EME 152 Discussion 7

November 10, 2021

Agenda

- C++ classes
  - Public vs. private members
  - Declaring member functions
- File system
  - Read
  - Write
- QuickAnimation
  - Format
  - Usage in Ch

Classes

- A class is a data structure in Ch/C++ programs
- Classes are an extension of the C “struct” (structure)
- Ch/C++ classes also support:
    - Public and private members
    - Member functions
- CPlot is an example of a Ch class.
    - `data2d()` is an example of a CPlot member function.

Source: C for Engineers and Scientists

Classes - Access Modifiers

- Like C structs, C++ classes may have members (variables, functions, etc.) However, C++ members may be declared public or private.
    - Public members may be accessed just like C struct members by any function or class.
    - Private members may only be accessed by that class’s own member functions.
    - “Public” and “private” are called access modifiers.

Source: C for Engineers and Scientists

Classes - Member Function

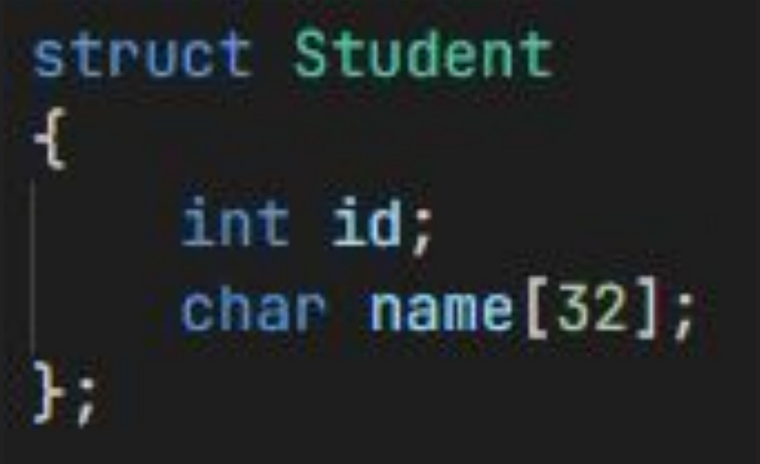
- A member function is a function that resides within a class. Member functions have access to all members within a class, including private members.
- To create a class member function, perform the following steps:
  - Declare the function within the body of the class declaration.
  - Define the function using the scope resolution operator, '::'.

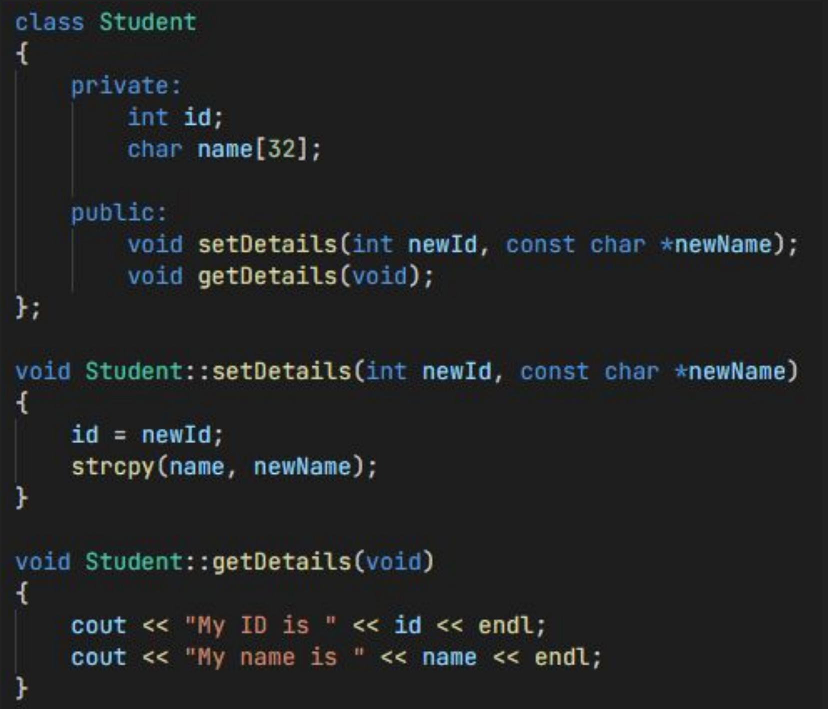
Source: C for Engineers and Scientists

Classes - Comparison

Right: C++ class with member functions.

