

Somchai Prasitjutrakul

Outline

- Why use C ?
- A brief history of C
- COBOL vs. C
- "Hello world" program in C
- C program development cycle
- C Program Components

Why use C ?

- C is a powerful and flexible language.
- C is a popular language.
- C is a portable language.
- C is a language of few keywords.
- C is modular.
- C++

A Brief History of the C Language

- 1969 : BCPL, M. Richards
- 1970 : B, Ken Thompson
- 1972 : C, Dennis Ritchie
- 1983 : ANSI C
- 1985 : C++, Bjarne Stroustrup

AT&T Bell Laboratories
UNIX Operating System

COBOL vs. C : Overview

COBOL

- Business applications.
- Perform simple computations on large amount of data.
- The most widely used language today, but did not play a significant influence on later languages.
- English-like syntax, programs easily become verbose.

C

- Then : system programs.
Now : all sorts of programs and applications.
- Support complex computations and algorithms
- Possibility of producing extremely terse code, so readability can become a problem.

COBOL vs. C : Data Types

COBOL

- All data are stored in character string at run-time
 - converted only when being used as operands
 - hardware independent.
- Data can be grouped as components of records and arrays.
- Global variables.

C

- Character, integer, floating point number, and *pointer*.
- Same as COBOL : structures and arrays.
- *Local* and Global variables.
- Dynamic memory allocation.

COBOL vs. C : Control Structures

COBOL

- IF-THEN-ELSE, GOTO, PERFORM, EVALUATE
- The basic subprogram is simply a paragraph.
- Programmer-defined subprogram with parameters and local environment is an optional feature.
- No recursive calls.

C

- if, if-else, while, for, do, break, continue, goto, switch.
- Function is the only program-structuring mechanism.
- Functions can be recursively called.

COBOL vs. C : Misc.

COBOL

- Batch-processing operating environment
- A rich set of input-output primitives: sequential, indexed-sequential and direct-access files, sort-merge, report generator.
- About 300 reserved words

C


- Input-output facilities are not part of the C language. (the standard I/O library)
- A small set of reserved words (30-40)

"Hello World"

hello.c

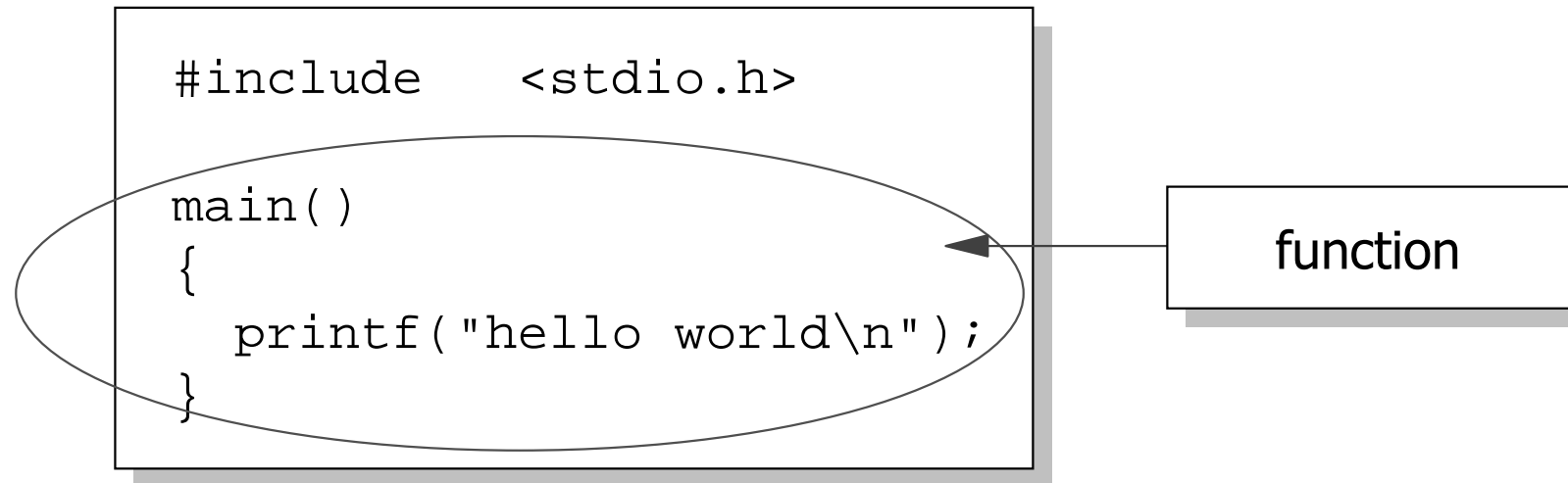
```
#include <stdio.h>

main()
{
    printf("hello world\n");
}
```



```
$ cc hello.c
$ a.out
hello world
$
```

