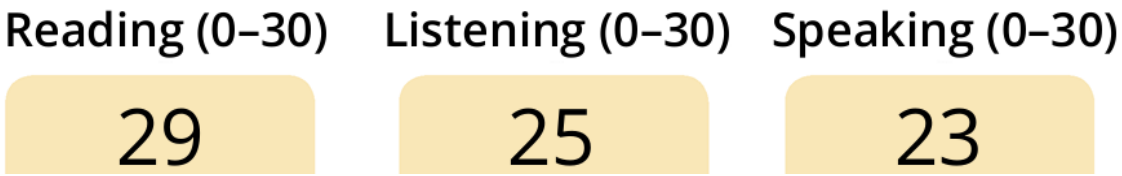
CS010C

Lab1

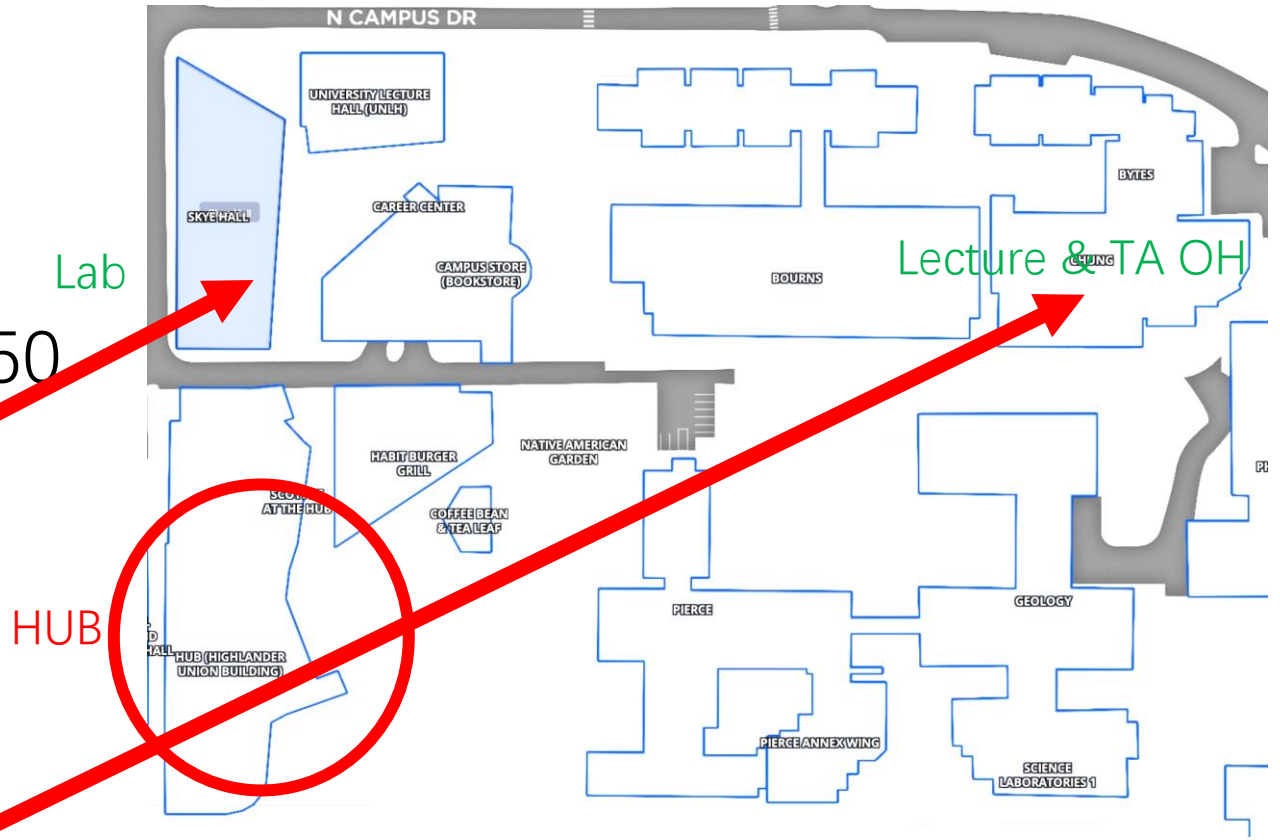
Introduction

- Yue Zhu (“Yoo-eh Joo”)
  - “Yue” is too short so sometimes I don’t notice if someone is calling me
  - Feel free to call me by my nickname “ZY”(or fullname ZhuYue, or YueZhu)
  - Email: yzhu303@ucr.edu
  - Slack: DM / channel + @
- To be honest, my English…
  - Even worse?
  - If you're not getting it, it's not your fault
- Just
  - simply ask me again
  - wait until after class
  - feel free to send me a message or email anytime



