



3. *Basic of C*

Dr. Weerakaset Suanpaga
(D.Eng)

**Department of Civil Engineering
Faculty of Engineering , Kasetsart
University
Bangkok, Thailand**

<https://pirun.ku.ac.th/~fengwks/ram/ge749/lect/lect1.pdf>

1

Hello, World 2

```
#include<stdio.h>
main()
{
    printf("How are you ? \n \t Fine ! \n");
}

    \n    newline
    \t    tab
```

add

```
#include<stdio.h>
main()
{
    int i, j;
    i = 1;
    j = 2;
    printf("The sum of i and j is %d \
n", i + j );
}
```

3

hypotenuse

```
#include<stdio.h>
#include<math.h>
main()
{
    double a = 3.0;
    double b = 4.0;
    double c;
    c = sqrt( a * a + b * b );
    printf("The hypotenuse = \t%f \n", c );
}
```

4

Variables I

- A space in the computer memory
- How to interpret bit pattern according to the declaration
- integer
- floating point
- pointer
- character in C, same as integer
- string in C, same as pointer

5

Variables II

- integer
 - char 1 byte -128 to 127
 - short 2 byte -32,768 to 32,767
 - int 2 or 4 byte same as short or same as long
- most suitable byte length for the system
 - long 4 byte -2,147,483,648 to 2,147,483,647
 - unsigned char 1 byte 0 to 255
 - unsigned short, int, long value is 0 or positive
 - » char i = 127;
 - » i = i + 1;
 - » printf(“%d \ n”, i);

6

Variables III

- Characters are stored as integers in C. Table is used to convert character to integer.
- Most computer systems use ASCII table
- Only few computer systems use EBCDIC table

7

Variables IV

- floating point
 - float 4 byte
 - 4 byte
 - IEEE format signature:1 bit, characteristic:8bit, mantisa:23bit
 - $-3.4028235E+38$ to $-1.1754944E-38$
 - 0
 - $1.175944E-38$ to $3.4028235E+38$
 - double 8 byte
 - IEEE format signature:1 bit, characteristic:11bit, mantisa:52bit
 - $-1.797693134862316D+308$ to $-2.225073858507201D-308$
 - 0
 - $2.225073858507201D-308$ to $1.797693134862316D+308$

8

Variables V

- pointer
 - pointer keeps address of memory.
 - 16bit OS 2byte
 - 0- 65535, pointer can deal with 64Kbyte memory space
 - 32bit OS 4byte
 - 0- 4,294,967,296, pointer can deal with 4Gbyte memory space
- `int * p; /* p is a pointer to int. */`
`int i = 3, j;`
`p = &i; /* p has a address of i */`
`j = *p; /* j has a value which p is pointing, so j is 3 now */`
- strings, array
`char *str = "Hello World"; /* str points the memory */`

9

Simplest Pattern of C Program

```
include_directives
void main()
{
    data_declaration_statement;
    executable_statement;
}
```

10

include directive

- Directives are the instructions for pre-compiler. Pre-compiler

modify source program before compiling

- Include directives use the `#include` keyword to read header files

before compiling.

- Header files provide information about
 - functions such as return type, type of argument, number of

argument

- constants, definitions commonly used in the program

```
#include<stdio.h>      #include<string.h>
```

```
#include<math.h>      #include<time.h>
```

```
#include<stdlib.h>     .....
```

```
#include"my_header.h"
```

11

data declaration statements

allocate memory for variables.

declare variables.

```
int          i;
```

```
unsigned char pixel;
```

```
long        count;
```

```
double      x, y, hypotenuse;
```

declare variables and initialize.

```
int         j, k = 1;
```

```
double     weight = 3.5, height;
```

12

executable statements

- call a function
- assign value to a variable (or other memory location)
- Transfer program control to a new location or perform an action conditionally

```
c = sqrt ( a * a + b * b );
*address = 4;
while( 1 ) {
    ....
    if( .... )
        break;
    ....
    continue;
    ....
}
```

13

Example 1

```
#include <stdio.h>
void main()
{
    int n;
    printf("Enter a number here:      ");
    scanf( "%d", &n );
    n = n * 2;
    printf("Twice the number is %d \n",
n);
}
```

14

Example 2

```
#include<stdio.h>
#include<math.h>
void main()
{
    double a, b, c;
    printf("Enter a here: ");
    scanf( "%lf", &a );
    printf("Enter b here: ");
    scanf( "%lf", &b );
    c = sqrt( a * a + b * b );
    printf("The hypotenuse is %f \n", c );
}
```

15

printf

```
#include<stdio.h>
main()
{
    int        i;
    float      f;
    double     d;
    i = 'A';
    printf("%d\t", i );
    printf("%c \ t", i );
    printf("%x \ n", i );
    f = 4.0;
    printf("%f \ t", f );
    printf("%8.2f \ n", f );
    d = 5.0;
    printf("%lf \ t", d );
    printf("%8.2lf \ n", f );
    printf("%8.2lg \ n", f );
    d = 5.0E25;
    printf("%8.2lg \ t", d );
    printf("%8.2le \ n", d );
}
```

