Jiangxi University of Science and Technology

Ch01 Introduction to Computer Programming

Lecture0101 introduction / History and Programming Languages

History and Hardware

JOIN NOW

DrAjm(阿 塔)

Jiangxi Ganzhou

MY WECHAT ID

2019lecture

THE C PROGRAMMING LANGUAGE

Our Lecture WE CHAT ID

1.1 History and Hardware

THE C PROGRAMMING LANGUAGE

➢**ENIAC** (/ˈini.æk/ or /ˈɛni.æk/)

- It **Electronic Numerical Integrator And Computer** was the first electronic general-purpose computer.

- was Turing-complete, digital, and capable of being reprogrammed to solve "a large class of numerical problems"

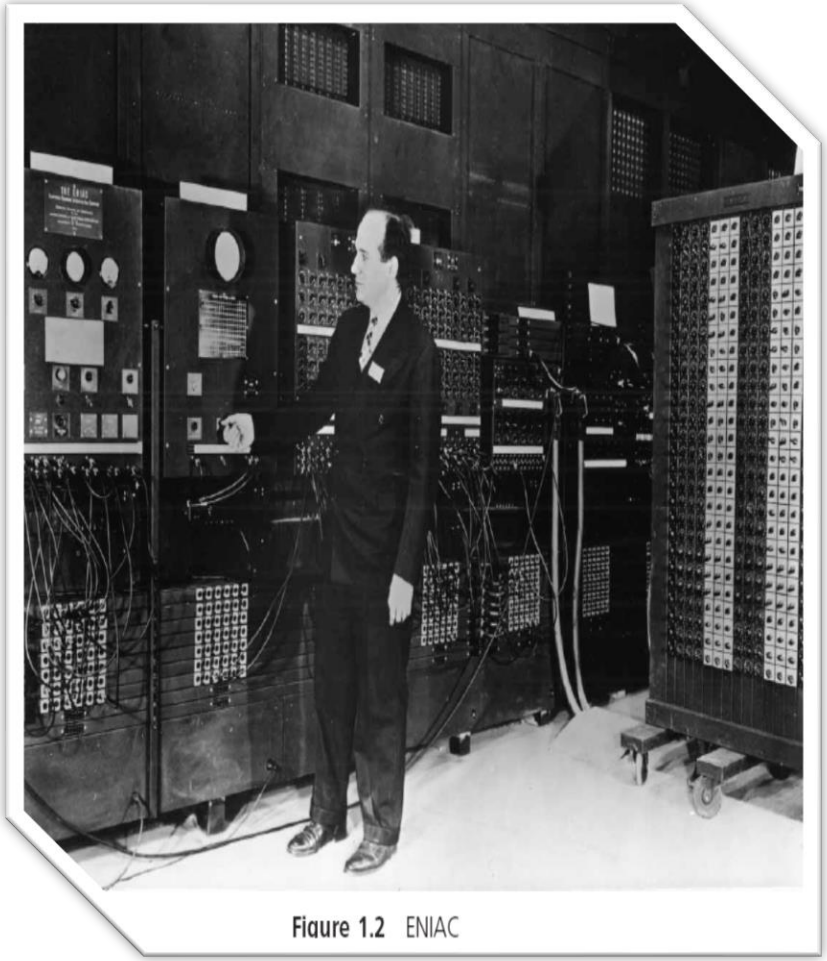


Figure 1.2 ENIAC

1.1 History and Hardware

THE C PROGRAMMING LANGUAGE

➢ Computer Hardware(计算机硬件)

- Computers are constructed from physical components referred to as **hardware**
- **Hardware** facilitates the storage and processing of data under the direction of a stored program
- Computer hardware does not store data using the same symbols that humans do

1.1 History and Hardware

THE C PROGRAMMING LANGUAGE

➢ **Bits** (位) and **Bytes** (字节)

- The smallest and most basic data item in a computer is a **bit**: 0 or 1
- The grouping of 8 bits to form a larger unit is referred to as a **byte**
  - Can represent any one of 256 distinct patterns
- The collections of patterns consisting of 0s and 1s used to represent letters, single digits, and other single characters are called **character codes**

