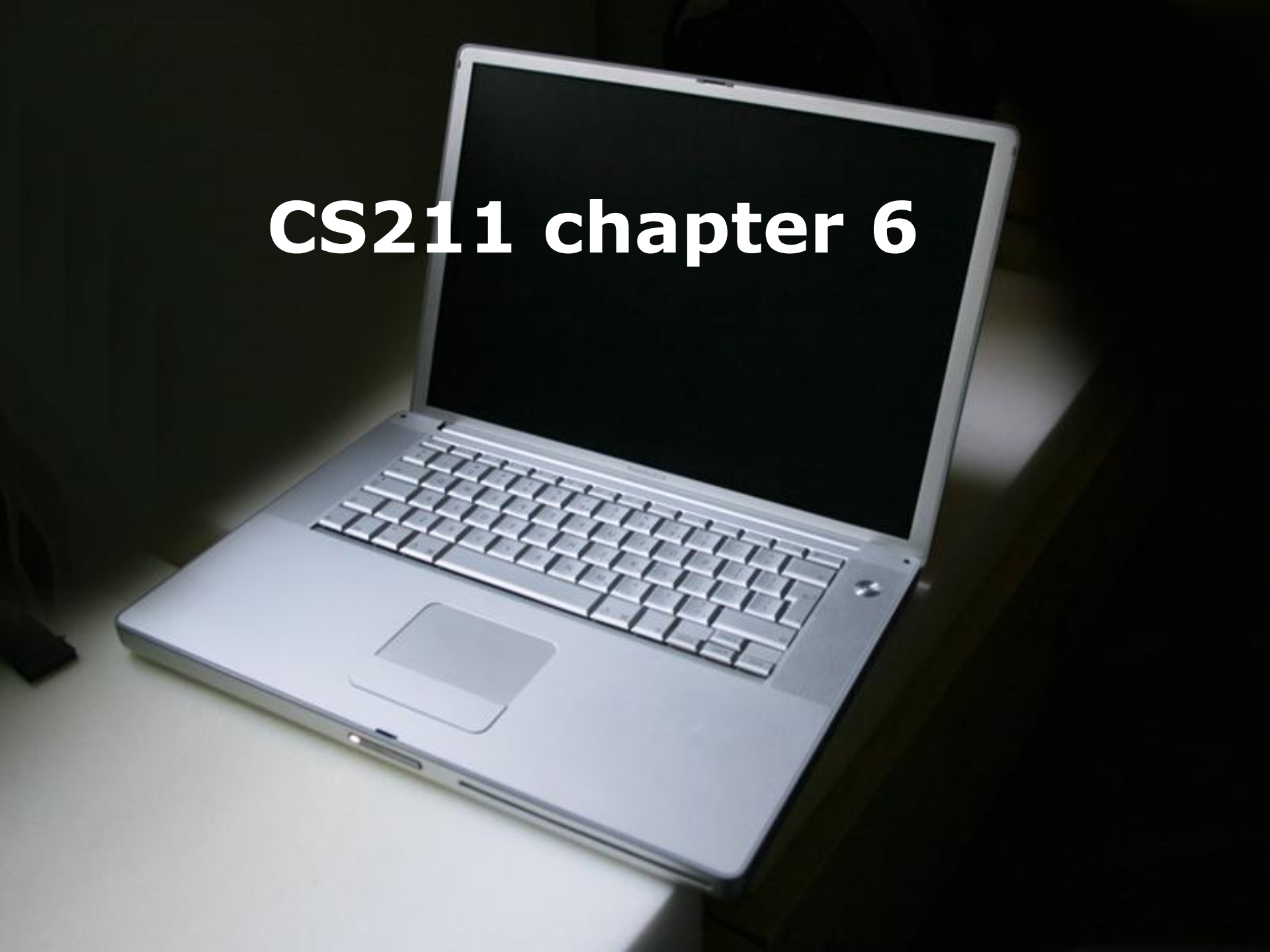

CS211 chapter 6

Type Definition - struct:

There are million type of complicated things, which can’t describe just as text, numbers, etc…

sturct - A collection of data items of diverse types

Structure:

```
struct Struct_tag {
  type1 member_name1;
  type2 member_name2;
  …
} ;

other functions …
int main() { … }
```

Example:

