LEC 11

CSE 121

Putting It All Together

Questions during Class?

Raise hand or send here

sli.do #cse121

BEFORE WE START

Talk to your neighbors:

What was the best part of your long weekend?

Music: ❄ CSE 121 26wi Lecture Tunes ❄

Instructor: Miya Natsuhara (feat. Matt Wang)

TAs:

Amogh	Hayden	Anum	Sam	Shayna
William	Aki	Abdul	Ethan	Jesse
Johnathan	Spencer	Janvi	Jessica	Minh
Anant	Savannah	Navya	Paul	Cayden
Reese	Tamsyn	Ruslana	Carson	

Announcements, Reminders

- Quiz 1 is tomorrow!
  - Quiz 1 reference sheet linked from Quiz 1 Practice Resources post
  - Email Miya *before* your section if you're not able to attend!
    - Do not come to section if you're sick! 😷
- Quiz 0 grades were released last week!
- P2 is out & due next **Tuesday, February 24th**
  - No new assignment this week
- R3 closes tomorrow
  - **P0**, C1, P1, C2 eligible
  - P0 cycling out of eligibility after R3!

… and things for this week

- Miya is at a conference this week!
  - Today: me (Matt) is guest lecturing
  - Friday: the *lovely* Hayden Feeney is guest lecturing
  - this week: **Miya's office hours are cancelled**

Today's Goals

1. How to write big programs
2. How to read specifications
3. How *everything* we've learned comes together

Skills that are helpful for your upcoming assignments (e.g. P2), for taking tests (e.g. Quiz 1), and for your future programming (TBD!).

Things I will ask you momentarily...

1. Which Scanner methods will help us solve this problem?
   - `next()`, `nextDouble()`, `nextInt()`, `nextLine()`
2. Which of these control structures will help us solve this problem?
   - for loops, while loops, nested loops, if statements
3. Which of these common patterns will help us solve this problem?
   - string iteration, cumulative algorithms, fencepost pattern

But first: spend a couple of minutes **reading through the SpotifyWrapped spec & expected output.**

Practice: Think

sli.do #cse121

Which Scanner methods will help us solve this problem?
(select all that apply)

A. `next()`

B. `nextDouble()`

C. `nextInt()`

D. `nextLine()`

Practice: Pair

sli.do #cse121

Which Scanner methods will help us solve this problem?
(select all that apply)

A. `next()`

B. `nextDouble()`

C. `nextInt()`

D. `nextLine()`

Searching for Scanners (1/2)

```
Welcome to 121 Wrapped!

Enter an artist (or 'end'): Alvvays
How many minutes did you listen to them? 21

Enter an artist (or 'end'): Laufey
How many minutes did you listen to them? 2000

Enter an artist (or 'end'): Geese
How many minutes did you listen to them? -5
How many minutes did you listen to them? 5

Enter an artist (or 'end'): end
```

single String token – next()

Searching for Scanners (2/2)

```
Welcome to 121 Wrapped!

Enter an artist (or 'end'): Alvvays
How many minutes did you listen to them? 21

Enter an artist (or 'end'): Laufey
How many minutes did you listen to them? 2000

Enter an artist (or 'end'): Geese
How many minutes did you listen to them? -5
How many minutes did you listen to them? 5

Enter an artist (or 'end'): end
```

single `int` token – `nextInt()`

Practice: Think

sli.do #cse121

Which of these control structures will help us solve this problem? (select all that apply)

A. for loops

B. while loops

C. nested loops

D. if statements

Practice: Pair

sli.do #cse121

Which of these control structures will help us solve this problem?
(select all that apply)

A. for loops

B. while loops

C. nested loops

D. if statements

Looking for Loops

```
Welcome to 121 Wrapped!

Enter an artist (or 'end'): Alvvays
How many minutes did you listen to them? 21

Enter an artist (or 'end'): Laufey
How many minutes did you listen to them? 2000

Enter an artist (or 'end'): Geese
How many minutes did you listen to them? -5
How many minutes did you listen to them? 5

Enter an artist (or 'end'): end

Your 121 Wrapped is here!!
You listened to 2022 minutes total this year.
Your top artist: Laufey with 2000 minutes.
You're a Laufey superfan!
```

Repeatedly entering an artist (and minutes) in a loop…

Q: do we know how many times we'll loop *before* we start?

No! "Indefinite Loop", while loop!

Q: *within* one iteration, are we doing any additional repeated actions?

No! No nesting!

But also: **Yes – we ask them the number of minutes repeatedly *until* it's nonnegative**

Identifying Ifs

[from spec]…

The `<fanType>` should be a different string depending on how many minutes they've listened to that artist:

- if they listened to the artist for under 60 minutes, say that they are a "budding fan"
- if they listened to the artist for 60-300 minutes, say that they are a "supporter"
- if they listened to the artist for 301-999 minutes, say that they are a "stan"
- if they listened to the artist for at least 1000 minutes, say that they are a "superfan"

The if statements are right there!!

