

25 YEARS ANNIVERSARY
SOICT

ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Introduction to C programming language

IT 3210 – C Programming Language

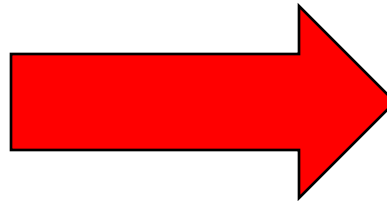
High level language

```
#include <stdio.h>

int main()
{
    printf("Hello World");

    return 0;
}
```

Source code



```
10100110 01110110
00100110 00000000
11111010 11111010
01001110 10100110
11100110 10010110
11001110 00101110
10100110 01001110
11111010 01100110
01001110 10000110
etc...
```

Machine code

- Compiler translates a program in high level programming language into machine language

Why C?

- Flexible language
 - Structured language
 - Low level activities possible
- Standard library exists, allowing portability
- It can produce lean and efficient code
- Wide availability on a variety of computers and widely used
- It is the foundation for other languages (C++, Java, Perl, awk)

History of C

- **CPL** Combined Programming Language (Barron et al., 1963)
- **BCPL** Basic CPL (Richards, 1969)
- **B** (Thompson, 1970)
- **C** K&R C (Ritchie, 1972)
- **ANSI C** American National Standards Institute C (X3J11, 1989)
- **C99** (JTC1/SC22/WG14, ISO/IEC 9899, 1999)

The first C program

Hello World

Algorithm

output “Hello World!”

C program

```
#include <stdio.h>

int main()
{
    printf("Hello World!");

    return 0;
}
```

C Language Structure

- `#include <stdio.h>`
 - To declare using the standard I/O library. Other libraries: string, time, math...
- `int main()`
 - To declare the `main()` function. An C program must declare only one `main()` function. The first line in the `main()` will implement when the program starts.
- `{ ... }`
 - The syntax to open and close a block of codes.
- `printf`
 - the `printf()` function sends the output to standard output (monitor). This function will be taught in the next week.
- `return 0;`
 - Stop the program.

Syntax of C programs

- A C program is written using:
 - Keywords: reserved words for specific meaning in a program, e.g., main, if, do, while, ...
 - User's names: names defined by user to specify a variable, a function, etc. in a program.
 - Specific characters: to represent expressions in a program and make the program have structure, for example:
 - Create a block of instructions { }
 - Create a string “ “

Keywords of C

- Flow control (6) – `if`, `else`, `return`, `switch`, `case`, `default`
- Loops (5) – `for`, `do`, `while`, `break`, `continue`
- Common types (5) – `int`, `float`, `double`, `char`, `void`
- structures (3) – `struct`, `typedef`, `union`
- Counting and sizing things (2) – `enum`, `sizeof`
- Rare but still useful types (7) – `extern`, `signed`, `unsigned`, `long`, `short`, `static`, `const`
- Evil keywords which we avoid (1) – `goto`
- Weirdies (3) – `auto`, `register`, `volatile`

Common characters used in a program

- `{...}` create a block of instructions
- `"..."` create a string to display
- `/* ... */` create a block of comment in the program
- `;` End of an instruction
- other characters for formulas such as `+`, `-`, `*`, `/`, `()`, ...

Identifiers

- When declare a variable or a procedure, we need to identify it
- Principles:
 - Only use alphabetic letters, numbers, underscore _ character to name an identify
 - Identify must begin with an alphabetic letter
 - Upper case and lower case are different
- Which identities are illegal:
 - tong, 2k, trung binh, %totnghiep

Example of writing a program

print number from 0 to 9

```
dem = 0
while (dem < 10)
do
{
    output dem
    dem = dem + 1
}
```

Create the main
function for the
program

```
int main()
{

    return 0;
}
```

Example ...

print number from 0 to 9

```
dem = 0
while (dem < 10)
do
{
    output dem
    dem = dem + 1
}
```

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

```
int main()
{
```

Declare the
standard
input/output
library

```
    return 0;
```

```
}
```

Example ...

print number from 0 to 9

```
dem = 0
while (dem < 10)
do
{
    output dem
    dem = dem + 1
}
```

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

```
/* In tu 0 toi 9 */
```

```
int main()
```

```
{
```

Comment

```
    return 0;
```

```
}
```

Example ...

print number from 0 to 9

```
dem = 0
while (dem < 10)
do
{
    output dem
    dem = dem + 1
}
```

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

```
/* In tu 0 toi 9 */
```

```
int main()
```

```
{
```

```
    int dem;
```

Declare a
variable

```
    return 0;
```

```
}
```

Example ...

print number from 0 to 9

```
dem = 0
while (dem < 10)
do
{
    output dem
    dem = dem + 1
}
```

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

```
/* In tu 0 toi 9 */
```

```
int main()
```

```
{
```

```
    int dem;
```

```
    dem = 0;
```

Assign a
value to dem

```
    return 0;
```

```
}
```


Example ...

print number from 0 to 9

```
dem = 0
while (dem < 10)
do
{
    output dem
    dem = dem + 1
}
```

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

```
/* In tu 0 toi 9 */
```

```
int main()
```

```
{
```

```
    int dem;
```

```
    dem = 0;
```

```
    while ( dem < 10 )
```

```
    {
```

```
    }
```

```
    return 0;
```

```
}
```

Creare a loop

Example ...

