struct
map

C++ struct (structured data)

Before there were “classes” there was “struct”.

structs are nothing more than associations/groupings of data elements. Very much like data-only classes.

C++ struct (structured data)

Before there were “classes” there was “struct”.

structs are nothing more than associations/groupings of data elements. Very much like data-only classes.

```
struct Animal {

char scientific_name[50];

int agv_weight_kg;

int lifespan_days;

}
```

struct vs data class

The only **difference between** a **struct** and **class** in **C++ is** the default accessibility of member variables and methods.

Whenever there is **at least one private member** in the structure, choose a class.

Example (see demo code):

d_class.pet_name = “spot”;

STL Map

Maps are associative containers that store elements in key: value pairs.

Each element has a key value and a mapped value. Keys must be unique. Values can be accessed by their keys.

```
#include <map>

std::map<std::string, int> my_map;
```

STL Map

```
std::map<std::string, int> my_map;
```

Key: std::string

Value: int

STL Map: `std::map<std::string, int> my_map;`

```
my_map[“Murder Moist Foul”] = 22312;    // title, word count

using std::string;

using std::pair;

my_map.insert(pair<string, int>(“The Quiet Type", 67554);
```

STL Map: `std::map<std::string, int> my_map;`

```
my_map[“Murder Moist Foul”] = 22312;    // title, word count
```

Key: “Murder Moist Foul”

Value: 22312

STL Map: `std::map<std::string, int> my_map;`

```
using std::string;

using std::pair;

my_map.insert(pair<string, int>(“The Quiet Type", 67554);
```

Map

begin()
end()
size()
max_size()
empty() – Returns whether the map is empty
insert(keyvalue, mapvalue)
erase(iterator position) – Removes the element at the position pointed by the iterator
erase(const g)– Removes the key value ‘g’ from the map
clear() – Removes all the elements from the map

readme.txt

trainingimages:

5000 training digits,

around 500 samples from each digit class.

Each digit is of size 28x28, and the digits are concatenated together vertically. The file is in ASCII format, with three possible characters. ' ' means a white (background) pixel, '+' means a gray pixel, and '#' means a black (foreground) pixel.

traininglabels:

a vector of ground truth labels for every digit from trainingimages.