1.1 History and Hardware

THE C PROGRAMMING LANGUAGE

- **Components (组件)**

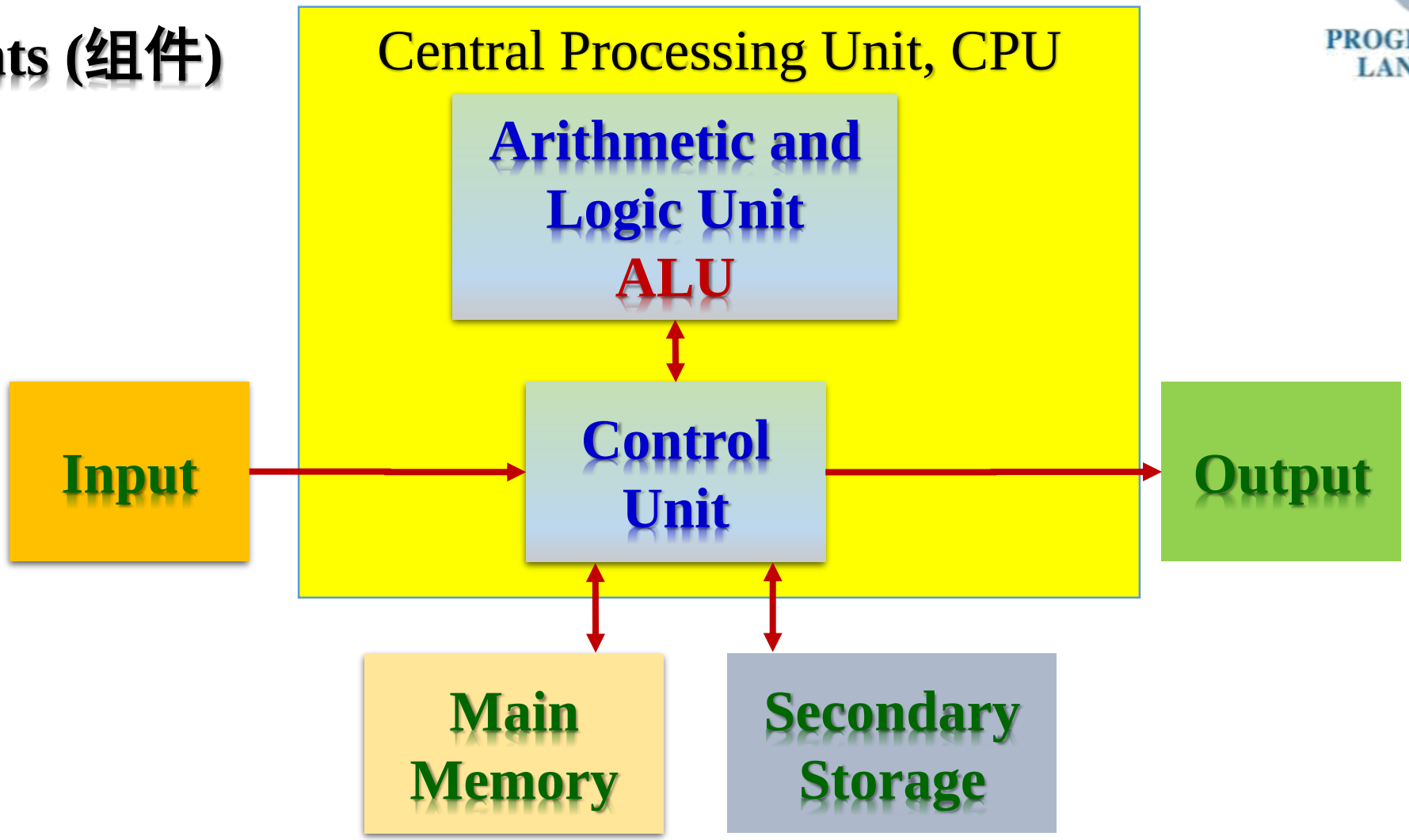


Figure 1.4 Basic Hardware Unit of a Computer

1.1 History and Hardware

THE C PROGRAMMING LANGUAGE

Central Processing Unit (CPU, 中央处理器)

- **Control unit (控制单元):**
  - directs and monitors the overall operation of the computer
- **Arithmetic and Logic Unit (ALU, 算术逻辑单元)**
  - performs all of the computations, such as addition, subtraction, comparisons, and so on
- **Microprocessor (微处理器)**
  - CPUs are constructed as a single microchip, which is referred to as a microprocessor