Bottom: The same structure in C.


```
struct Student
{
        int id;
        char name[32];
};
```

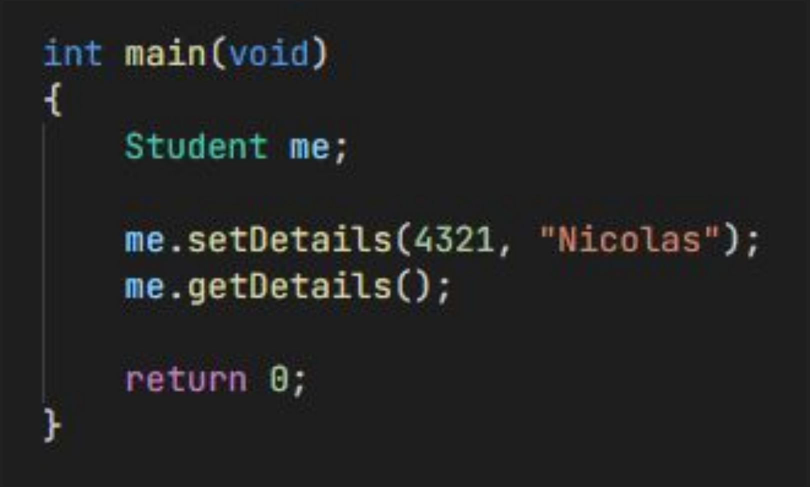

```
class Student
{
    private:
        int id;
        char name[32];

    public:
        void setDetails(int newId, const char *newName);
        void getDetails(void);
};

void Student::setDetails(int newId, const char *newName)
{
    id = newId;
    strcpy(name, newName);
}

void Student::getDetails(void)
{
    cout << "My ID is " << id << endl;
    cout << "My name is " << name << endl;
}
```

Classes - Usage



```
int main(void)
{
    Student me;

    me.setDetails(4321, "Nicolas");
    me.getDetails();

    return 0;
}
```

Using a third party class in Ch

- If/when you use third party C++ packages, typically you will only have access to the header files: This means you will only have access to the public member variables and functions.
- The header files along with documentation should provide the programmer with enough information to use the provided classes. The actual implementation is hidden from the programmer.

Source: C for Engineers and Scientists

File System

- In C, file manipulation is done via the functions `fopen()`, `fclose()`, and file pointers. Other useful functions are `fprintf()`, `fscanf()`, and `feof()`.
- `fprintf()` and `fscanf()` are analogues of `printf()` and `scanf()`. `feof()` is used to detect if a file pointer is pointing to the end of a file.

Source: C for Engineers and Scientists

File System

`fopen(filename, mode);`	Open a file and return the input stream.
`fclose(stream);`	Close the file. (End the stream.)
`fprintf(stream, format, …);`	Write formatted data to the stream.
`fscanf(stream, format, …);`	Read formatted data from the stream.
`feof(stream);`	Check if the end of file is reached.

What is QuickAnimation

$$\begin{tikzcd}
\text{Quick Animation file} \arrow[d, "\text{input}"'] \\
\text{Quick Animation} \arrow[d, "\text{output}"'] \\
\text{Graphical animation}
\end{tikzcd}$$

Format of QuickAnimation File

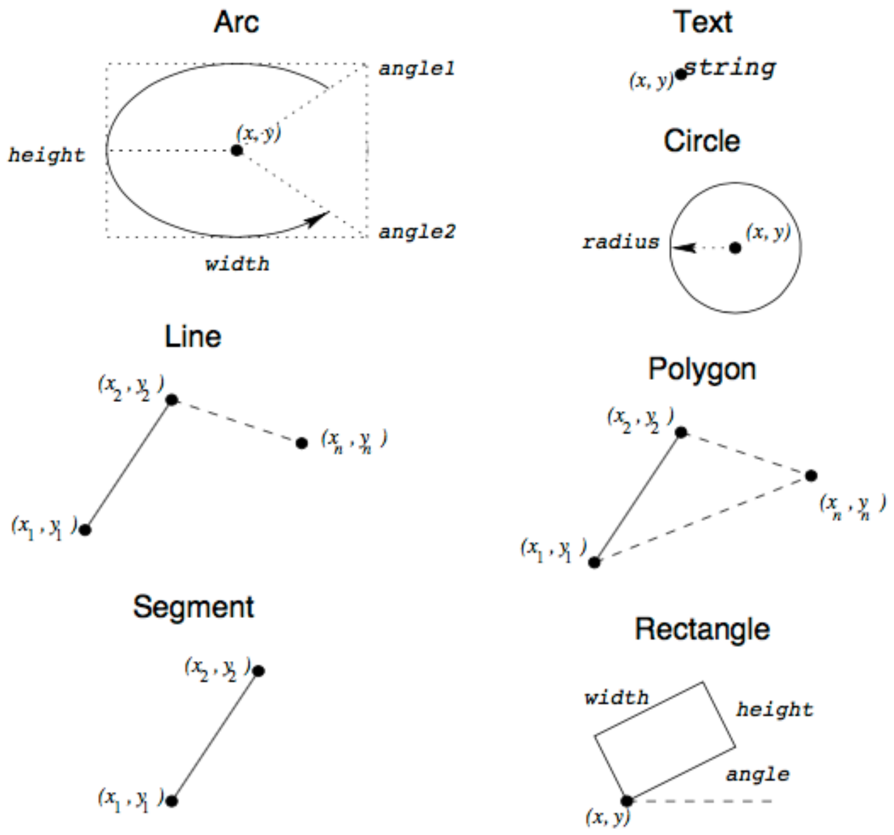
# *comment*
title "*title string*"
fixture
primitives data

