The slide features a light blue background with abstract circuit-like patterns in purple and orange. These patterns include lines, dots, and geometric shapes that resemble electronic components and wiring, scattered across the top, bottom, and right sides of the slide.

Separate Compilation

CSE 232 – Dr. Josh Nahum

Reading:

Sections 3.1, 3.2, and 3.3

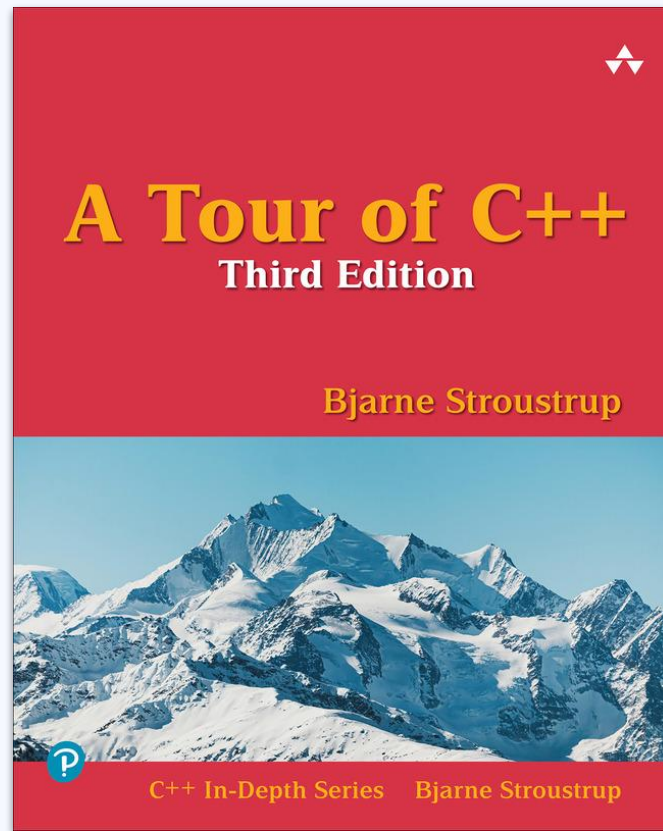




Table of contents

00

Modules

01

Header Guards

02

Multi File Compilation

03

Namespaces






00

Modules



The background is a vibrant yellow with a complex network of black lines and dots, resembling a molecular structure or a data network. The lines are thin and black, connecting various black dots of different sizes. Some dots are larger and more prominent, while others are smaller. The overall effect is a dense, interconnected web of points and lines, creating a sense of complexity and connectivity. In the top left corner, there is a small, stylized graphic of a white circle with a black outline, and a small, colorful, abstract shape. In the bottom right corner, there is a small, stylized graphic of a white circle with a black outline, and a small, colorful, abstract shape.

**Unfortunately
most compilers
don't support
modules yet.**

**“
COMING
SOON
”**

**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



The background of the slide is a collage of various colored sticky notes (pink, blue, yellow, green) scattered across the surface. Each note has a name written on it in a black, cursive script. Some of the visible names include 'Kexin', 'Theng Shuen', 'Eleanor', 'Li', 'Sadell', and 'Hing'.

**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](#)

© Michigan State University – CSE 232 – Introduction to Programming II