1.1 History and Hardware

THE C PROGRAMMING LANGUAGE

- **Microprocessor**

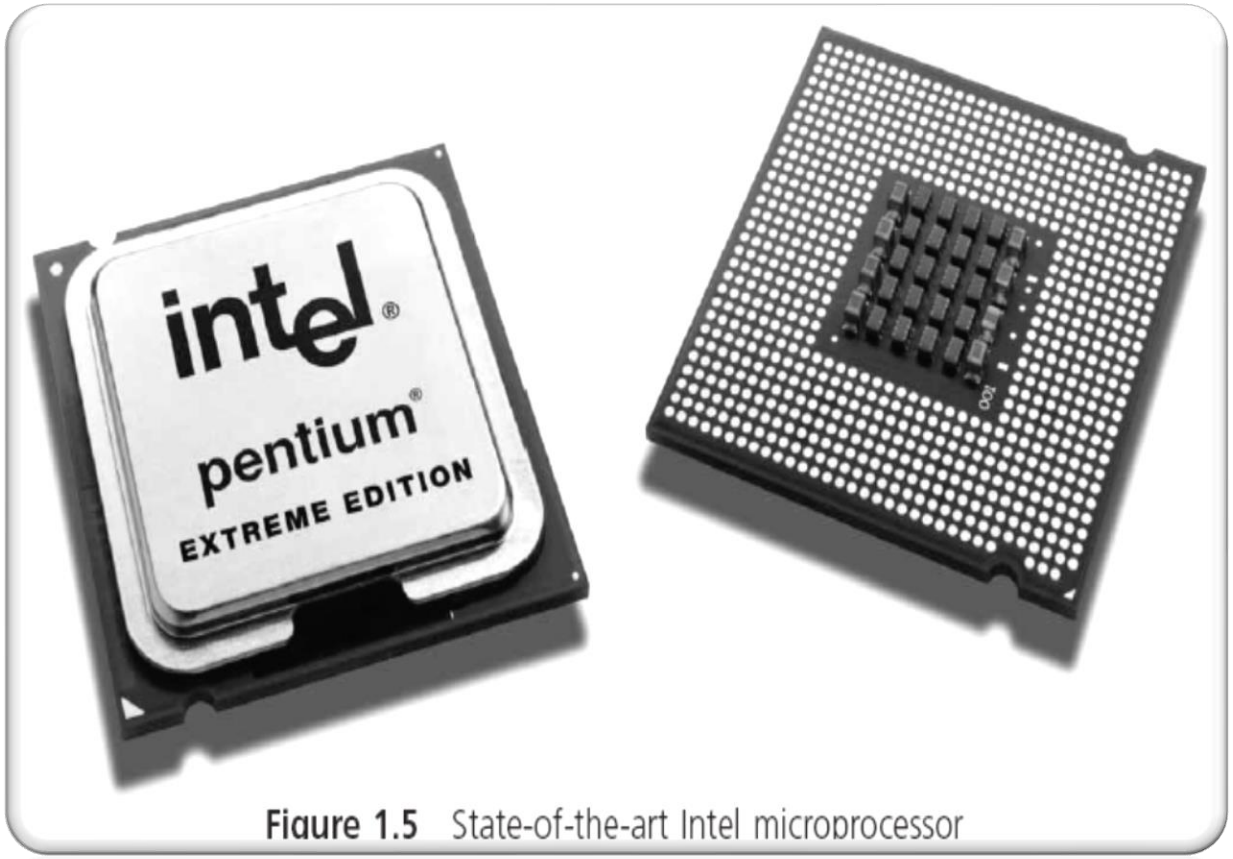


Figure 1.5 State-of-the-art Intel microprocessor

Figure 1.5 State of the art Intel Microprocessor

1.1 History and Hardware

THE C PROGRAMMING LANGUAGE

➢ Main Memory Unit (主存单元)

- Stores data and instructions as sequence of bytes
- A program must reside in main memory if it is to operate on the computer
- Combines 1 or more bytes into a single unit, referred to as a **word (字)**
- Constructed as random access memory, or **RAM 随机存储器**
  - Every section of memory can be accessed randomly as quickly as any other section
  - **Volatile (易失性):** data is lost when power is turned off
- **Size** is usually specified in **bytes** (**MB** or **GB**)

1.1 History and Hardware

THE C PROGRAMMING LANGUAGE

➢Input/output Unit (输入输出单元)

- The input/output(I/O) unit provides access to the computer, allowing it to input and output data
- It is the **interface接口** to which **peripheral devices** (外围设备), such as keyboards, console screens, and printers, are attached

➢Secondary Storage (辅助存储器)

- Used as permanent storage for programs and data, such as magnetic tape, magnetic disks, and CD-ROMs
- **Direct access storage device (DASD, 直接存取存储器)**
  - allows a computer to read or write any one file or program independent of its position on the storage medium

1.2 Programming Languages

- **Computer program (计算机程序):**
  - data and instructions used to operate a computer and produce a specific result
  - A program or set of programs is called **software**
- **Programming (编程):**
  - writing instructions in a language that the computer can respond to and that other programmers can understand
- **Programming language (编程语言):**
  - set of instructions that can be used to construct a program

1.2 Programming Languages

THE C PROGRAMMING LANGUAGE

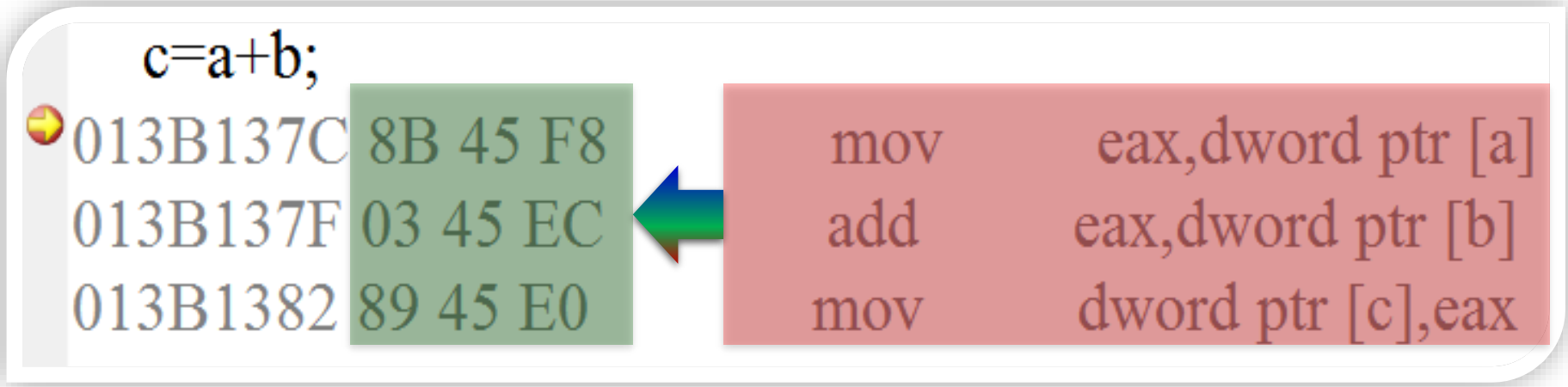
➢Machine Language(机器语言)

- **machine language** is a set of instructions executed directly by a computer's central processing unit (CPU).
- Each **instruction(指令)** performs a very specific task, such as a load, a jump, or an ALU operation.

```
    c=a+b;
013B137C 8B 45 F8     mov     eax,dword ptr [a]
013B137F 03 45 EC     add     eax,dword ptr [b]
013B1382 89 45 E0     mov     dword ptr [c],eax
```

1.2 Programming Languages

- **Assembly language (汇编语言):**
  - Assembly language uses a mnemonic to represent each low-level machine instruction or operation.
  - Typical operations require one or more operands in order to form a complete instruction
  - Assembly program must be translated into machine language program.



