structure

Dr. Hyun Lim

h.lim@lboro.ac.uk

Institute for Digital Technologies

Loughborough University London

struct (structure in C)

- A **struct** in C is a **complex data type** declaration that defines a physically **grouped list of variables** to be placed **under one name** in a block of memory

- It allows the **different variables** to be accessed via a single pointer, or the struct declared name which returns the same address.

- struct(in C), class (in C++/Python)

- *cf.*, An **array** is also a **kind of data structure** that can store a collection of data elements of the **same type**.

Name of a structure
(or *Tag*)

```
struct book{
   char author[20];
    char title[30];
    int page, year;
    …
  } Book;
```

Member variables

Member functions (***method***)

instance
(It is optional as we can create it somewhere else in the code)

```c
#include <stdio.h>
#include <string.h>
struct book // define a structure
  {  char author[20];
     char title[30];
     int page, year; };

int main() {
  struct book bk1, bk2; // declare structure instances
  bk1.year = 2019;    bk1.page = 230; // assign values to those structures' members
  strcpy(bk1.author, "Mike Taylor");       strcpy(bk1.title, "C Programming");
    // In C we cannot assign a string value to a char array (string), but have to copy
    // into them with strcpy(), memcpy() or similar.
    // bk1.author = "Mike Taylor"; // error C2106
    // let copy the string into the char array, instead
    // access and print those values to standard output
  printf("\n\n[bk1]\nAuthor: %s\n", bk1.author);
  printf("Title: %s\nYear: %i, Page: %i \n\n", bk1.title, bk1.year, bk1.page);
 // while (!kbhit()); // optional
return 0;
}
```

Exercise

- Assign values to bk2
- And show the info of bk2

Structure Array

Questions

- If we have to store information for 1000 books, do we need to create 1000 instances, e.g. bk1, bk2, bk3,…..bk1000?
- Let's be smart. If items have the same data type/ structure, we can use an **array**, i.e. **structure array**, such as:

```
struct book bk[1000];
```

an array

data type

if the type is ***struct**ure*

Homework

1) Assign values for 3 books to bk[ ], i.e. bk[0], bk[1], bk[2]

✓ Use a **Structure Array**, e.g. **struct** book bk[100];

2) Use a (**for** or **while**) **loop** to show all the info of bk[0]~bk[2]

Summary - struct (in C)

- A `struct` (in C) is very similar to a `class` (in C++/ Python)
- In order to use a structure, we must first declare a structure template:

```
struct str_name {
   int mbr0; float mbr1;
 } name1, name2;
```

- The variables in a structure are called ***members***.
- In C, you must explicitly use the `struct` keyword to declare a structure and create an ***instance***, i.e. *my_str*:

```
int main() { ................
   struct str_name my_str; ...... }
```

- Optionally declaring ***instances*** by placing one or more comma-separated instance names at the end of a struct. ***instance*** can be declared somewhere in the code.