Information

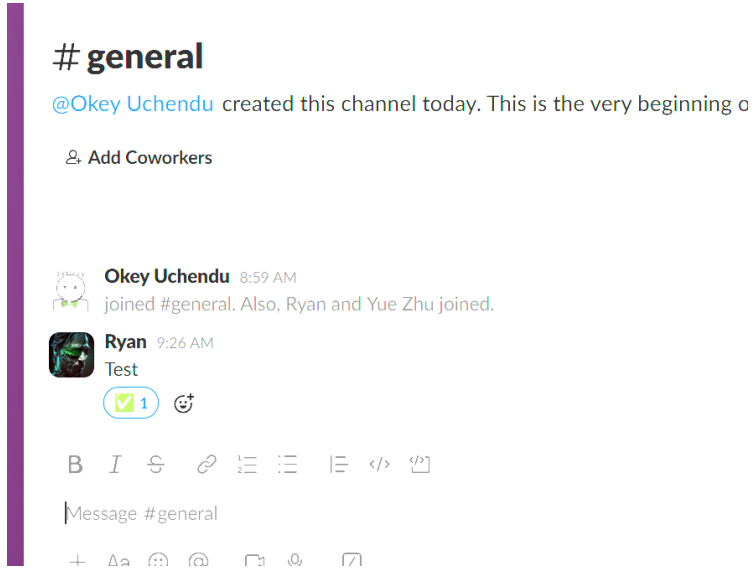
- Tuesday & Thursday 10:00-12:50
- Lab: Skye Hall | Room 171
- Attendance: 10:10 & 12:00
  - Attendance sheet
  - $10 * 0.5 = 5pt$
- Recording
- TA Office Hours(3h)
  - In-person at WCH 459(2h?) & ZOOM(1h?)
  - Thursday 2-5pm? Friday 2-5pm?



Test Student	√	√	ZY	Y	Y

Preparation

- Slack channel
  - Canvas -> CS010C -> Slack -> Join -> “general”
- Zybook labs
  - https://learn.zybooks.com/zybook/UCRCS010CRusichSummerSessionB2025
- AWS labs(TBD)
- ~~Code Assist?~~



Review and Hints

If you’ve mastered the material from class, you can start now — no need to follow me!

Lab 1

Each use of `std::someFunction` is syntactically correct, theoretically, there's no need to use using namespace std; anymore. But unexpectedly fails the test in this case!

- **Bug**: `using namespace std;` **is necessary!**

- main.cpp - main() function
  - Complex!(~200 lines)
  - May take several attempts to debug successfully.
- Playlist.h - Class declaration
  - Very simple (~20 lines)
- Playlist.cpp - Class definition
  - Simple (~50 lines)

- Local debugging
  - Compile
    - g++ main.cpp Playlist.cpp -Wall -o a.out
  - Run
    - ./a.out

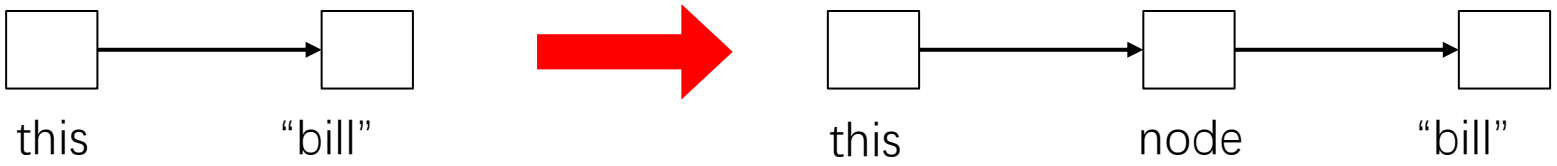
Playlist.h

- Declaration
  - `someType FunctionName(Some Parameters);`

- public & private member
- constructor
  - Default constructor: {"none", "none", "none", 0,0 *or nullptr*}
    - Neither “” nor nullptr!
  - Parameterized constructor
- mutator & accessor
  - Setxxx: `void SetXXX(some Parameters);`
  - Getxxx: `someType GetXXX() const;`

