

Introduction to C Programming

– Basics of Programming (1) : Variables · Standard Input/Output (1) –

Waseda University

Today's Topics

- How to write a program in C language.
- How to use **variable**
 - integer type
 - double type
 - character type
- Using assignment and four arithmetic operations
 - Assignment `r = j` , Arithmetic operations `r + - * ÷ j` , Remainder `r % j`
- How to use standard input/output
 - Output to the display: `printf();`
 - Input by using keyboard: `scanf();`

How to create

Process1

Creating a source file by using editor (Emacs)

Process 2

Building a executable file by using gcc (gnu C complier). If compiler outputs any errors and warnings, you will edit the source file.

Process 3

Running the excuse file and looking into the program. If the result is wrong, you will edit the source file again.

Example 1

- Create a file "hello.c" by using editor (Emacs)
- Write a program like the followings:

```
#include <stdio.h>
int main(void){
    printf("Hello, world!! \n");
    return 0;
}
```

- printf() : Output characters and variables

```
printf(" Hello, world!!\n");
```

```
printf(" Lucky number is %d.\n", Age);
```

- "\n" means line break

Compiler

Building an executable file by using gcc

```
[~/work] $ gcc hello.c
```

- If compiler outputs any errors and warnings, you will edit the program by using editor.
- If compiling is succeeded, creating the executable file 「**a.out**」

```
[~/work] $ gcc hello.c -o hello
```

Output all warnings

```
[~/work] $ gcc -Wall -o hello hello.c
```

Examples of errors and warnings

hello.c:1:19 error: studio.h: No such file or directory

- 「File name: Line: Word:」
- Wrong in `#include<>`

hello.c:3: warning: character constant too long for its type

- Mistake 「`"`」 for 「`'`」;

hello.c:6: error: expected '`'`' before '`return`'

hello.c:3: error: undefined reference to '`print`'

- Mistake 「`printf`」 for 「`print`」

Execute and Validate the Program

Running the executable file

```
[~/work] $ ./a.out
```

```
Hello, world!
```

- If you run the executable file in current directory, you have to write `./` in front of the file name.

Examples of failure

```
[~/work] $ ./a.out
```

```
Hello, world!
[~/work] $
```

Mistaking `¥` for `/`

Variables

- To handle numbers and characters, we use a "Variable" as data storage.
- In order to use a variable, you must declare a variable name in the source code first.

Variables

- Each variable has a specific type.
- int : This data type can contain an integer (-2,147,483,648 ~ 2,147,483,647)
- double : This data type can contain a real number (a floating point number). (0 and 2.23×10^{-308} ~ 1.79×10^{308})
- char : This data type can contain a character.

Variable

Variable name

- Variable name is made with alphabetical letters and `_` (`ab_c`)
- You can use numbers after the first character. (`a2`)
- You can not use **Reserved word**. (`int`, `double`, and so on)
- lower-case characters and upper-case characters will be treated as different names. (`abc` \neq `Abc`)
- It is better to choose a name describing the data stored in the variable.

Reserved word for C

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

Example of programming with the int type

```
int.c
#include <stdio.h>
int main(void){
    int Age ;                               /*Declaring a variable*/
    printf("Input your age: ");              /*Output*/
    scanf("%d", &Age);                      /*Input*/
    Age = (Age - 18) * 3+2;                  /*Calcuration and assignment*/
    printf("Lucky number is %d.¥n", Age);
    return 0;
}
```

Example: Running the above program

- File name: int.c
- [~/work] \$ gcc int.c
- [~/work] \$./a.out
- Input your age: 19 **【Enter】**
- Lucky number is 5.

Explanation of the program

- Don't forget to put " ; " at the end of each line.
 - 1~2: Fixed phrases
 - 3: Declaring the int type variable **Age**
 - 4: printf(" ~"), output ~.
 - 5: scanf is input routine. %d means integer, it inserts the input number in Age.
 - 6: " =" means assignment. (not an equation)
 - 7: "\n" indicates line break.
 - 8: Fixed phrases
- The others
 - Priority of arithmetic operations is the same as in mathematics.
 - "/* ... */" means comment out. "... " are ignored.
 - You need the "&" to use the scanf function.
 - Note that "&" is not required for the printf function.

Standard output

printf(): Output routine for numbers and characters

- Output
(The part enclosed within double quotes will be output on the screen)

```
printf("Input your age: ");
```

- The value stored in the int type variable **Age** is output

```
printf("Lucky number is %d.\n", Age);
```

- Output the number of int type data **Age**
- "%" is called string variables. Each string variable has specific type. For example, int type is "%d"
- The variable value will be displayed in "%d"

Standard input

scanf(): Input routine from the Keyboard

- Input numbers and characters

```
scanf("%d", &Age);
```

- The keyboard value input is stored as the int type variable "**Age**"
- You need to write a specific code to a variable type int type is "%d"
- Writing ' & ' before variable names.

Arithmetic operations and assignment

Operations "+, -, *, /, %"

- C Programming Language supports arithmetic operations "+, -, *, /, %"
- Order of operations is the same as in mathematics
- The result of division "int/int" becomes int type. (Fractional part is truncated)

- Example (operator.c)

```
#include <stdio.h>

int main(void) {
    int a=5, b=3;
    printf("a+b=%d\n", a+b); /*add*/
    printf("a-b=%d\n", a-b); /*sub*/
    printf("a*b=%d\n", a*b); /*mul*/
    printf("a/b=%d\n", a/b); /*div*/
    printf("a%b=%d\n", a%b); /*mod*/
    return 0;
}
```

- Result

```
[~ /work] $ gcc operator.c
```

```
[~ /work] $ ./a.out
```

```
a+b=8
```

```
a-b=2
```

```
a*b=15
```

```
a/b=1
```

```
a%b=2
```

Operations and assignment

Assignment 【=】

- "=" means assignment. (not equality)
- Assignment of a number: **a** = 3;
- Assignment of a variable: **a** = **b**;
- Assignment of the result of the right side: **Age** = (**Age** - 18) * 3 + 2;
/*Calculation, assignment*/
 - Note that the result of the operation overwrote the variable **Age**
- You can update the variable itself: **a** = **a** + 2;
 - It means that **a** is increased by 2.

Example 3

The program to input 3 integer values and display the sum of these values

- For example:

Input three integers:

a: 15 【Enter】

b: 23 【Enter】

c: 32 【Enter】

Sum is 70.

- Declare the 4 variables (**a, b, c, Sum**)
- The display asks to input 3 numbers in the **a, b, c**
- Calculate the summation of three integers
- Display the summation

Example4

The program to input 3 integer values and display the sum of these values and average

- For example:

Input three integers:

a: 15 【Enter】

b: 23 【Enter】

c: 32 【Enter】

Sum is 70, Average is 23.3.

- The result of division of int/int becomes the int data type (Fractional part is truncated) \Rightarrow We need to use double type.

Summary

- How to write a program in C language.
- How to use **variable**
 - integer type
 - double type
 - character type
- Using assignment and four arithmetic operations
 - Assignment `r = j` , Arithmetic operations `r + - * ÷ j` , Remainder `r % j`
- How to use standard input/output
 - Output to the display: `printf();`
 - Input by using keyboard: `scanf();`