1.2 Programming Languages

➢Low- and High-Level Languages

- Machine and assembly languages are **low-level languages (低级语言)** because they both use instructions that are directly tied to one type of computer
- **a high-level programming language (高级语言)** is a programming language with strong abstraction from the details of the computer.
- it may use natural language elements, be easier to use.
- Such as Fortran, **C**, **C++**, Java and so on

1.2 Programming Languages

➢Low- and High-Level Languages

- **Interpreter (解释器)**:
  - program that translates each statement in a high-level source program and executes it immediately upon translation
- **Compiler (编译器)**:
  - translates a high-level source program as a complete unit before any statement is executed
  - The output produced by the compiler is called an **object program (目标程序)**
- **Linker (连接器)**:
  - combines additional machine language code with the object program to create a final **executable program (可执行程序)**

1.2 Programming Languages

THE C PROGRAMMING LANGUAGE

- Low- and High-Level Languages

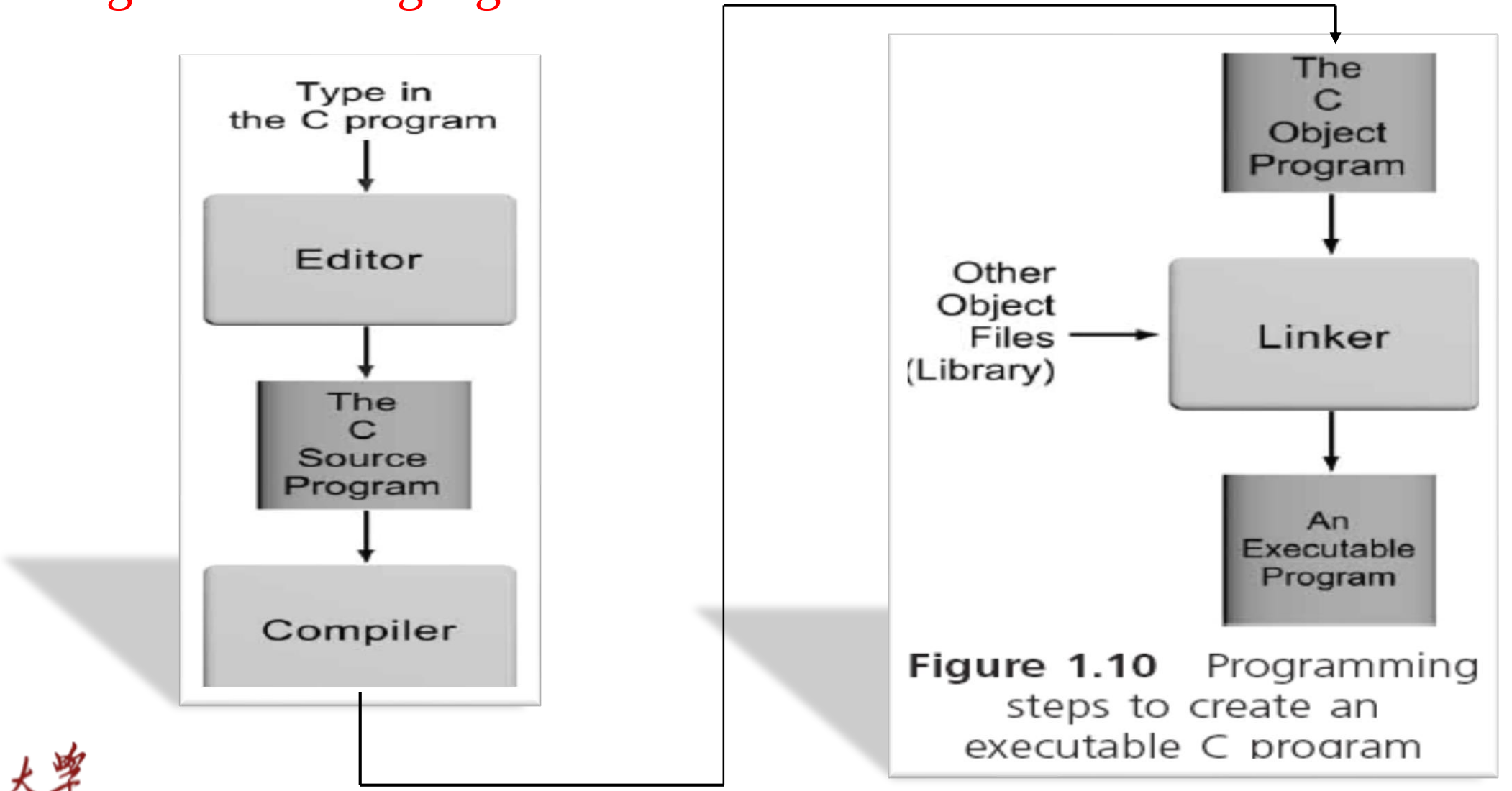


Figure 1.10 Programming steps to create an executable C program

1.2 Programming Languages

- **Structured language** (结构化语言):
  - high-level procedural language (e.g., C) that enforces structured procedures
- **Object-oriented languages** (面向对象语言):
  - languages with object orientation such as C++, Java, Visual Basic, and C#

1.2 Programming Languages

- **Application software (应用软件) :**
  - programs written to perform particular tasks required by users

- **System Software (系统软件):**
  - collection of programs that must be readily available to any computer system to enable the computer to operate

- **Operating System (操作系统):**
  - set of system programs used to operate and control a computer