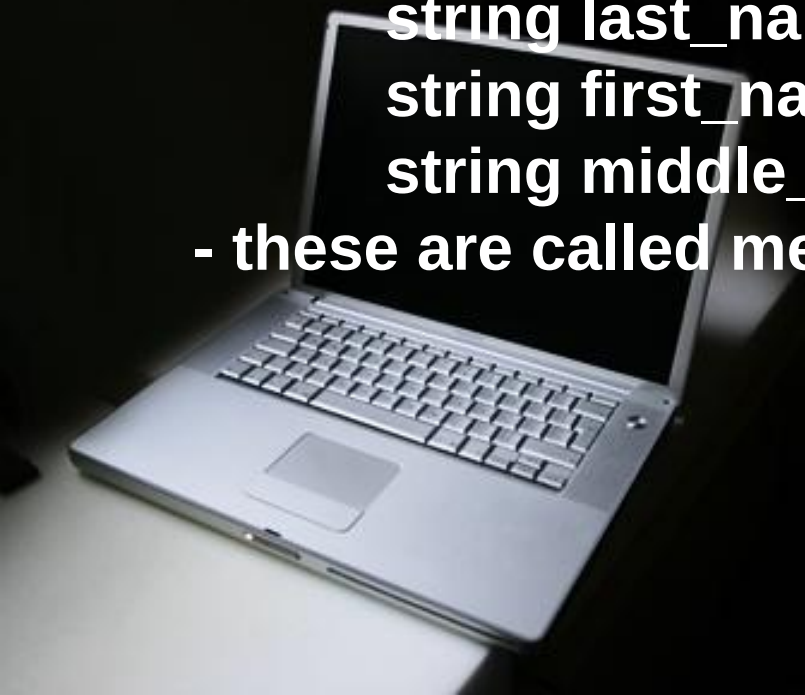
Let's say we want to define a type – Name

What component makes up the whole name?
Given name, Surname, middle name.

struct Name need variables to hold these values.

```
string last_name,
string first_name,
string middle_name
```

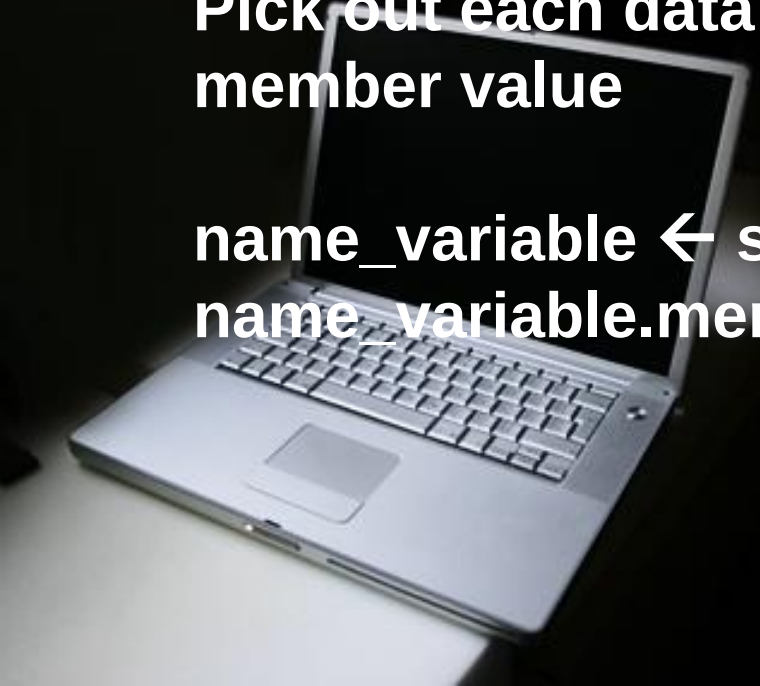
- these are called member names.



Example:

After defining structure - Name, you may create Name variables.
Ex:
Name name_variable;
You may use this collection of data as a whole – called structure value.
Pick out each data from this collection – called member value

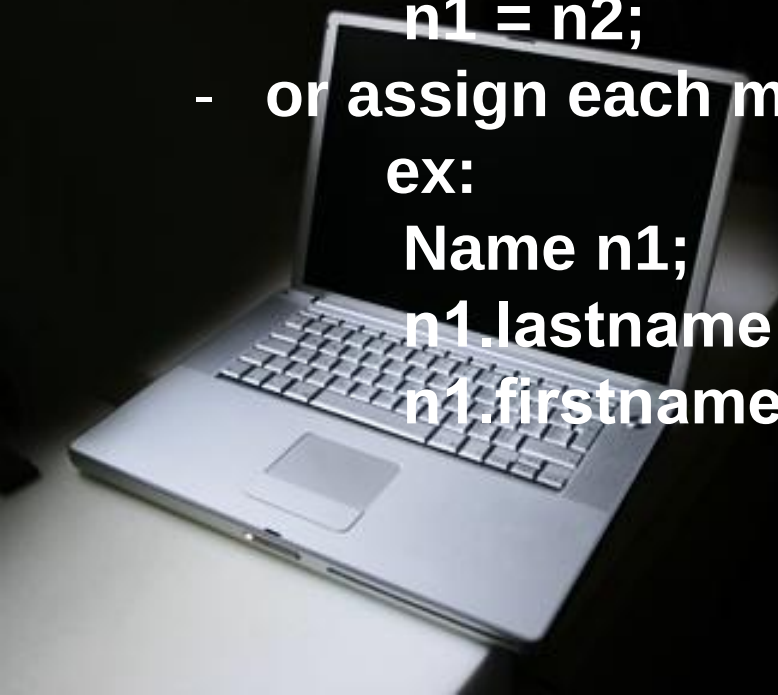
name_variable ← structure variable
name_variable.member_name ← member variable



Example:

Each Name variable now has 3 member variables
To assign the value to these variable

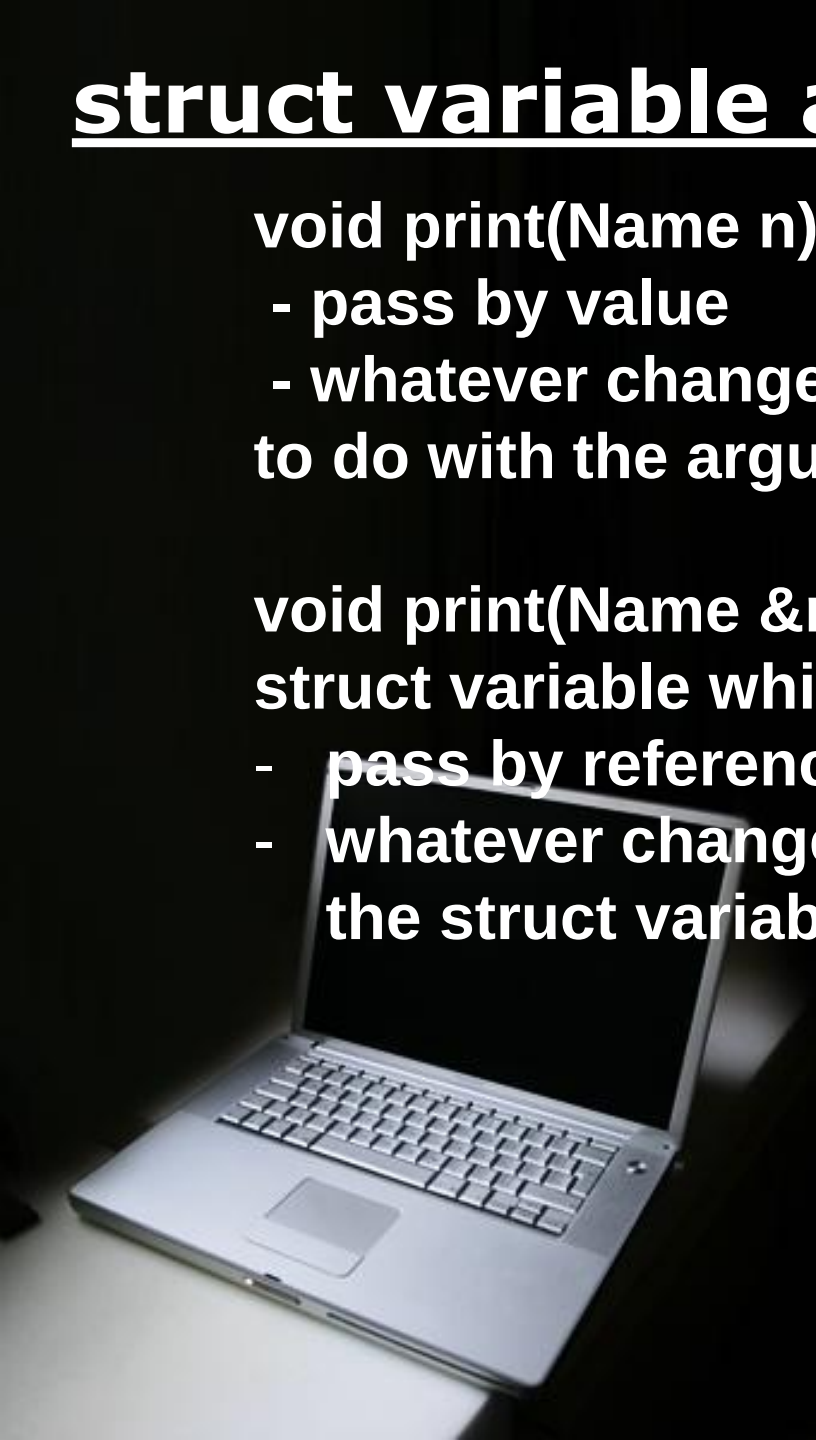
- we may assign the structure variable as a whole
  ex:
  Name n1, n2;
  n1 = n2;
- or assign each member variable individually
  ex:
  Name n1;
  n1.lastname = “Yang”;
  n1.firstname = “Kangmei”;



struct variable as function arguments:

void print(Name n) ← this creates a copy of type Name
- pass by value
- whatever change made to this parameter, has nothing to do with the argument one

void print(Name &n) ← n in this function is alias for the struct variable which has passed in as an argument
- pass by reference
- whatever change made in this function, also effects the struct variable which passed in.



May return a struct:

```
Name createNew(){
 Name temp;
 temp.lastname = “A”;
 temp.firstname = “B”;
 return temp;
}
```

Using struct in another sturct:

As the struct is a collection of data of diverse type, which including other defined type.

```
struct Student{
 Name name;
 double GPA;
 string id;
};
```

Student s1; ← Student
s1.GPA; ← double
s1.id; ← string
s1.name; ← Name;
s1.name.lastname; ← string
s1.name.hasMiddle; ← bool