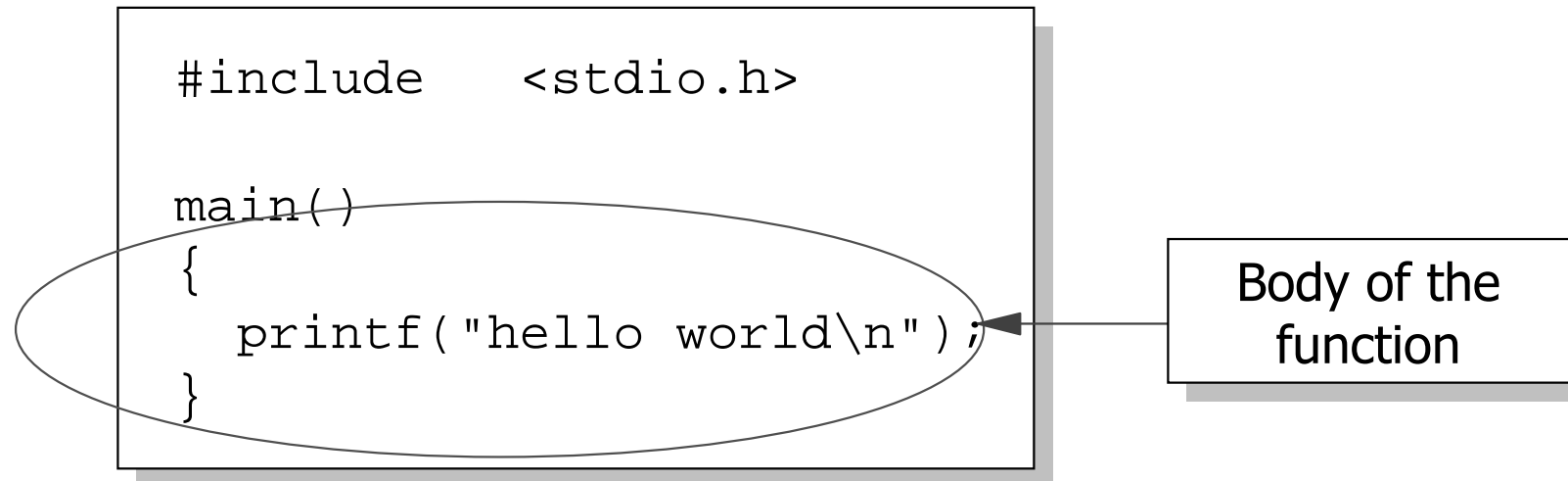
"Hello World"



