

# Ch14-HeaderFiles

November 4, 2020

## 1 Header Files

- header files and their purpose
- standard library header files
- header file content rules
- using header files
- implementation files

### 1.1 HeadeF Files and their purposes

- we've used library header files such as `iostream`, `string`, etc.
- the purpose of this chapter is to learn how to create your own header files and why
- as program grows bigger, code need to be divided into many files
- using function prototype (forward declaration) and defining functions after main is not scalable
- breaking solution code into many files has many advantages:
  - makes program easier to manage, read, update, debug
  - makes it easier to work in a team where each member works on a separate file
  - avoid conflicts while using version control such as git
- Generally, header files allow us to put declarations in one or more files and then include them wherever we need them
  - this can save a lot of typing in multi-file programs
  - helps us create our own library of important functions

### 1.2 Creating header files

- header best practices:
  - header include guards (to avoid being included multiple times in the final binary)
  - contains struct and class definitions
  - contains function prototypes
  - avoid function definitions
  - may include other header files
  - do NOT define global variables
  - each header file should be as independent as possible with specific purpose
  - file has `.h` extension

### 1.3 Creating implementation files

- typically a header file are implmented in a separate corresponding `.cpp` file

- functions and classes are implemented or defined in implementation files
- implementation file are regular .cpp files that must include the header being included
- must also include any library header files required to implment the functions
- syntax:

```
#include "headerFileName.h"
```

## 1.4 Using header files

- include user-defined header files similar to library header file
- include only the header files that
- syntax:

```
#include "headerFileName.h"
```

- use the functions and user-defined types, etc. defined in the included header file

### 1.4.1 Demos - see folder [demo\\_programgs/Ch14/](#)

[ ]: