

# C++: Header Files

## Header File - Overview

Break up large files, Speeds compilation process

Organizes code

Separates interface from implementation  
(and reduces your need to know what goes on 'under the hood')

But adds slight complexity

# Header File Rules –

## 1. YOU MUST USE INCLUDE GUARDS

```
//a.h
const int myInt=3;
//main.h
#include "a.h" //define myInt here
#include "a.h" //attempt to redefine
               //error C2370
```

No include guards you  
get multiply defined  
symbols

**Instead wrap in an include guard**

```
//a.h
#ifndef MY_UNIQUEID //if not included yet
#define MY_UNIQUEID //then define this symbol
                   //and include the const def
                   //next time included,
                   //MY_UNIQUEID defined
                   //so const def not included
const int myInt=3;
#endif
```

VC++ uses

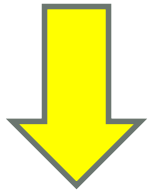
```
//a.h
#pragma once //only once
const int myInt=3;
```

**Upshot:: ALWAYS USE INCLUDE GUARDS ON HEADERS**

# Header File Rules – Just declarations no definitions

declaration

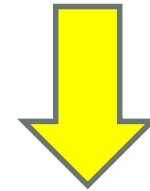
In .h file



```
int a2();
```

definition

In .cpp file



```
int a2(){  
    return 2;  
}
```

# Header File Rules – minimal exposure

## In .h file

Only include those files necessary to make header self contained (no compiler errors).

```
#pragma once
//B function definitions
#include <string>

std::string b1();
std::string b2();
std::string b3();
```

## In .cpp file

All other includes

# Header File General Rules

- `<>` for system header files
- `"` for your header files
- Only const variables (unless part of a class)
- Header file should contain only related stuff
- Never include a .cpp or source file
- Never put a “using namespace ...” declaration in a header file (forces anyone including your header to also use that namespace)
- General strive for complete AND minimal (only what's necessary)

# Includes should be sorted and grouped

```
#include <fstream>  
#include <iomanip>
```



System header  
files first

```
#include "com/company/ui/PropertiesDialog.h"  
#include "com/company/ui/MainWindow.h"
```



Then user  
header files

Include statements go at top of file only

# Header Files – Location (eclipse)

- Big projects – Organization is key
- Source in one dir, Headers in another
  - Use relative paths (ex. `#include "../includes_usr/constants.h"`)
  - Or let IDE find headers by specifying which directories to search