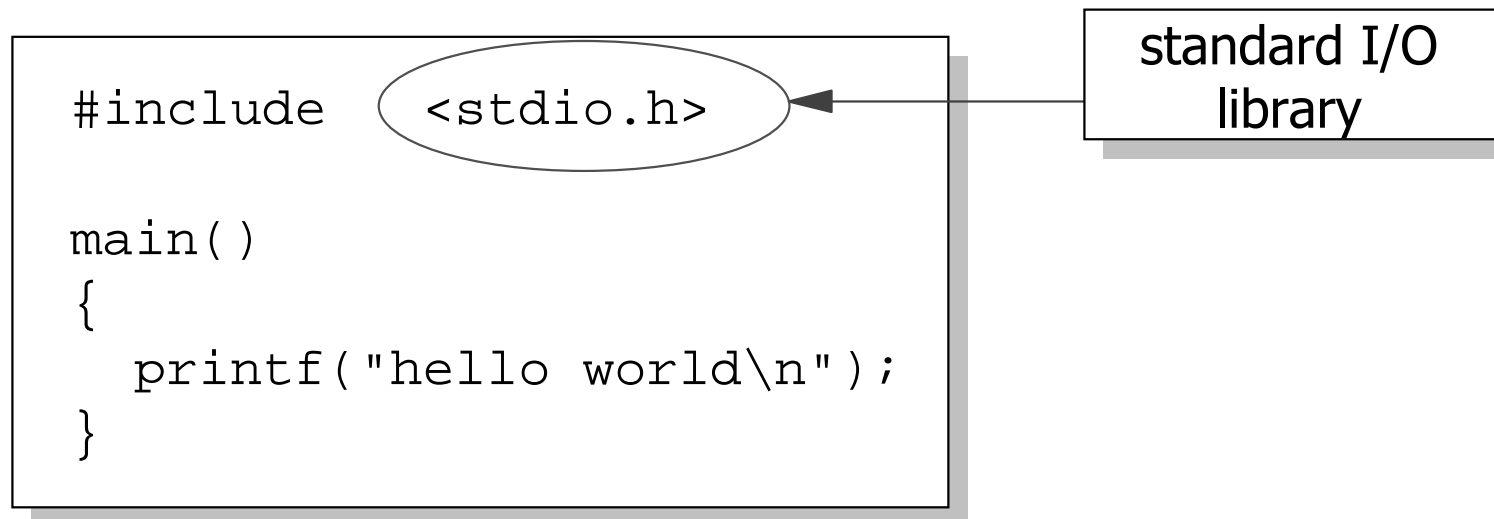
"Hello World"



"Hello World"



"Hello World"

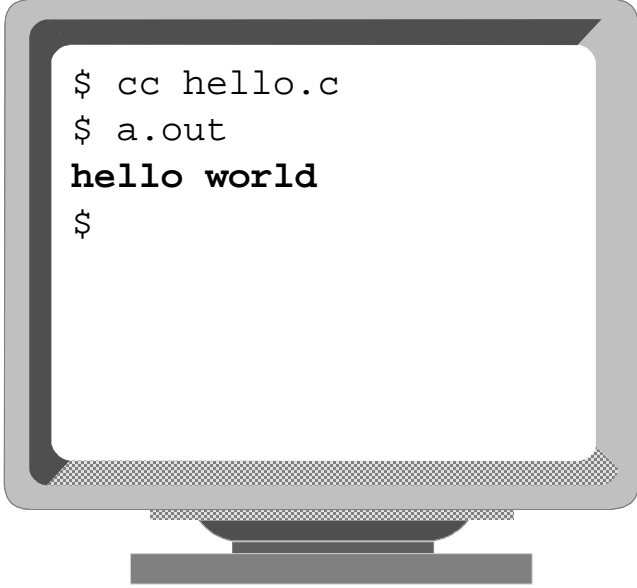


"Hello World"