Jiangxi University of Science and Technology

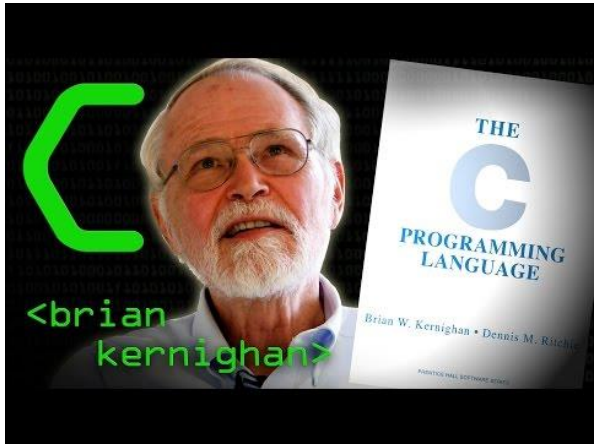
Ch01 :Introduction to Computer Programming

Lecture0102 Algorithm and Software Development

1.2 Programming Languages

➢The Development of C

- **Developed in the 1970s at AT&T Bell Laboratories by K. Thompson, D. Ritchie, and B. Kernighan**



- **Standard maintained by the American National Standards Institute (ANSI) ['ænsi]**

- **In the 1980s, Bjarne Stroustrup developed C++, C with object-oriented capabilities**

1.3 Algorithms

THE C PROGRAMMING LANGUAGE

- Algorithm (算法):
  - Specific steps required to produce a desired result

1. Set n equal to 100

2. Set a equal to 1

3. Set b equal to 100

4. Calculate sum = n(a+ b)/2

5. Display the sum

1.3 Algorithms

THE C PROGRAMMING LANGUAGE

➢Pseudocode (伪代码)

—When English phrases are used to describe an algorithm, the description is called pseudocode

1. *Input the three numbers into the computer*
2. *Calculate the average by adding the numbers and dividing the sum by three*
3. *Display the average*

1.3 Algorithms

THE C PROGRAMMING LANGUAGE

- Formula (公式)
  - When mathematical equations are used to describe an algorithm, the description is called a formula

- Flowchart (流程图)
  - A flowchart is a type of diagram that represents an algorithm showing the steps as boxes of various kinds, and their order by connecting them with arrows.

1.3 Algorithms

THE C PROGRAMMING LANGUAGE

➢Flowchart

1. Terminal

- Indicates the beginning or end of an algorithm

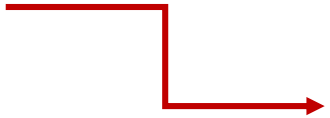
2. Input/ouput

- Indicates an input or output operation

3. Process

- indicates computation or data manipulation

4. Flow Lines



- connects the flowchart symbols and indicates logic flow

1.3 Algorithms

THE C PROGRAMMING LANGUAGE

➢Flowchart

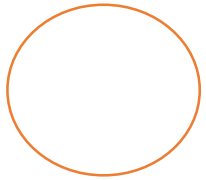
5. Loop

- indicates the initial, final, and increment values of a loop

6. Predefined process

- Indicates a predefined process, as in calling a sorting process

7. Connector

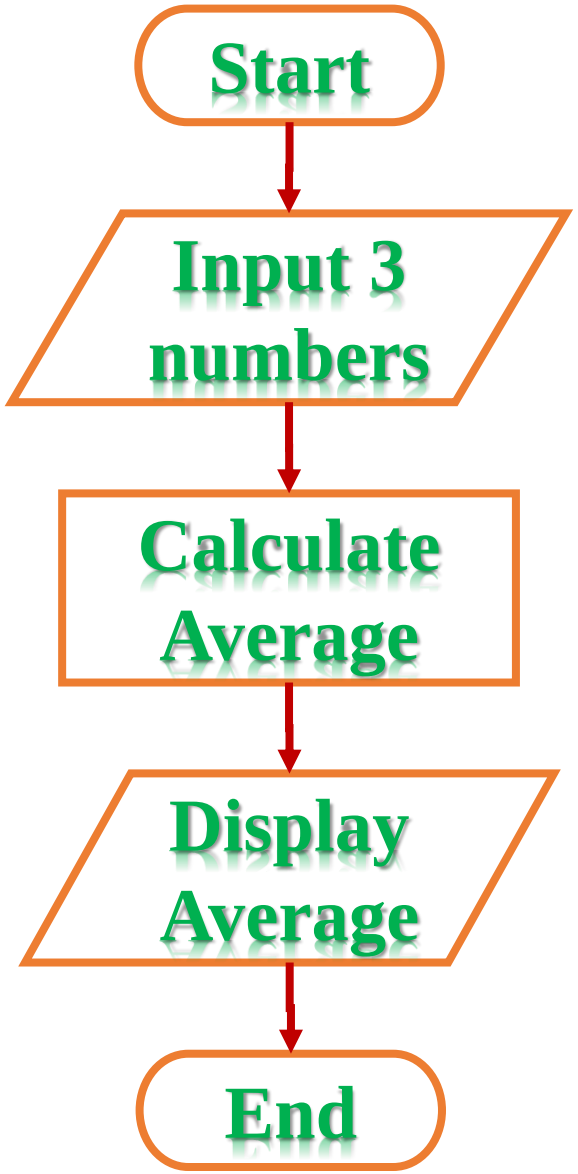


- Indicates an entry to or exit from another part of the flowchart

1.3 Algorithms

THE C PROGRAMMING LANGUAGE

➢Flowchart



1.3 Algorithms

THE C PROGRAMMING LANGUAGE

- **coding the algorithm (编码算法)**
  - Converting an algorithm into a computer program, using a language such as C, is called **coding the algorithm**
  - The program instructions resulting from coding an algorithm are called **program code**, or simply **code**

