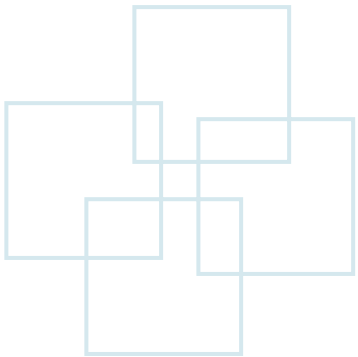


Chapter 2 Overview





Outline

- Structure of a program
- Programming style



Simple Example

```
01 /* a simple example */
02 #include <stdio.h>           /* include stdio.h */
03 #include <stdlib.h>         /* include stdlib.h */
04 /* main(): main function main */
04 int main(void)
05 {
06     int num;                 /* declare num as a variable*/
07     num=2;                   /* assign 2 to num */
08
09     /* call printf() function */
10     printf("I have %d cats.\n",num);
11     printf("You have %d cats.\n",num+2);
12     system("pause");        /* call system lib "pause" */
13     return 0;
14 } /* end of main() function */
```

Output:

I have 2 cats.
You have 4 cats.



How and Why to Comment

• How

1. Multi-line comment: Enclosed by `/* */`

- `/* this is a comment ;
this is the second line of a comment */`
- `/*

* This is a block of comments
******/`

2. Single-line comment: After `//`

- `printf("hello"); // print a message`

• Why

- Enhance the readability of a program



Header Files (#include <..>)

- **#include**: a type of preprocessor macro

- Put at the beginning of the program

- System header files

- Supply the need to invoke system calls and libraries

- Syntax: **#include <header_file_name>**

- Example: **#include <stdio.h>**

- Example: **#include <stdlib.h>**

- User-defined header files

- Define the related definitions needed in different source files

- Syntax: **#include "header_file_name"**

- Example: **#include "myheader.h"**



Header Files (Cont.)

```
/* prog 2_1, 簡單的C語言 */
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
    ...
}
```

含括標頭檔前

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

```
/*
 * stdio.h
 *
 * Copyright (c) 1989 by AT&T Intellectual Property.
 * All rights reserved.
 *
 * See the file stdio.h for details.
 */
#ifndef _STDIO_H
#define _STDIO_H
...
#endif
```

標頭檔 stdio.h

```
/*
 * stdlib.h
 *
 * Copyright (c) 1989 by AT&T Intellectual Property.
 * All rights reserved.
 *
 * See the file stdlib.h for details.
 */
#ifndef _STDLIB_H
#define _STDLIB_H
...
#endif
```

標頭檔 stdlib.h

```
/* prog 2_1, 簡單的C語言 */
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
    ...
}
```

含括標頭檔後

Remove #include



Compile error



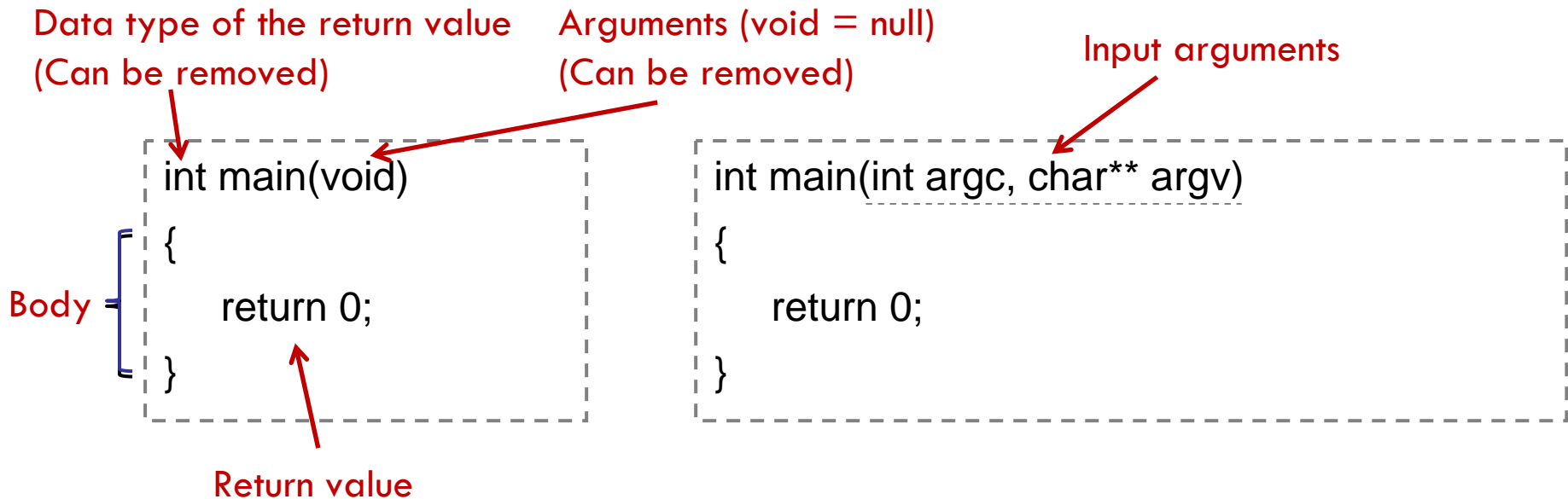
Header Files (Cont..)