hello.c

```
#include <stdio.h>

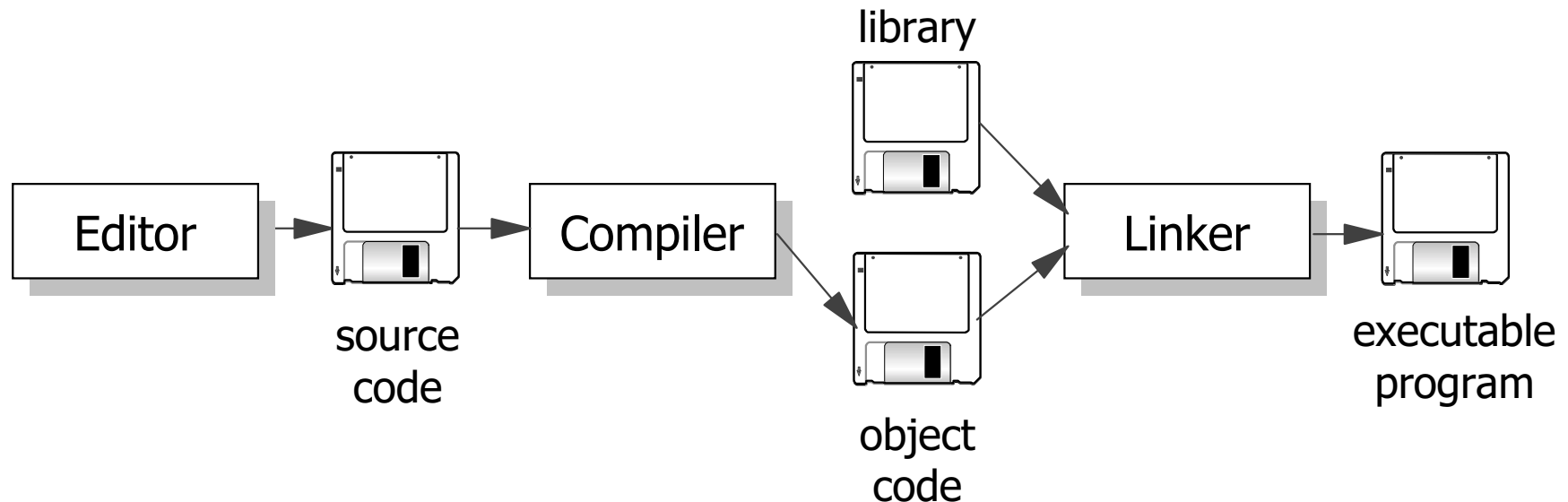
main()
{
    printf("hello world\n");
}
```



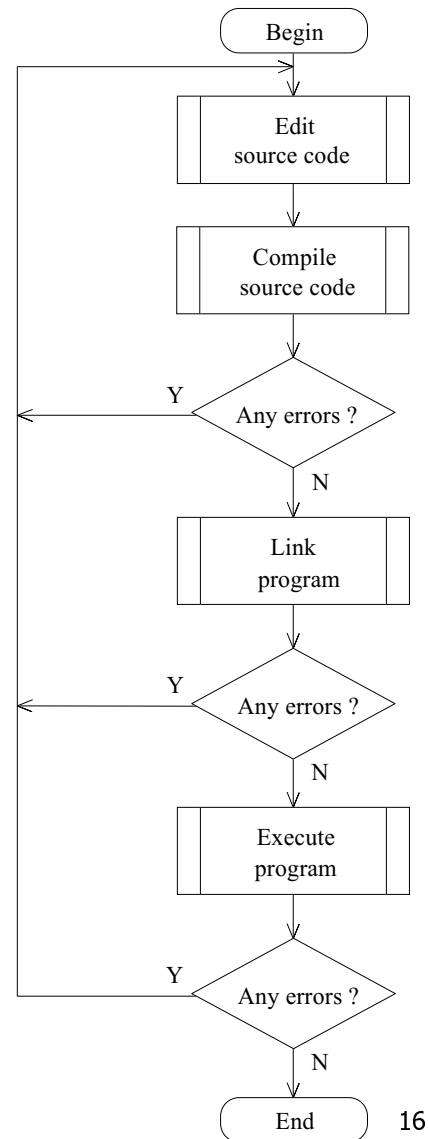
```
$ cc hello.c
$ a.out
hello world
$
```

The Program Development Cycle

- Creating the source code.
- Compiling the source code.
- Linking to create an executable file.
- Executing the program.