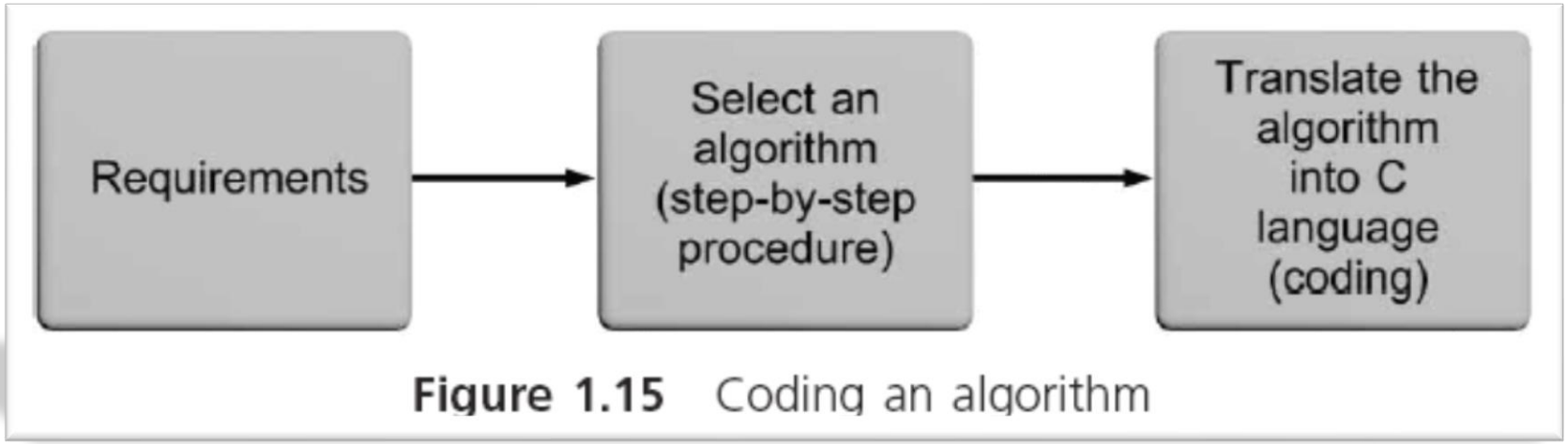


Figure 1.15 Coding an algorithm

1.4 The Software Development Process

THE C PROGRAMMING LANGUAGE

- 1.4 软件开发过程

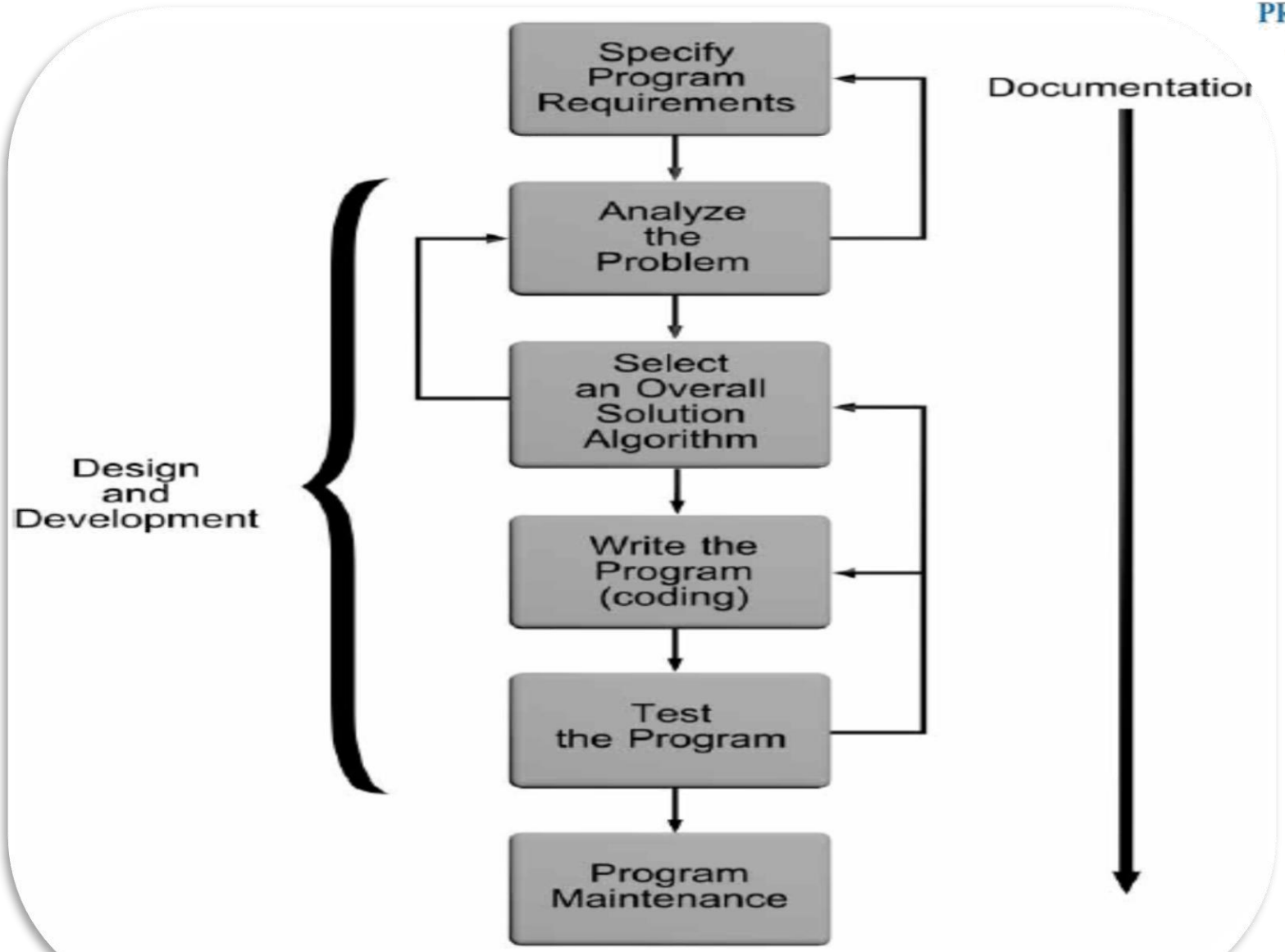
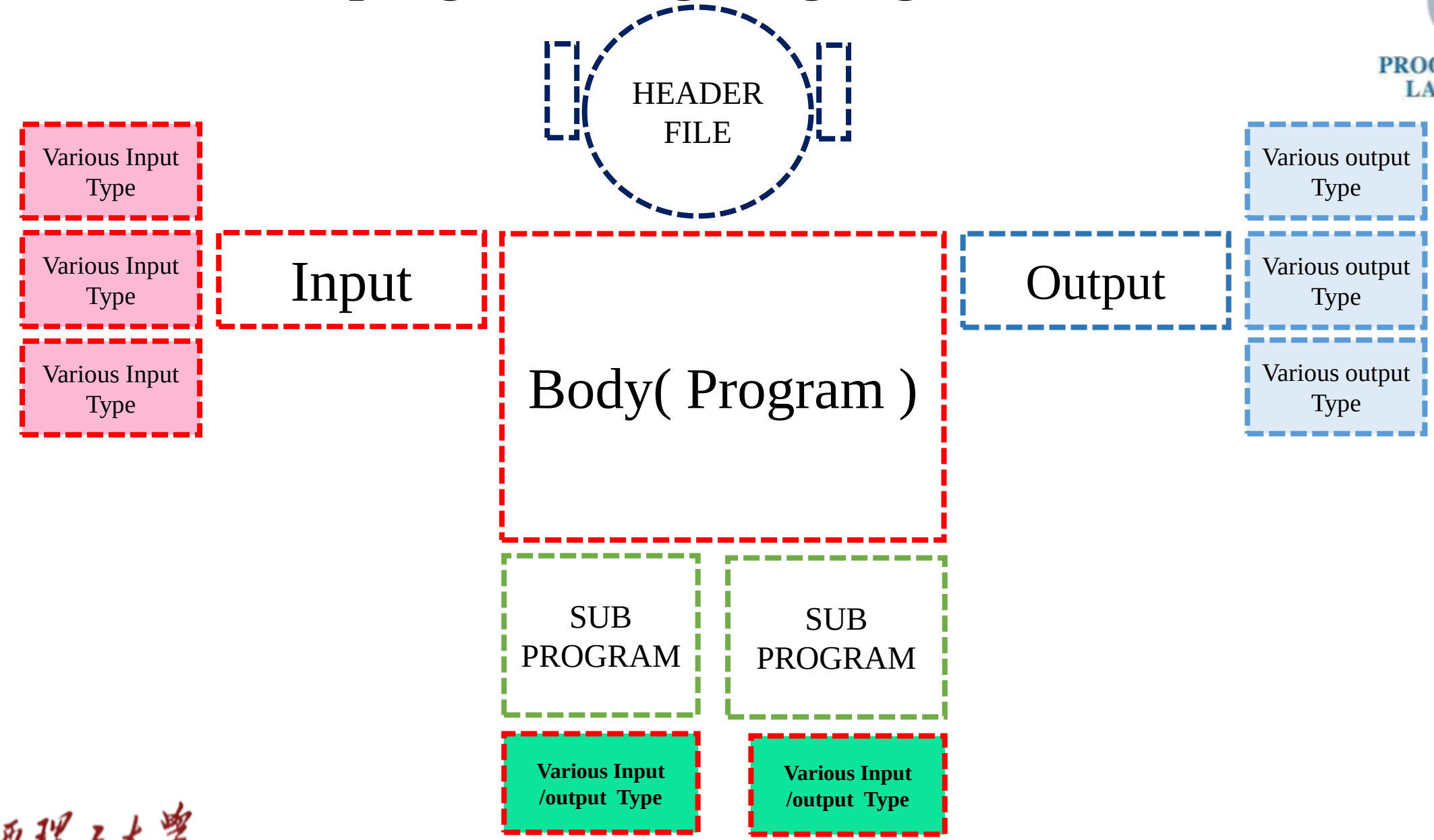


Figure 1.16 The software development process

My idea about all programming language

THE C PROGRAMMING LANGUAGE

My idea about all programming language



First c program: **hello world**

THE C PROGRAMMING LANGUAGE

With this line of code we include a file called stdio.h. (Standard Input/output header file). This file lets us use certain commands for input or output which we can use in our program. (Look at it as lines of code commands) that have been written for us by someone else). For instance it has commands for input like reading from the keyboard and output commands like printing things on the screen.

The int is what is called the return value (in this case of the type integer). Where it used for will be explained further down. Every program must have a main(). It is the starting point of every program. The round brackets are there for a reason, in a later tutorial it will be explained, but for now it is enough to know that they have to be there.

