Day02 A

Young W. Lim

2017-10-07 Sat

Outline

1. Based on

2. Introduction (2) - Basic Elements
   - Basic Elements in C Programming

Based on

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

comment and white space

- comments // : to the end of a line
- comments /* . . . */ : start and end
- every c statement ; (statement terminator)
- ignore white spaces : spaces, tabs, newlines(enter)
- a statement across many lines

main()

- # preprocessor directives
- `<stdio.h>` file must be included to use printf()
- must have 1+ functions
- must always have `main()` function
- *building* block `{ ... }`
- function : many inputs, one output (return value)
- `main()` returns an int value (used by the shell)
- void means no input in `main(void)`

variables

- a variable is stored in a location in memory
- a variable of type `int` can hold numbers without decimal point
- a variable definition: type + name
- a variable name : a valid identifier {letter, digits, _}
- case sensitive : upper case and lower case
- `x = ...` : writing a new value to `x` , LHS(Left Hand Side)
- `...  = x` : reading a curent value of `x` , RHS(Right Hand Side)
- access: read or write operation
- a variable must be *defined* bedfore it can be *accessed*

characters and strings

- "xyz..." : string with termination '\0'
- "x" : 1-character string with termination '\0'
- 'x' : a single character
- escape character \ to provide special meaning to escape sequence
- escape sequence \n : new line
- to print \ inside a string : use \\
- to print " inside a string : use \"

printf() and scanf()

- `printf()` is a standard library <u>output</u> function (stdio)
- `scanf()` is a standard library <u>input</u> function (stdio)
- `"%d %c %s"`: format string
- `%d` for `int` type i/o
- `%c` for `char` type i/o
- `%s` for *string* type i/o
- to print % inside a string : use %%
- `&x` : the location of a variable x in memory : address
- `printf( ... x ...);` call by value : printf <u>cannot</u> change x
- `scanf( ... &x ...);` call by refrence : scanf <u>can</u> change x

arithmetic operators

- + addition
- - subtraction
- * multiplication
- / division
- no power operator (^), use `pow(x,y)` in `<math.h>`
- many mathematical functions in math standard library
- parenthesis
- operator precedence
- from left to right
- expression

equality and relational operators

- equality operators
  - == equal?
  - != not equal?
- relational operators
  - > greater?
  - >= greater or equal?
- assignment
  - = : LHS <- RHS
- equality op > relational op > assignment op in precedence level

errors

- Syntax Error: grammatically wrong
  - error messages
  - warning messages
  - note messages
- Semantic Error: algorithmically wrong

secure c programming

- use `printf("%s", "Hello\n");` instead of `printf("Hello\n");`
- difference between `printf_s` and `printf`
  - `printf_s` checks the format string for valid formatting characters
  - `printf` only checks if the format string is a null pointer