


# Today - Lecture 9 - CS162

1) Continue with the class construct  
creating a "List" class for our show list program.

- review concepts
- examine the design
- implement another class

## Announcements:

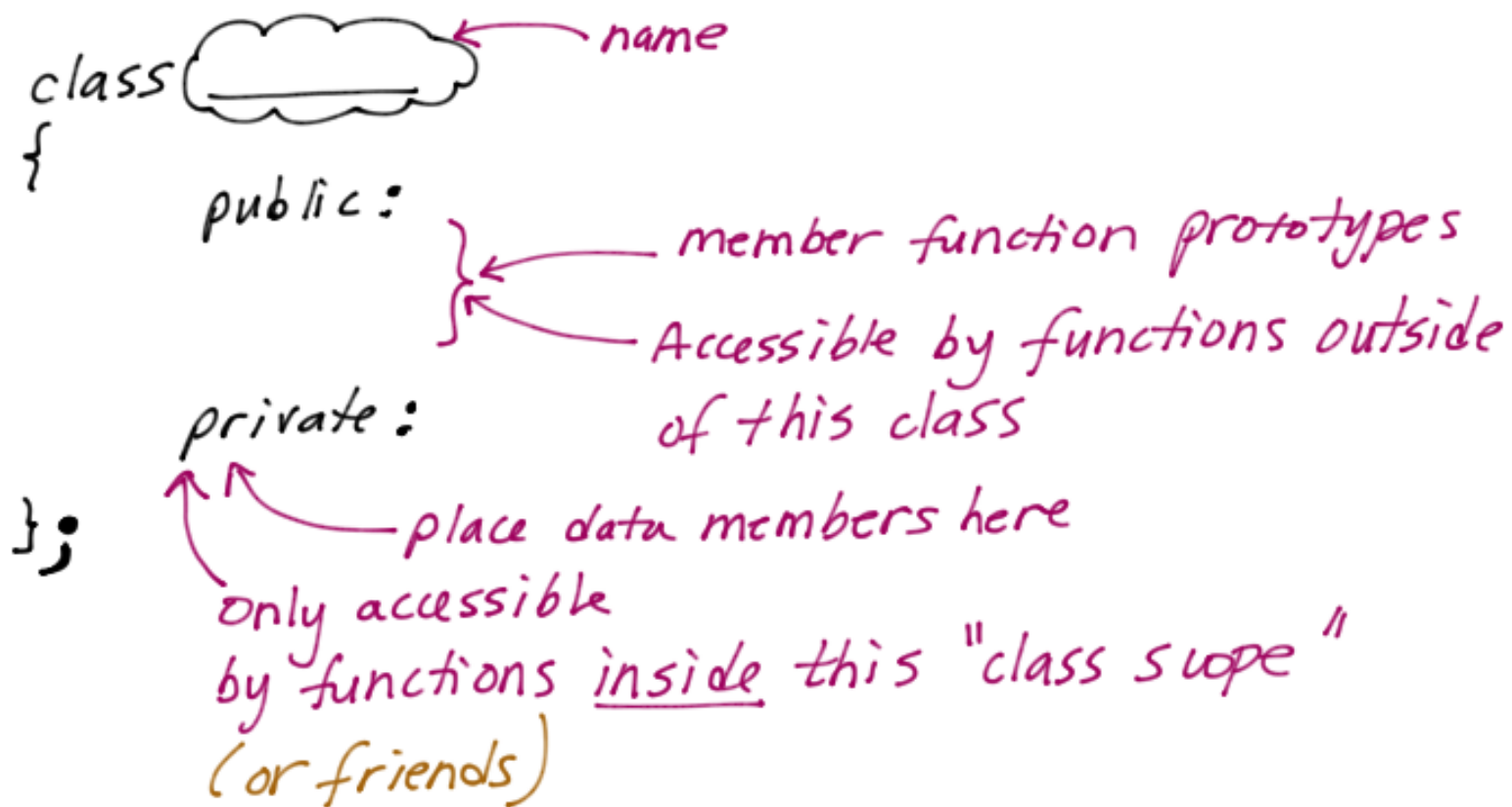
- Exams are continuing all week
- The next quiz (on classes) is active. It ends  
Monday Feb 20<sup>th</sup> at 11:59pm
- Homework 3 due date: Feb 23<sup>rd</sup>  ← changed!

# Class Construct

class  $\longrightarrow$  data type

object  $\longrightarrow$  variable, instance of a data type

class interface  $\longrightarrow$  where we declare functions (function prototypes) and specify the data that will be available for all objects of this class.

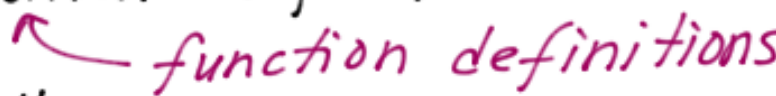
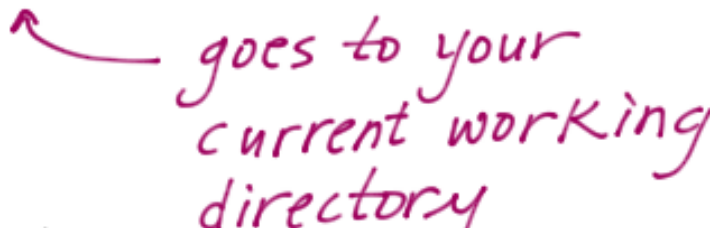


# Multiple Files

## •h file (declarations)

- 1) #includes
- 2) constants
- 3) structs also prototypes
- 4) class interfaces
- 5) DO NOT implement the "body" of functions in the •h file
- 6) DO NOT #include any •cpp file

## •cpp files (implementation file)

- 1) #include " ~•h "  function definitions
- 2) Function bodies  goes to your current working directory
- 3) There can be only 1 main function in all of the .cpp files put together

On unix, compile via:

```
g++ main.cpp video.cpp
```

or

```
g++ *.cpp
```

this works if all of the functions in your directory are part of this "project"

To use the gdb or ddd debuggers, compile with the `-g` option

```
g++ -g *.cpp
```

When implementing member functions

1) In the .cpp file **ALL** prototypes listed in the class interface (.h) **MUST** be implemented

2) Precede function name with the class name and the scope resolution operator (::)

↓  
**video::**video()  
{

// body of the function

}

↓  
void **video::**display()  
{

// body of the function

}

Now... using classes for the list of shows:

1) Design the solution by thinking about the data and thinking about what operations make sense working on that data — **GROUPING IT TOGETHER**