- Where are the header files?
 - Visual Studio
 - C:\Program Files\Microsoft Visual Studio 9.0\VC\include
 - Dev-C++
 - **C:\Dev-Cpp\include**
- Some important headers
 - stdio.h: input and output
 - stdlib.h: system library
 - math.h: mathematical functions
 - string.h: string process function



Main Function: main()

- The starting point of a program
- Each program must contain **only one** main()





Variables

□ Declaration

- `int num;` `/* declare num as an integer variable */`
- `int a, b, c;` `/* declare a,b,c as three variables */`
- `float f = 0.5;` `/* declare f as a floating variable,`
`and set it to 0.5 */`

□ Case sensitive

- `num`, `Num`, and `NUM` are three different variables

□ The value of a variable can be updated in the program



Data Types

- **char**: character
 - Example: 'a', 'b', '2'
- **int**: integer,
 - Example: 23, -12, 12345
- **long**: long integer (occupying a larger size of memory)
- **short**: short integer (occupying a smaller size of memory)
- **float**: single-precision floating variable
 - Example: 0.123, -22.242
- **double**: double-precision floating variable



Naming Rules

- Can be **alphabet, digit, or underscore** (`_`)
 - Can begin with an alphabet or underscore (e.g., **abc** or **_NTUT**)
 - Can not include spaces (ex: a dog)
 - Can not begin with a digit (ex: 2num)
 - Can not use keywords (ex: for)
 - Uppercase characters are distinct from lowercase characters
- Improve readability
 - Begin with a lowercase letter
 - Use ***underscore*** or ***uppercase letter*** to combine multiple words
 - Example: totalnum => total_num or totalNum



Assign Value to a Variable

- Assign value during declaration

- int num = 2;
- int a = 1, b = 2;

Invalid assignment:

`int num = 1.5;`



Not an integer

- Assign value after declaration

- int num1, num2;
- char ch;
- num1 = 1; /* assign 1 to num1 */
- num2 = 2; /* assign 2 to num2 */
- ch = 'm'; /* assign 'm' to ch */

Invalid assignment:
`num1 = 1, num2 = 2;`

Cannot assign more than one variable
in one line after declaration



Output Function: printf()

- Print strings in the screen

- Syntax

- printf("string");
- printf("There are %d students.\n", numStudents);

The value of **numStudents** will be put in the position of **%d**



```
01 #include <stdio.h>
02 #include <stdlib.h>
03 int main(void) {
04 int numStudents = 50;
05 printf("There are %d students.\n", numStudents);
06 system("pause");
07 return 0;
08 }
```

Output:
There are 50 students.



Keywords

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



Type of Errors

- Syntax error
 - ***Invalid syntax***, which can be found by compiler.
- Semantic error
 - Also called ***logical error***, which results in the unexpected results



Syntax Error

```
01 #include <stdio.h>
02 #include <stdlib.h>
03 int main(void)
04 {
05     int num;                /* declare a variable "num" */
06     num = 2;                /* set "num" to 2 */
07     printf("I have %d dogs. \n"); /* print variable */
08     printf("You have %d dogs, too. \n"); /* print variable */
09     system("pause")
10     return 0;
11 )
```




Semantic Error

```
01 /* prog2_5, Syntax */
02 #include <stdio.h>
03 #include <stdlib.h>
04
05 int main(void)
06 {
07     int num = -2; /* declare num and assign -2 to it */
08     printf("I have %d dogs.\n", num);
09     system("pause");
10     return 0;
11 }
```


Can not be found by compiler



Programming Style


Indenting

```
main() {
    printf("a");
    printf("b");
}
```




or

```
main() {
printf("a");
printf("b");
}
```




Vertical alignment

```
main() {
    printf("a");
    printf("b");
}
```




or

```
main() {
    printf("a");
        printf("b");
}
```




Tabs

```
main() {
    printf("a");
    printf("b");
}
```



or

```
main() {
printf("a");
printf("b");
}
```





Programming Style

- Put a space after a comma
 - Example: `printf("test %d" (num));`
- Put space before and after an operator
 - Example: `sum = a + b;`



Lab 02

- Please write a program to output the following strings on the screen:

See you tomorrow.
Have a good night.

- Write a program to compute $12 + 34 + 56$ and print the result as follows (Please mind the spaces among integers):

Output: $12 + 34 + 56 =$ “the result”