print number from 0 to 9

```
dem = 0
while (dem < 10)
do
{
    output dem
    dem = dem + 1
}
```

```
#include <stdio.h>

/* In tu 0 toi 9 */
int main()
{
    int dem;
    dem = 0;
    while ( dem < 10 )
    {
        printf("%d\n", dem) ;

    }
    return 0;
}
```

Example ...

print number from 0 to 9

```
dem = 0
while (dem < 10)
do
{
    output dem
    dem = dem + 1
}
```

```
#include <stdio.h>

/* In tu 0 toi 9 */
int main()
{
    int dem;
    dem = 0;
    while ( dem < 10 )
    {
        printf("%d\n", dem) ;
        dem = dem + 1;
    }
    return 0;
}
```

What does this program do?

```
#include <stdio.h>

int main(){
    float num;

    printf("Enter a number: ");
    scanf("%f", &num);

    if ( num < 0 ) {
        printf("%f is negative", num);
    } else {
        printf("%f is positive", num);
    }

    return 0;
}
```

Algorithm of the program

/ Find the sign of a number */*

input num

if (num < 0) then

{

 output “negative number”

}

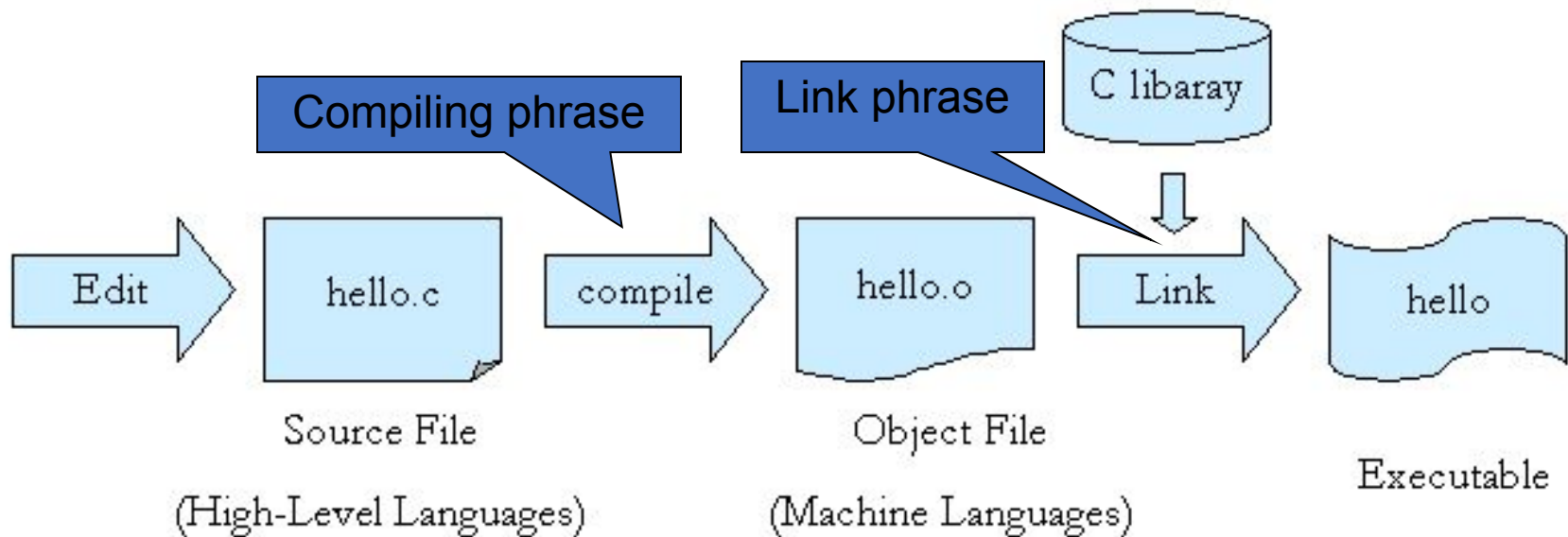
else

{

 output “positive number”

}

Compile a C program



Error can appear at compiling phrase or link phrase

Compiler

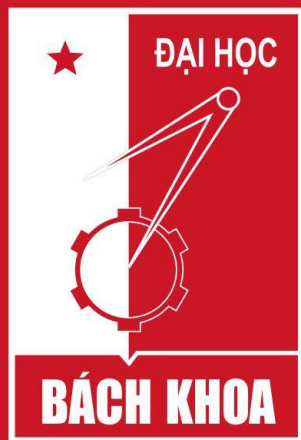
- To translate a program, we need a compiler, for example: gcc
- The compiler of C always supports parameter to perform two phrases of the compiling process. For example, gcc -c to do the compiling phrase, gcc -o to do the link phrase.
- If your program is written in only one file, a single Unix command can help to make an executable program from the source code.
 - \$ gcc -o *program-name filename*
 - Ex: \$gcc -o hello hello.c

IDE: Integrated Development Environment

- Programming is a process that repeatedly carries out operations: source code editing, compiling, debugging.
- These operations can carry out independently by different tools. For example, edit by emacs, compile by gcc.
- However, a more convenient way is to integrate all programming tools to an unique environment to support the programming. This environment is called IDE.
- IDE - an environment of 3 in 1: editor, compiler, debugger

Products of IDE

- On Linux:
 - KDevelop
- On Window:
 - Dev-C++,
 - Turbo C++,
 - Visual C++,
 - etc.



25 YEARS ANNIVERSARY
SOICT

HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY



25 YEARS ANNIVERSARY
SOICT

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

**Thank you
for your
attentions!**



soict.hust.edu.vn/



fb.com/groups/soict

