Structures and Functions

Last updated 6/3/24

These slides introduce using structures in functions

Structures and Functions

```
typedef struct {
  int id;
  char name[26];
  float age;
} student ;
```

- Passing individual structure member values
  - Passing individual structure member values works just like any other value

```
void fun1 (float zoo);
void fun2 (float * soo);
```

by value

```
fun1(my_struct.age);      // passes the member value age of
                          // my_struct to function fun1
```

via pointer

```
fun2(&my_struct.age);     // passes a pointer to my_struct
                          // member age (the address)
                          // to function fun2
```

Structures and Functions

```
typedef struct {
   int id;
   char name[26];
   float age;
} student ;
```

- Passing individual structure pointer member values
  - Passing individual structure member pointer values works just like any other pointer

```
void fun1 (float zoo);
void fun2 (float * soo);
```

by value

```
fun1(my_struct_ptr->age);      // passes the member value age of
                               // my_struct_ptr to function fun1
```

via pointer

```
fun2(&my_struct_ptr->age);     // passes a pointer to my_struct_ptr
                               // member age (the address)
                               // to function fun2
```

Structures and Functions

```
typedef struct {
   int id;
   char name[26];
   float age;
} student ;
```

- Passing the whole structure to a function – by value
  - When a structure is passed to a function by value, a copy of the entire structure is made for the function to use
  - Stored on the stack just like any other variable

function declaration

```
void fun3(student the_struct);   // the notation type name
                                 // tells the compiler it is expecting
                                 // structure of type student
```

call

```
fun3(my_struct1);                // passes a copy of the entire structure
```

Structures and Functions

```
typedef struct {
   int id;
   char name[26];
   float age;
} student ;
```

- Passing the whole structure to a function – by pointer
  - When a structure is passed to a function by pointer, no copy of the structure is made

function declaration

```
void fun3(student * the_struct_ptr);      // the notation type * name
                                          // tells the compiler it is expecting
                                          // pointer to a structure of type
                                          // student
```

call

```
fun3(&my_struct1);          // passes a pointer to the structure
   or
fun3(my_struct1_ptr);       // passes a pointer to the structure
```

Structures and Functions

```
typedef struct {
   int id;
   char name[26];
   float age;
} student ;
```

- Accessing the structure inside a function – by value
  - We passed a copy to the structure into the function

```
void fun3(student the_struct);          // the notation type name
                                        // tells the compiler it is expecting
                                        // structure of type student

fun3(my_struct1);                       // passes a copy of the entire structure
```

  - Inside the function, the_struct is a copy of the passed structure
  - To access an element of the structure we can use the structure access operator

inside the function

```
foo = the_struct.age;

the_struct.id = 22345;          // remember – we are changing
                                // the copy

scanf("%s", &the_struct.name);
```

Structures and Functions

```
typedef struct {
   int id;
   char name[26];
   float age;
} student ;
```

- Accessing the structure inside a function – by pointer
  - We passed a pointer to the structure into the function

```
void fun3(student * the_struct_ptr);     // the notation type name
                                         // tells the compiler it is expecting a pointer
                                         // to a structure of type student

fun3(&my_struct1);                       // passes a pointer to the entire structure
  or
fun3(my_struct1_ptr);                    // passes a pointer to the entire structure
```

  - Inside the function, the_struct_ptr is a pointer to the structure
  - To access an element of the structure we can use the structure pointer access operator

inside the function

```
foo = the_struct_ptr->age;

the_struct_ptr->id = 22345;          // remember – we are changing the
                                     // original

scanf("%s", &the_struct_ptr->name);
```

Structures and Functions

- Passing structures – non-modifiable
  - What if we want to pass the structure to a function but we do not want the function to modify the structure?
  - Declare the structure as a constant in the function declaration and definition

```
float fun1(student the_struct);              // modifiable
→
float fun1(const student the_struct);        // non-modifiable


float fun1(student * the_struct_ptr);        // modifiable
→
float fun1(const student * the_struct_ptr); // non-modifiable
```