C Program Development



HELLO.C with an error

hello.c

```
#include <stdio.h>

main()
{
    printf("hello world\n")
}
```

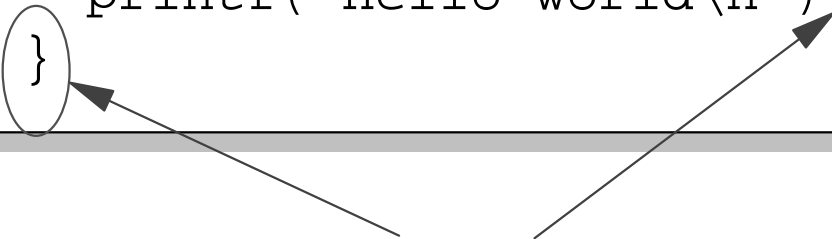
Compilation Errors

HELLO.C with an error

hello.c

```
#include <stdio.h>

main()
{
    printf("hello world\n")
}
```

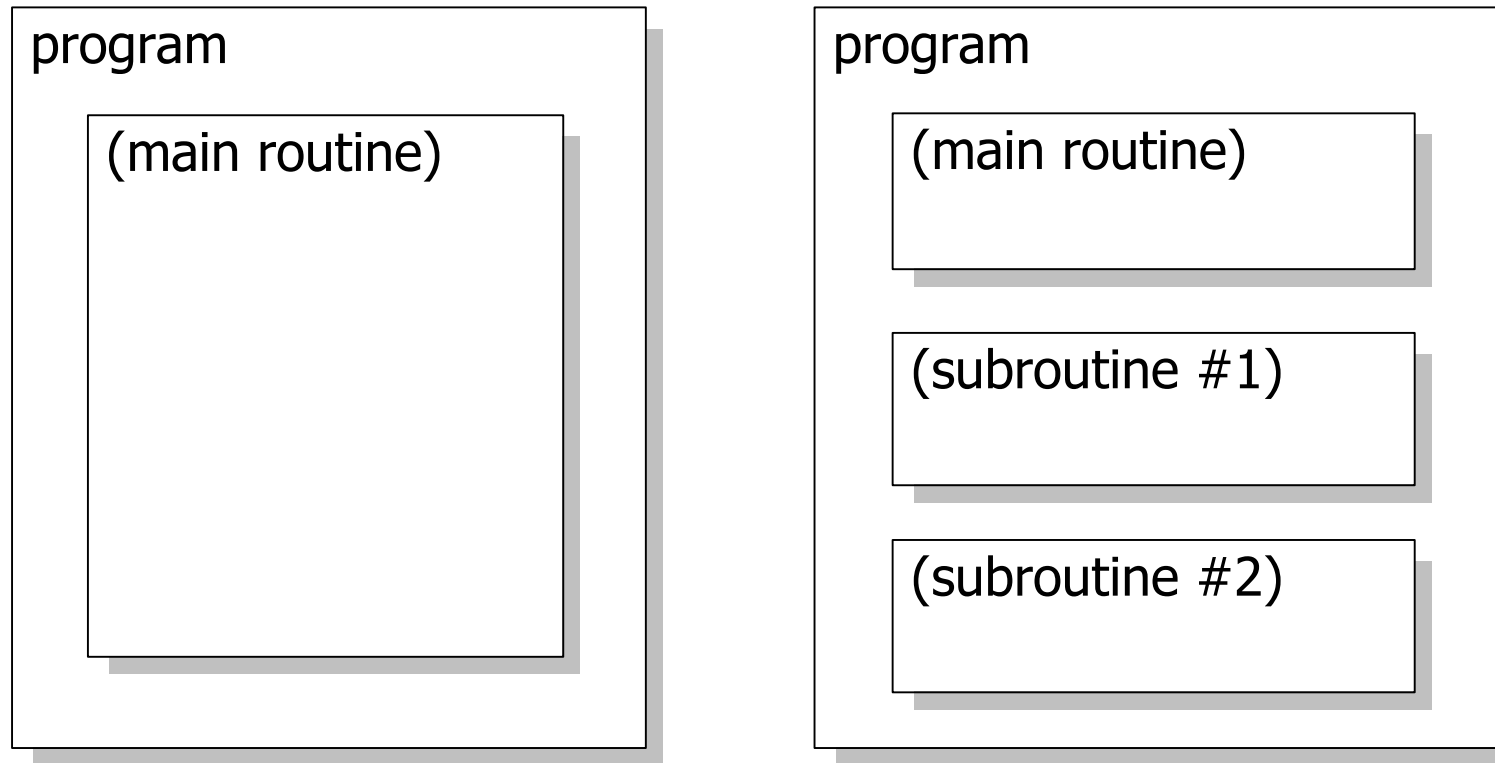


```
hello.c(6):Error: ';' expected
```

