Zero to Hero: C++ Classes & Structs

[aps]

meeting[14] = '05/02/2019'



Society updates

- Discord link in the email
- AGM date TBD
 - Will have 2 weeks notice

Google Kick Start 2019 💂

- A global online coding competition
- Aimed at students
- Algorithmic challenges designed by Google engineers
- Top participants may be invited to interview at Google
- https://codingcompetitions.withgoogle.com/kickstart

Feedback & Meetings \(\price \pi \price \pr

- Feedback
 - o Good
 - Bad
 - General comments
 - Improvements
- Meetings
 - Talks
 - Workshops
 - Anything else ???

- Where to send:
 - In person
 - Discord
 - Email
 - (abertaysa.com)
 - Airtable



Structs

```
/* A predefined object structure that is made up of
other types. E.g: */
struct ApsStruct{
  int integer;
   char character;
   custom myType;
// Access defaults to public.
```

Classes

```
/* A predefined object structure that is made up of
other types. E.g: */
class ApsClass{
  int integer;
   char character;
   custom myCustomType;
/* Essentially the same as a struct (in C++) except
everything defaults to private. */
```

What does that mean?

```
/* struct members can be accessed easily*/
ApsStruct exampleStruct;
exampleStruct.integer = 5;
std::cout << exampleStruct.integer << std::endl;</pre>
/* unless public, class members cannot be accessed
like this ^
Two options available; the public/private labels, or
special functions called getters/setters*/
```

What does that mean? (to clarify)

```
class ApsClass{
public:
  int integer;
   char character;
   custom myCustomType;
} /* In this scenario ApsClass could be used similarly
to ApsStruct, however the getter/setter build pattern
is more than just an annoyance - it saves you from
yourself */
```

Saves me from myself?

```
ApsClass exampleClass;
exampleClass.integer = 1337;
if (exampleClass.integer = 1234) {
   // will always do thing (notice above)
/* Among other things, Getters and setters can allow
different access levels -the get may be public, but
the set could be protected, and/or lazy loading (only
storing resource when it is needed (file handling?).*/
```

Inheritance

```
// Structs and classes are 2 sides of the same coin
// Which allows the following (I'm sorry!)
struct InheritanceStruct : ApsClass {
   // Default Inheritance = private
// And also this
class InheritanceClass : ApsStruct {
   // Default Inheritance = public
```

Inheritance

```
// More examples
  bool isPet = false;
  int lives = 9;
  bool isPet = true;
cat miaow;
std::cout << "Miaow is a pet: " << miaow.isPet << ", and likes "<< miaow.likes << std::endl;</pre>
```

Inheritance

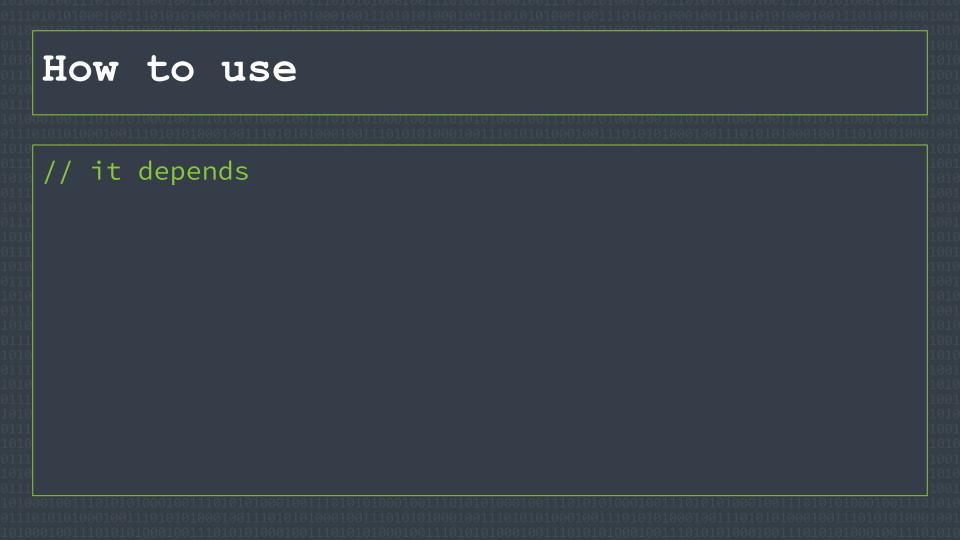
```
// You can extend any class/struct that allows it...
// (example has been trimmed for demo purposes)
     std::string default message = "An unknown error occurred!";
      DetailedException(std::string &custom message) throw() : std::runtime error(custom message.c str()) {}
      DetailedException() throw() : std::runtime error(default message.c str()) {}
  std::string default message = "Could not open file: FileNotFound";
```

Tip for constructors

```
DeviceManager(const std::string &ip or devname, bool verbose) {
DeviceManager(const std::string &ip or devname) : DeviceManager(ip or devname, false) {
```

When to use

```
// students example
All students have:
   names
   modules (name, desc, grade)
   start year, end year
student functions:
   getGPA(), getName(), setName(), getTerm(), setEnd()
module functions:
   getName(), getDesc(), getGrade(), setGrade()
```



Top tips

- Class = noun
- Function = verb
- Attribute = adjective

Extra problem

A local library is cataloguing their collection of books. Each book has an author, title, and year associated with it. Different books might have the same author, title, or year.