Playlist.cpp

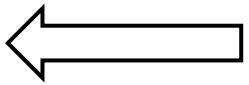
- Definition
  - `someType FunctionName(Some Parameters){xxx}`
- `void PlaylistNode::InsertAfter(PlaylistNode *anothernode)`
  - Insert “anothernode” behind “this” node
  - Class member function example
    - someObject.InsertAfter(someObjectElse)
    - [This] is the pointer pointing to “someObject”



main.cpp

- Framework
- How to handle interactive menus?

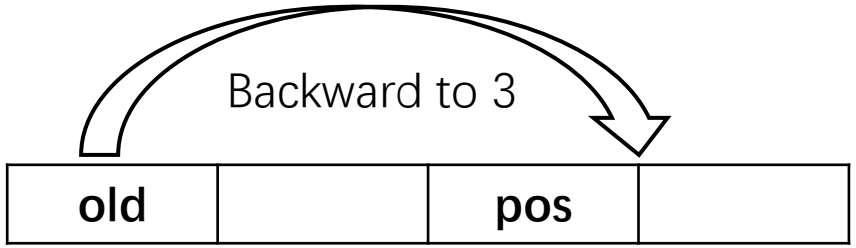
```
while (true)
{
    …
    char choice = PrintMenu(title);
    switch (choice)
    {
        case '?':
            do_something(parameter);
            break;
        case '?':
            …
    }
}
```

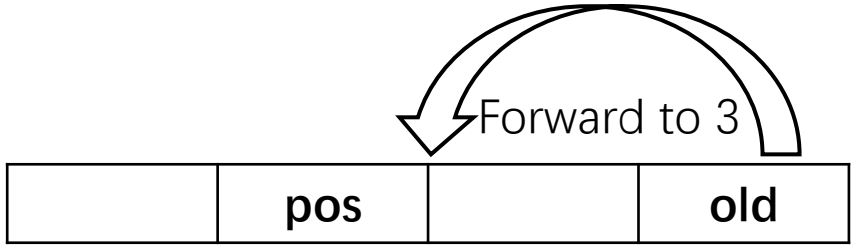


Just 1 implementation,
Not necessary to be the same

```
  6 > char PrintMenu(std::string &t
 22 > void Output_full_playlist(stc
 39 > void Add_song(PlaylistNode *&
 73 > void Remove_song(PlaylistNode
106 > void Change_position_of_song(
150 > void Output_songs_by_artist(P
172 > void Output_total_time(Playli
184
185 > int main()…
```

Change position of song





- Solution 1
  - Find the corresponding position to insert
    - Changing forward: $k-1$
    - Changing backward: k
  - move the node directly
- Solution 2 (more intuitive)
  - Extract(remove) the node
  - then find the corresponding position to insert: $k-1$
- corner case: head
  - new position is 1
  - old position is 1

I/O Hints

- Output: std::endl or "\n" or '\n'

```
while (true)
{
    std::cout << "\n";
    char choice = PrintMenu(title);
    switch (choice)
        …
}
```

- Input: when it encounters a blank space
  - cin stops reading
  - getline is more suitable for reading a whole line that may contain spaces
    - `std::getline(std::cin, something);`
  - Bug? It's newline character when using getline after cin
    - Solution1: Ignore that character
      - `std::cin >> something;`
      - `std::cin.ignore();  // Clear the newline character from the input buffer`
      - `std::getline(std::cin, somethingelse);`
    - Solution2: Or read that whitespace first, then get the remain input
      - `std::cin >> something;`
      - `std::getline(std::cin >> std::ws, somethingelse);`
  - Solution3: Or just always use get line
    - `std::getline(std::cin, something);`

```
a
SD567
I Got The News
Steely Dan
```