Q & A

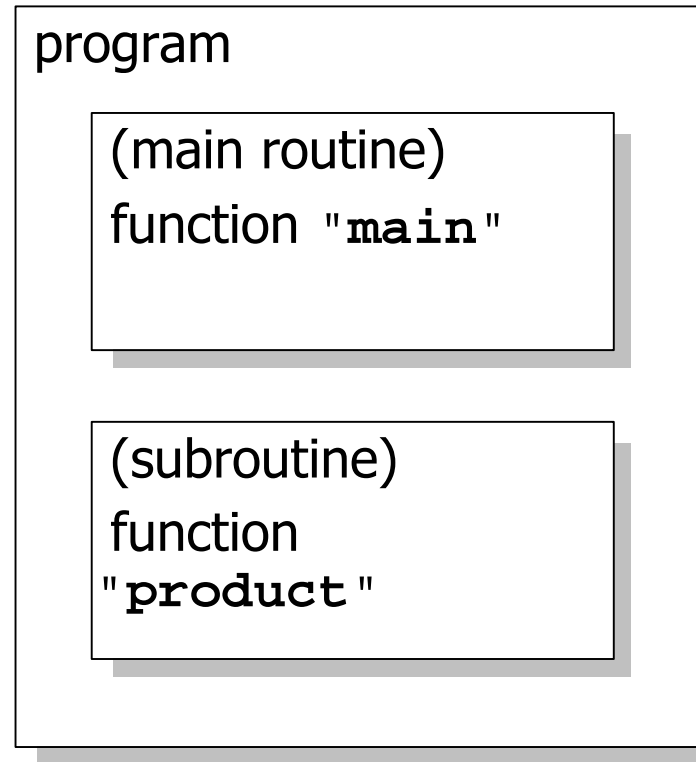
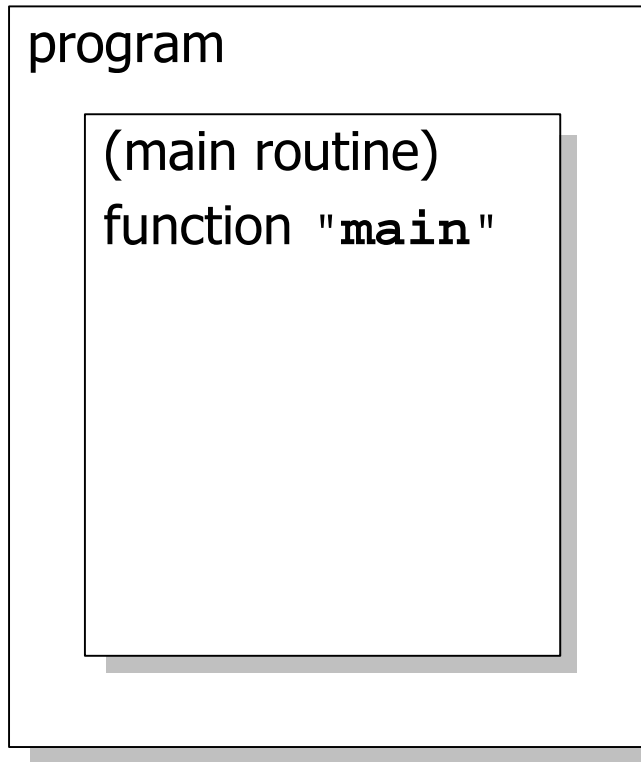
- If I want to give someone a program I wrote, which files do I need to give them ?
- After I create an executable file, do I need to keep the source file (.C) or object file (.OBJ) ?
- If my compiler came with an editor, do I have to use it ?

The Structure of a C Program



Main characteristics of C language is that a C program is organized by main routine and subroutines.

Functions



A *function* is an independent section of program code that performs a certain task and has been assigned a name.

