

Day17 A

Young W. Lim

2017-10-31 Tue

1 Based on

2 Structures

- Overview
- Definitions of Structures

"C How to Program", Paul Deitel and Harvey Deitel

I, the copyright holder of this work, hereby publish it under the following licenses: GNU head Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled GNU Free Documentation License.

CC BY SA This file is licensed under the Creative Commons Attribution ShareAlike 3.0 Unported License. In short: you are free to share and make derivative works of the file under the conditions that you appropriately attribute it, and that you distribute it only under a license compatible with this one.

- collections of related variables under one name (as one unit)
 - the contained variables may have different types
- commonly used to define records to be stored in files
- can form more complex data structures together with pointers
 - linked list
 - queues
 - stacks
 - trees

Keyword struct and tag

- keyword `struct`
- identifier *tag*
 - names the structure definition
 - make the structure definition unique
- `struct tag-name` can be thought as a new type
 - can declare variables of this new type

Structure Members

- variables declared within the braces of of the structure definition
- { member declarations }
- the collection of related variables of different types
- records in files to be stored
- these variables must be unique within a structure definition
- structure definitions ends with a semicolon ;

Member Types

- primitive data types
 - int, float, ...
- aggregate data types
 - arrays, other structure types
- members cannot contain
 - an instance of itself (the defining structure type)
- members can contain
 - pointer to itself (the defining structure type)
- self-referential structure
 - a structure containing a pointer member which points to the same defining structure type
 - used in linked list data structure

Structure Variable Declarations (1)

- a structure definition creates a new data type
- can declare variables of this new data type
- `struct tag-name { ... } ;`
- `struct tag-name var1, var2 ;`

- can put a comma-separated list of variable names between the closing brace and ending semicolon
- `struct tag-name { ... } var1, var2 ;`

- when there is only one structure type
 - no need to differentiate the structure type
 - can omit the tag-name (tag-name is optional)
- `struct { ... } var1, var2 ;`

Structure Variable Declarations (2)

- `struct tag-name { ... } ;`
 - defining a new type
- `struct tag-name var1, var2 ;`
 - variable declaration
- `struct tag-name { ... } var1, var2 ;`
 - defining a new type and
 - declaring variables
- `struct { ... } var1, var2 ;`
 - defining a new type and
 - declaring variables
 - no other structure type

Structure Variable Operations

- assigning a structure variables to variables of the same type
- taking the address of a structure variable (&)
- accessing the members of a structure variable (.)
- determining the size of a structure variable (sizeof)