exercise 1 due Thursday at 11:59pm

Questions?

Thanks for coming - see you next lecture!