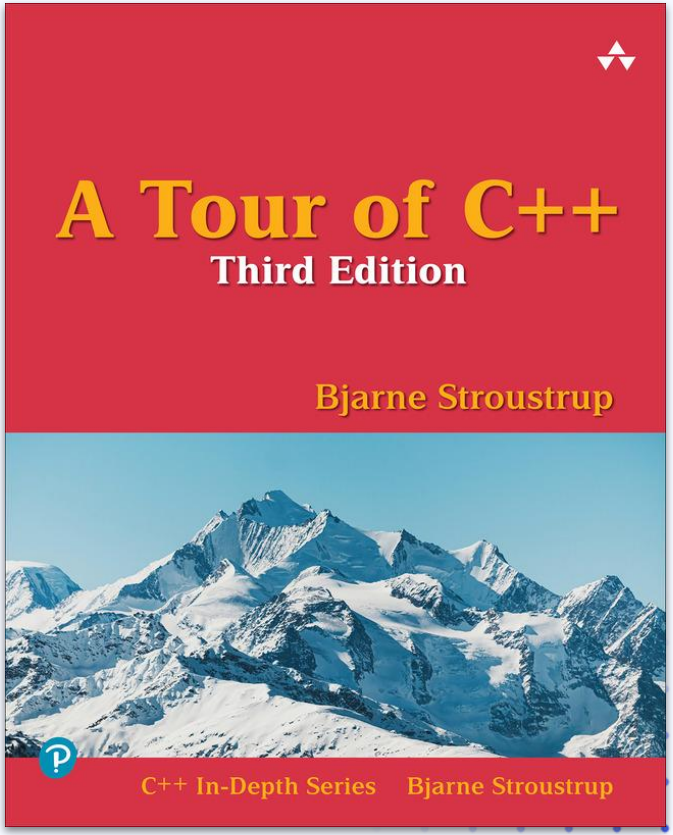
# Separate Compilation

CSE 232 - Dr. Josh Nahum

# Reading:

Sections 3.1, 3.2, and 3.3



# Table of contents

# OO

# Modules

Unfortunately most compilers don't support modules yet.


You won't be expected to understand or use them this semester.

# 01

# Header Guards

# Problem: Re-definitions

In C++, a function and a class can only have **ONE** definition in a program. If you have multiple definitions for the same function/class, the compiler will give you a "Re-definition" error at compile time.

However, since the same header file can be used in multiple places in the same program (like "`#include <iostream>`" in multiple implementation files), this means that contents of the file will be duplicated, unless we do something!

# Method One: C Macros

```
// header.h
#ifndef UNIQUE_NAME
#define UNIQUE_NAME
// File contents
#endif
```

# Method Two: #pragma once

```
// header.h
#pragma once
// File contents
```

Pros of this method:

- Simpler
- Don't need a unique name for each header file

Cons of this method:

- Not part of the C++ Standard
  - However, basically every compiler supports this method regardless

Advice: use this method with every header file

# 03
# Multi File Compilation

# Only compile implementation files!

Files:

- **main.cpp** // has the main function
- student.h // has a class declaration
- **student.cpp** // has method definitions
- gui.h // library that enables windowing
- stats.h // various useful function declarations
- **stats.cpp** // function definitions

g++ -Wall -std=c++20 **main.cpp student.cpp stats.cpp**

# Only include header files!

Files:

- **main.cpp** // has the main function
- student.h // has a class declaration
- **student.cpp** // has method definitions
- gui.h // library that enables windowing
- stats.h // various useful function declarations
- **stats.cpp** // function definitions

Implementation files (e.g. **main.cpp student.cpp stats.cpp**) should never be included in any file.