16


```
#include<stdio.h>
main()
{
    int            i;
    float          f;
    double         d;
    double         x1, x2;
    scanf("%d", &i );           printf("%d \ n", i );
    scanf("%f", &f );           printf("%f \ n", f );
    scanf("%lf", &d );          printf("%lf \ n", d );
    );
    i = scanf("%lf %lf", &x1, &x2 );
    printf("i:%d \ tx1:%lf \ tx2:%lf \ n", i, x1,
    x2 );
}
```

17

Change value of variables in calling function

```
#include<stdio.h>
int mul2( int x );
```

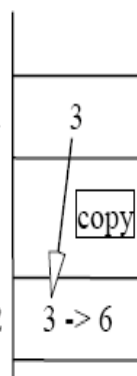
```
main()
```

```
{
    int a = 3;
    mul2( a );
    printf("%d\n", a );
}
```

```
mul2( int x)
```

```
{
    x = x * 2;
}
```

memory



```
#include<stdio.h>
int mul2( int *x );
```

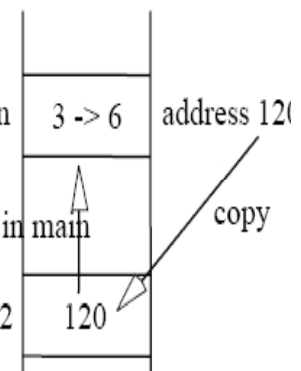
```
main()
```

```
{
    int a = 3;
    mul2( &a );
    printf("%d\n", a );
}
```

```
mul2( int *x)
```

```
{
    *x = (*x) * 2;
}
```

memory



18

```
#include <stdio.h>
int main()
{
    int divisor, quotient;
    scanf("%d", &divisor );
    if( divisor == 0 )
        return -1;
    quotient = 100 / divisor;
    printf("The quotient is %d \n", quotient );
    return 0;
}
if( condition )
    statement;
if(1)
    return -1;
printf("%d \n", divisor == 0 );
```

19

Function

```
include_directives
function_prototypes
void main()
{
    data_declaration_statement
    executable_statement
    variable = function_name( arguments );
}
return_type function_name( arguments )
{
    data_declaration_statement
    executable_statement
}
```

Example function

```
#include <stdio.h>

int plus(int x,int y);

int main(void)
{
    int a,b;
    int tot=0;

    printf("input 2 nubmer\n");scanf("%d %d",&a,&b);
    tot=plus(a,b);
    printf("Your result %d+%d=%d\n",a,b,tot);
    return 0;
}
int plus(int x,int y)
{
    int ans=0;
    ans=x+y;
    return ans;
}
```

21

example of a function

```
#include<stdio.h>
#include<math.h>
double hypotenuse( double a, double b );
void main()
{
    double  x1, x2, hypo;

    scanf("%lf %lf", &x1, &x2 );
    hypo = hypotenuse( x1, x2 );
    printf("hypotenuse is %lf \ n", hypo );
}
double hypotenuse( double a, double b )
{
    double c;
    c = sqrt( a * a + b * b );
    return c;
}
```

22

multiple source files

```
/*main.c*/
#include<stdio.h>
#include"my_head.h"
void main()
{
    double    x1, x2, hypo;
    while( 1){
        scanf("%lf %lf", &x1, &x2);
        printf("hypotenuse is %lf \n", hypotenuse(x1,x2);
    }
}
/* hypot.c */
#include<math.h>
#include"my_head.h"
double hypotenuse( double a, double b )
{
    return sqrt( a * a + b * b );
}
/* my_head.h */
/* hypotenuse takes two double
and returns double */
double hypotenuse( double a, double b );
```

23

Exercise - example

- Write a program to calculate the area of a square
 - Write all the steps in main function.
 - Separate the calculation part using function
 - Separate source file into 3. Main, function of calculating, and header file.

```
#include<stdio.h>
main()
{
    double side, area;
    printf("Area of Square \n");
    printf("Side \ t"); scanf("%lf", &side );
    area = side * side;
    printf("Area \ t%lf \n", area );
}
#include<stdio.h>
double area_square( double side );
main()
{
    double side, area;
    printf("Area of Square \n");
    printf("Side \ t"); scanf("%lf", &side );
    area = area_square( side );
    printf("Area \ t%lf \n", area );
}
double area_square( double side )
{
    return side * side;
```

24



Exercise

- Write a program to calculate the area of a triangle from base and perpendicular
 - Write all the steps in main function.
 - Separate the calculation part using function
 - Separate source file into 3. Main, function of calculating, and header file.
- Write a program to solve quadratic equation
 - Write all the steps in main function.
 - Separate a calculation part using function
 - The function should have 5 arguments, 3 for giving coefficient, 2 for getting answers
 - The function will return 0 when the equation is solved, -1 if cannot be solved
 - Separate source file into 3. Main, function of calculating, and header file.

25



Reference:

Assoc.Prof.Dr.HONDA Kiyoshi, Lecture Note .RS and GIS Field of study, School of Engineering and Technology ,AIT Thailand.2005

**Thank you for your kind
attention**

