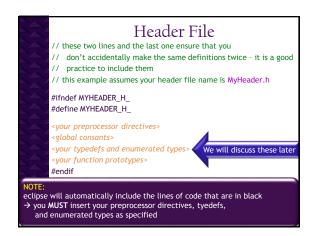


Header files • So far we've worked with several header • files that follow #include <iostream> <iomanip> <fstream> <string> • We include these to be able to access certain predefined functions, classes, or variables in C++ Creating our own o It is often convenient to create your own header files • To do this we need to create the file • Include it in our source code • Creating the file • create a new file filename.h end it with .h o Including the file • #include "filename.h"



Example: Creating a header file // this file is called myheader.h #ifndef MYHEADER_H_ #define MYHEADER_H_ // preprocessor directives go here #include <iostream> #include <iomanip> #include <string> using namespace std; // Global Constants // User Defined Types go here (more on this later) // Prototypes go here int SearchStArray(string stAr[], string searchStr); #endif /* MYHEADER_H_*/ • To include this file #include "MyHeader.h"

Some points to mention • you must use quotes in your header file • "MyHeader.h" → NOT <MyHeader.h> • the file must be located in your project folder • otherwise C++ can't find it

How to include source code • Create the file as a source file (.cpp)

- add your code (make sure you use the headers that you need)
- put the source file in the same folder as your main.cpp
- Don't #include for source code

nic 21 - Header fil

Common Errors

- o Make sure your files are all in the same folder
- Make sure that you have your preprocessor directives BEFORE your prototypes
 - ORDER MATTERS
 - ■1 preprocessor directives
 - # includes & namespace
 - ■2 global constants
 - ■3 typedefs and enumerated types
 - 4 prototypes
- You can't have code in the header file
- You can have code in a separate file
- You can only have 1 int main()

c 21 - Header fil

Good Practices

- Keep related functions in the same files
 - e.g. I/O
- Separating your files makes them easier to manage
 - your main.cpp can get long and difficult to find things

- 24 Use des files

Some Notes on Functions

- $\ensuremath{\mathbf{o}}$ Keep them simple and try to make them generic
 - \rightarrow that way you can reuse them

Example:

float AverageTwoInts (int int1, int int2)

Instead of

float AverageTwoAges (int age1, int age2)

- Keep them Simple!
 - each function should do 1 thing
 - In otherwords → if you need to search for something your function should just search for that something not deal with I/O specific to your project

c 21 - Header files.

10