Practice: Think

sli.do #cse121

Which of these common patterns will help us solve this problem?
(select all that apply)

A. string iteration

B. cumulative sum

C. fencepost pattern

Practice: Pair

sli.do #cse121

Which of these control structures will help us solve this problem? (select all that apply)

A. string iteration

B. cumulative sum

C. fencepost pattern

Cumulative Sum in Action

```
Welcome to 121 Wrapped!

Enter an artist (or 'end'): Alvvays
How many minutes did you listen to them? 21

Enter an artist (or 'end'): Laufey
How many minutes did you listen to them? 2000

Enter an artist (or 'end'): Geese
How many minutes did you listen to them? -5
How many minutes did you listen to them? 5

Enter an artist (or 'end'): end

Your 121 Wrapped is here!!
You listened to 2026 minutes total this year.
Your top artist: Laufey with 2000 minutes.
You're a Laufey superfan!
```

total minutes: 0

total minutes: 0 + 21 = 21

total minutes: 21 + 2000 = 2021

total minutes: 2021 + 5 = 2026

total minutes: 2026

Cumulative *Max* in Action

```
Welcome to 121 Wrapped!

Enter an artist (or 'end'): Alvvays
How many minutes did you listen to them? 21

Enter an artist (or 'end'): Laufey
How many minutes did you listen to them? 2000

Enter an artist (or 'end'): Geese
How many minutes did you listen to them? -5
How many minutes did you listen to them? 5

Enter an artist (or 'end'): end

Your 121 Wrapped is here!!
You listened to 2026 minutes total this year.
Your top artist: Laufey with 2000 minutes.
You're a Laufey superfan!
```

max minutes: 0

max minutes: max(0, 21) = 21

max minutes: max(21, 2000) = 2000

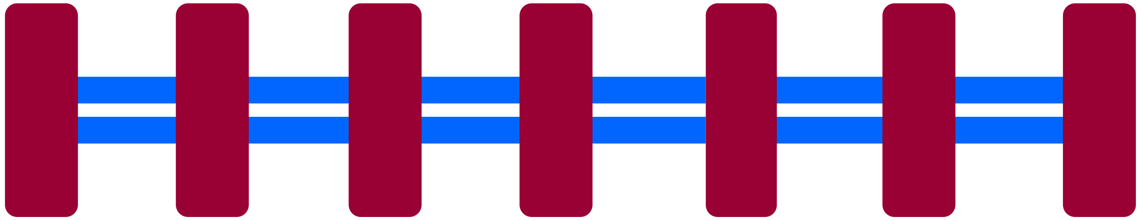
max minutes: max(2000, 5) = 2000

max minutes: 2000

Reminder (again): Fencepost Pattern

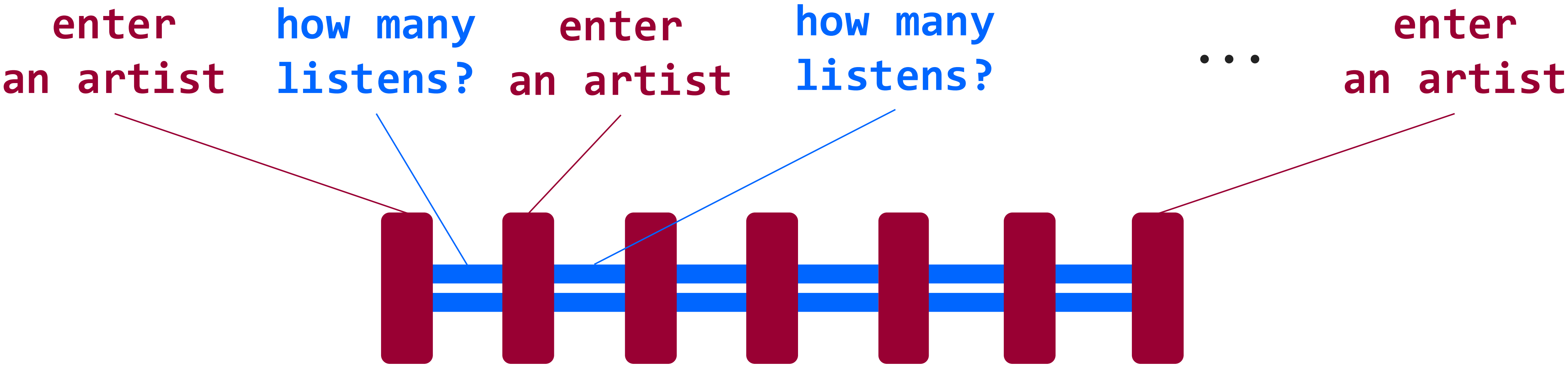
Some task where one piece is repeated *n* times, and another piece is repeated *n-1* times and they alternate

```
h-u-s-k-i-e-s
```



Fencepost Pattern … for User Input?

Some task where one piece is repeated *n* times, and another piece is repeated *n-1* times and they alternate



Fencepost Pattern in Action

```
Welcome to 121 Wrapped!

Enter an artist (or 'end'): Alvvays
How many minutes did you listen to them? 21

Enter an artist (or 'end'): Laufey
How many minutes did you listen to them? 2000

Enter an artist (or 'end'): Geese
How many minutes did you listen to them? -5
How many minutes did you listen to them? 5

Enter an artist (or 'end'): end

Your 121 Wrapped is here!!
You listened to 2026 minutes total this year.
Your top artist: Laufey with 2000 minutes.
You're a Laufey superfan!
```

enter an artist
how many listens?

enter an artist
how many listens?

enter an artist
how many listens?

enter an artist

Tackling Big Programs

Two big pieces of advice:

1. Start small
   - can you do a simpler version of the problem?
   - often: ignoring a part of the problem, *for now*

2. Test *frequently*
   - make sure that your code works *before* moving on
   - **don’t only test at the end!** (very hard to debug!!)