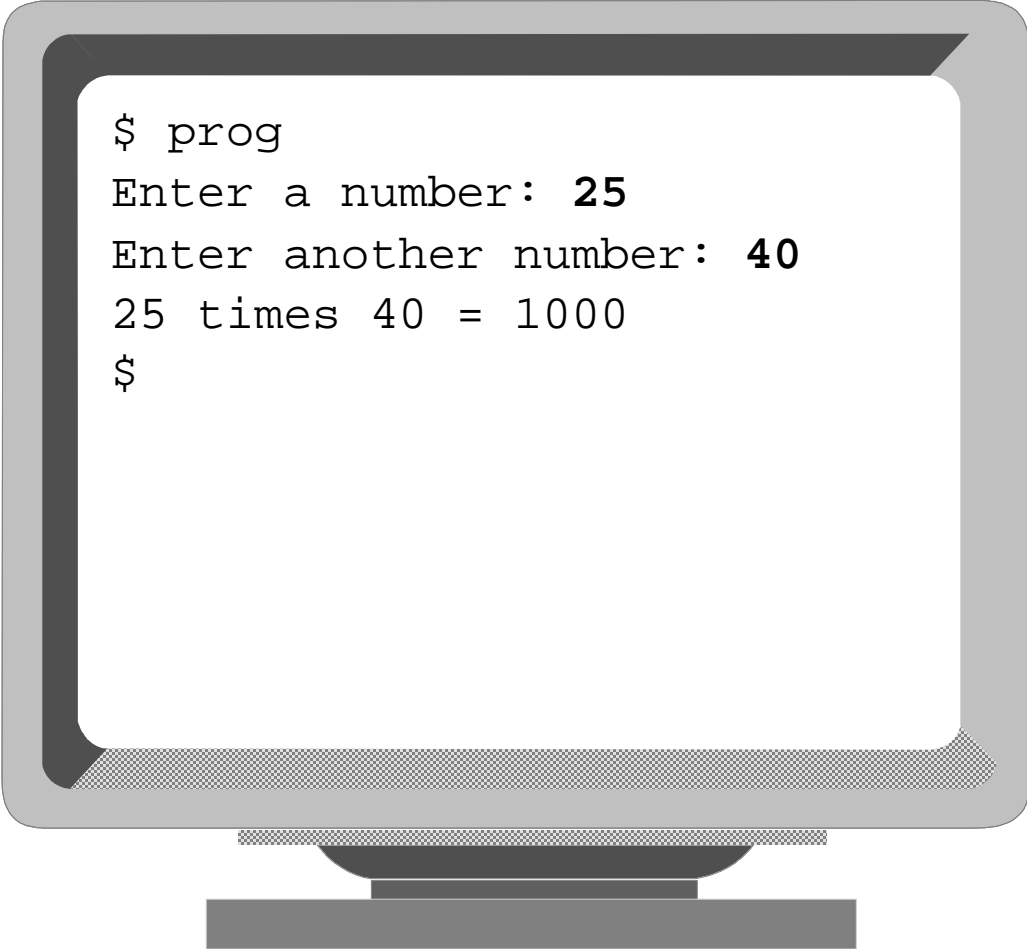
A Short C Program

```
/* Calculate the product of 2 numbers */
#include <stdio.h>

int a, b, c;
int product( int x, int y );

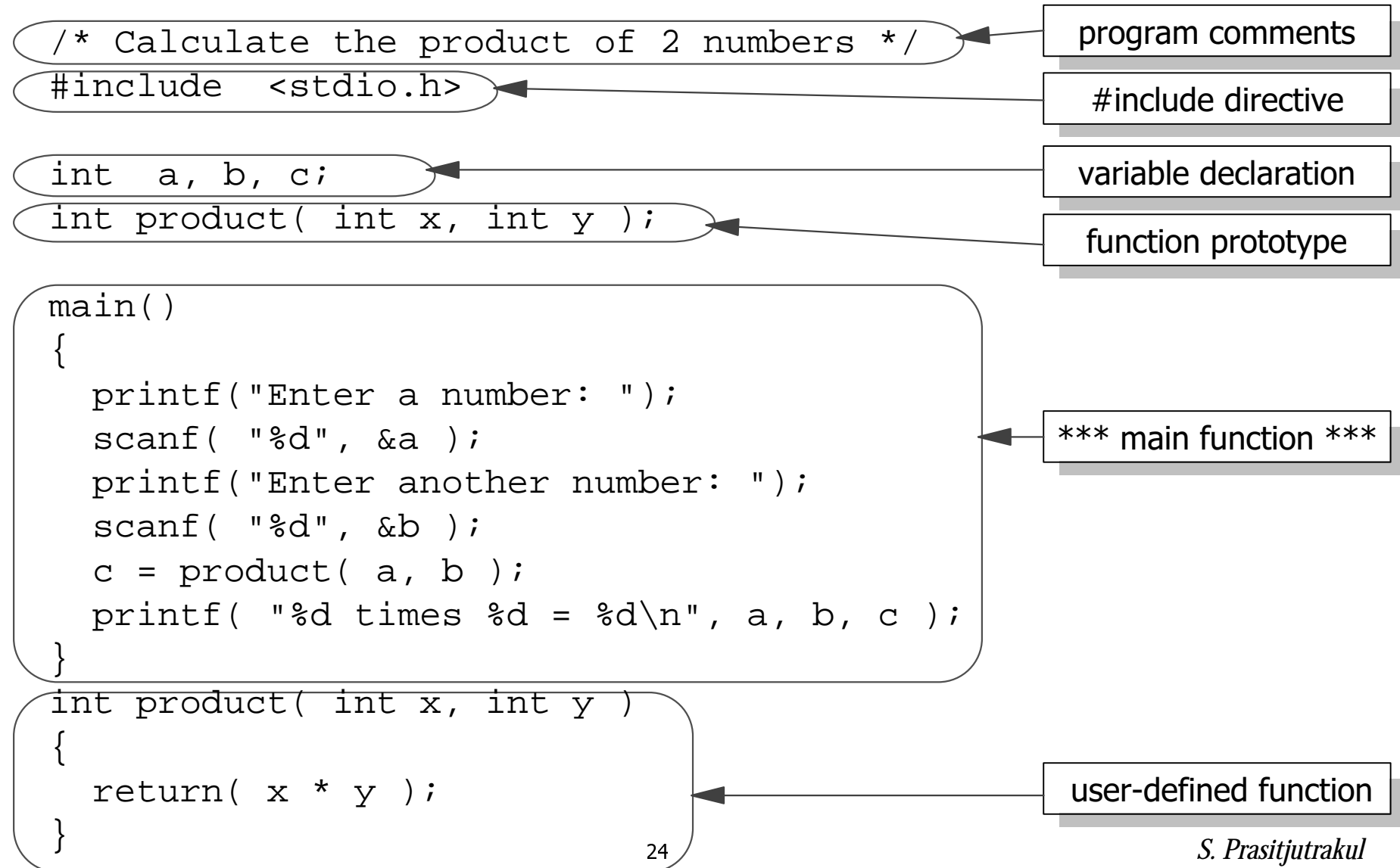
main()
{
    printf("Enter a number: ");
    scanf( "%d", &a );
    printf("Enter another number: ");
    scanf( "%d", &b );
    c = product( a, b );
    printf( "%d times %d = %d\n", a, b, c );
}
int product( int x, int y )
{
    return( x * y );
}
```