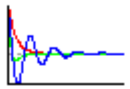
animate [restart | reverse]
# frame1
*primitives*1 data
stopped primitives2 data
. . .
primitivesn data

# frame2
*primitives*1 data
stopped primitives2 data
. . .
primitivesn data

General Primitives

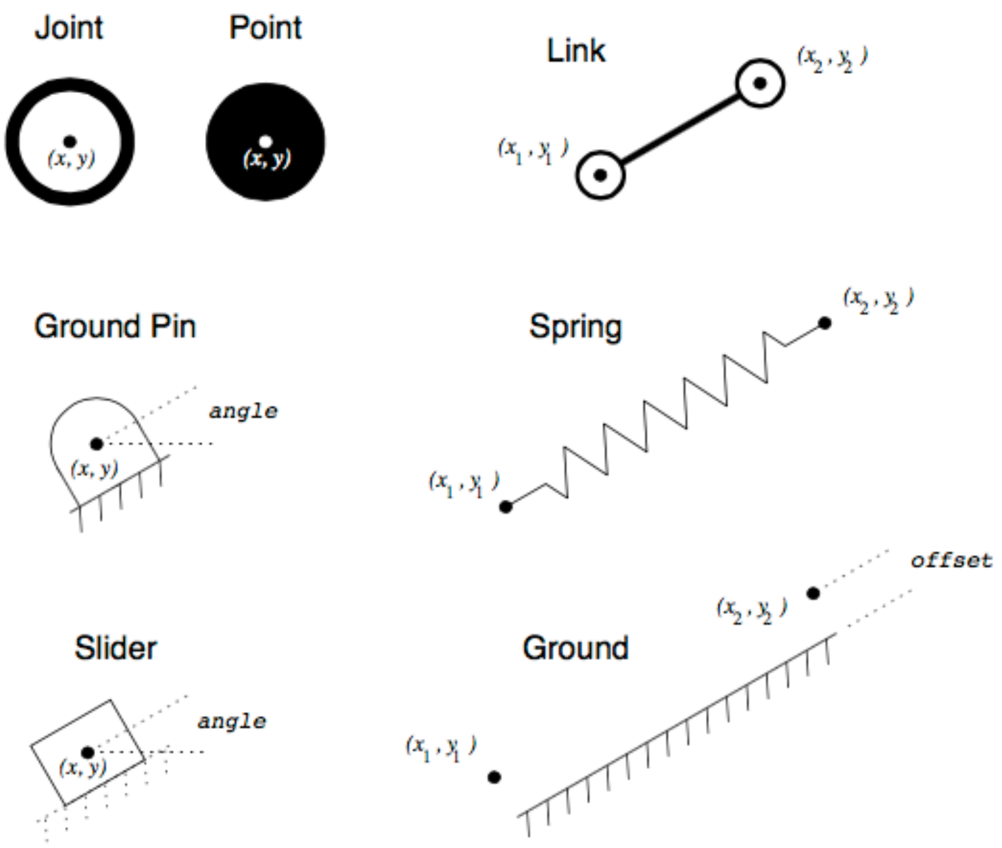
line x1 y1 x2 y2 [... xn yn]
arc x y *width height angle1 angle2*
segment x1 y1 x2 y2
rectangle x y *width height* [angle *angle*]
polygon x1 y1 x2 y2 x3 y3 ... xn yn
text x y *string*
circle x y *radius*
dot x y

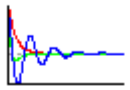




Mechanical Primitives

point x1 y1 [x2 y2 ... xn yn] [trace]
link x1 x2 x2 y2 [... xn yn]
ground x1 y1 x2 y2 [offset *pixeloffset*] [ticks *forward* | *backward*]
groundpin *x y* [angle *angle*]
slider *x y* [angle *angle*]
spring x1 y1 x2 y2





Drawing Options

- line/segment
  - [pen *color*]
  - [linewidth *pixelwidth*]
  - [linestyle solid | dashed [length *pixellength*] | dotted [gap *pixelgap*]]
  - [capstyle butt | round | projecting]
  - [joinstyle miter | round | bevel]
  - [depth *depth*]
- arc/circle/polygon/rectangle
  - [pen *color*]
    [fill *color* [intensity *percent*] [pattern *number*]]
  - [linewidth *pixelwidth*]
  - [linestyle solid | dashed [length *pixellength*] | dotted [gap *pixelgap*]]
  - [capstyle butt | round | projecting]
  - [joinstyle miter | round | bevel]
  - [depth *depth*]
- text
  - [pen *color*]
  - [depth *depth*]
  - [font *fontname*]
- dot
  - [pen *color*]
  - [depth *depth*]

Use QuickAnimation in Ch

- Output to QuickAnimation directly
- Output to files and then open the files with QuickAnimation