# Using Headers

## Standard Library

#include <vector>

## File

#include "student.h"

# Never use using in a header!

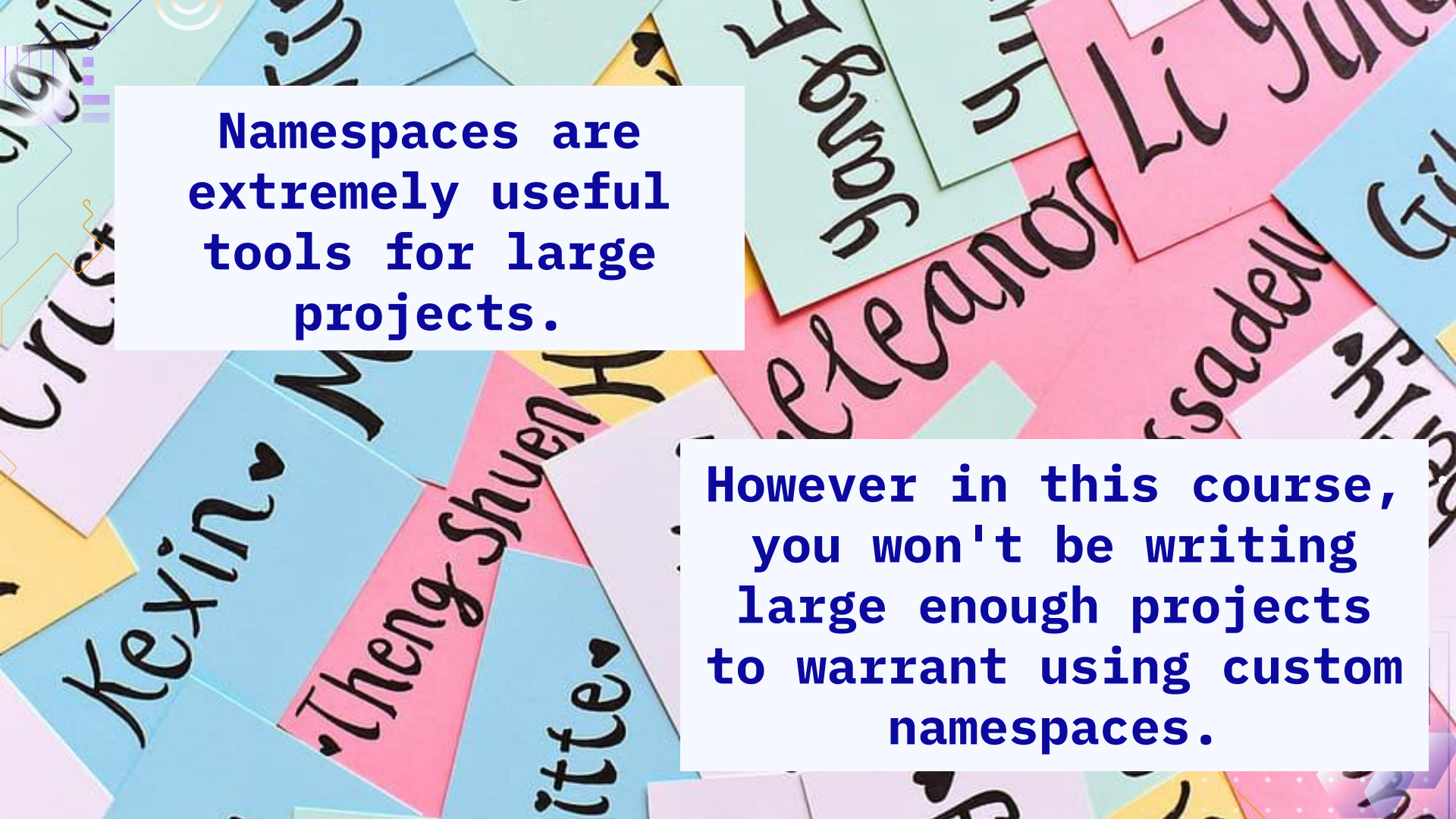
**Example using declarations:**

- `using namespace std;`
- `using std::cout;`

**Never put these in header files as they will pollute the namespace of any file that includes the header.**

# 03

# Namespaces


Namespaces are extremely useful tools for large projects.
However in this course, you won't be writing large enough projects to warrant using custom namespaces.

# Attribution

## Please ask questions via Piazza

Dr. Joshua Nahum
www.nahum.us
EB 3504



**CREDITS:** This presentation template was created by **Slidesgo**, and includes icons by **Flaticon**, and infographics & images by **Freepik**