A Short C Program



```
$ prog
Enter a number: 25
Enter another number: 40
25 times 40 = 1000
$
```

C Program Components



Statements and Blocks

```
main()  
{  
    printf("Enter another number: ");  
    scanf( "%d", &b );  
    c = product( a, b );  
    printf( "%d times %d = %d\n", a, b, c );  
}
```

statement

```
main()  
{  
    printf("Enter another number: ");  
    scanf( "%d", &b );  
    c = product( a, b );  
    printf( "%d times %d = %d\n", a, b, c );  
}
```

block

Program Comments

- Comments are used to make a program easier to understand.
- Any characters between `/*` and `*/` are ignored by the C compiler.
- A comment can span part of a line, an entire line, or multiple lines.

```
/* A single line comment */  
int    a, b, c;    /* A partial line comment */  
/* a multiple  
   line  
   comment  
*/
```

Do & Don't

- Do add program comments.
- Don't add unnecessary comments.

```
/*
    stereo.c - This program projects a list of 3d objects
               given from stdin to two images for left
               and right eyes.
*/
void stereo( ... )
{
    ...
    /* the following line prints "Done" on the screen */
    printf( "Done\n" );
}
```

Summary

- C is a wise and right choice (powerful, portable, and popular).
- Edit-compile-link-test cycle
- Error messages
- The major components of a C program
 - the `main()` function
 - user-defined functions and library functions
 - statements and blocks
 - program comments