The two curly brackets (one in the beginning and one at the end) are used to group all commands together. In this case all the commands between the two curly brackets belong to main(). The curly brackets are often used in the C language to group commands together. (To mark the beginning and end of a group or function.).

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

First c program: hello world

THE C PROGRAMMING LANGUAGE

The printf is used for printing things on the screen, in this case the words: Hello World. As you can see the data that is to be printed is put inside round brackets.

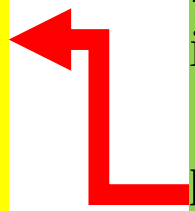
The words Hello World are inside inverted ommas, because they are what is called a string. (A single letter is called a character and a series of characters is called a string). Strings must always be put between inverted commas. The \n is called an escape sequence. In this case it represents a newline character. After printing something to the screen you usually want to print something on the next line.

If there is no \n then a next printf command will print the string on the same line.

Commonly used escape sequences are:

- **\n (newline)**
- **\t (tab)**
- **\v (vertical tab)**
- **\f (new page)**
- **\b (backspace)**
- **\r (carriage return)**

After the last round bracket there must be a semi-colon. The semi-colon shows that it is the end of the command. (So in the future, don't forget to put a semi-colon if a command ended).



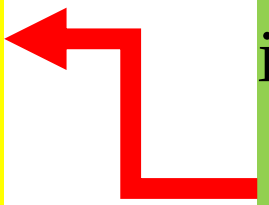
```
#include<stdio.h>
int main()
 {
printf("Hello World\n");
return 0;
}
```

First c program: hello world

When we wrote the first line “int main()”, we declared that main must return an integer int main().

(int is short for integer which is another word for number).

With the command return 0; we can return the value null to the operating system. When you return with a zero you tell the operating system that there were no errors while running the program.



```
#include<stdio.h>
int main()
 {
printf(“Hello World\n”);
return 0;
}
```

Congratulations!!!!!

You have just made your first program in C.

Some point to have the better code

Comments in your program

- The Hello World program is a small program that is easy to understand. But a program can contain thousands of lines of code and can be so complex that it is hard for us to understand. To make our lives easier it is possible to write an explanation or comment in a program.
- This makes it much easier to understand the code. (Even if you did not look at the code for years).These comments will be ignored by the compiler at compilation time.
- Comments have to be put after // or be placed between /* */ .

```
/* Description : Print Hello
World on the screen
Author : Your name
Date : 01/01/2018 */

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

Here is an example of how to comment the Hello World source code :

Some point to have the better code

Indentation

- As you can see the printf and return statements have been indented or moved to the right side. This is done to make the code more readable. In a program as Hello World, it seems a stupid thing to do. But as the programs become more complex, you will see that it makes the code more readable.
- So, always use indentations and comments to make the code more readable. It will make your life much easier if the code becomes more complex.

```
/* Description : Print Hello World on the screen
Author : Your name
Date : 01/01/2007 */
#include<stdio.h>
int main()
        {
           //Print something and then newline
              printf("Hello World\n");
              return 0;
         }
```

Here is an example of how to comment the Hello World source code :

1.5 Case Study-Design and Development

THE C PROGRAMMING LANGUAGE

➢Problem

The circumference, C, of a circle is given by the formula C = 2πr, where π is the constant 3.1416, and r is the radius of the circle.

Using this information, write a C program to calculate the circumference of a circle that has a 2-inch radius.

➢Step 1: Analyze the problem

1. Determine the desired outputs
   - To calculate the circumference of a circle.
2. Determine the input items
   - The input item is the radius of a circle.
3. List the formulas relating the inputs to the outputs
   - C=2*π*r, where C is the output item and r is the input item.
4. Perform a hand calculation

1.5 Case Study-Design and Development

THE C PROGRAMMING LANGUAGE

➢Step 2: Select the algorithm

1. Get the inputs to the problem
2. Calculate the desired outputs
3. Report the results of the calculation.

– for determining the circumference of a circle, this algorithm becomes.

1. Set the radius value to 2
2. Calculate the circumference, using the formula C = 2 π r
3. Display the calculated value for C

1.5 Case Study-Design and Development

THE C PROGRAMMING LANGUAGE

➢Step 3: Write the program

Program 1.1

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.          float radius, circumference;
5.          radius=2;
6.          circumference=2*3.14156*radius;
7.          printf("the circumference of the circle is %f\n", circumference);
8.          return 0;
9.  }
```

1.5 Case Study-Design and Development

THE C PROGRAMMING LANGUAGE

➢ Step 4: Test and correct the program

```
1   #include <stdio.h>
2   int main()
3   {
4       float radius, circumference;
5       radius=2;
6       circumference=2*3.14156*radius;
7       printf("the circumference of the circle is %f\n",
8           circumference);
9       return 0;
10  }
```

236 %

C:\Users\Chengtian Ouyang\Documents\Visual Studio 2010\Projects\qq\Debug\qq.exe

1.7 Summary

1. Electronic Numerical Integrator And Computer(**ENIC**) was the first electronic general-purpose computer. 2
2. The physical components used in constructing a computer are called **hardware**
3. The programs used to operate a computer are referred to as **software**
4. **Programming languages** come in a variety of forms and types
5. **Compiler** and **interpreter languages** are referred to as high-level languages

1.7 Summary

7. **Algorithm**: step-by-step sequence of instructions that must terminate and describes how to perform an operation to produce a desired output
8. **The software development procedure** consists of the following four phases:
   A. Specification of the program's requirements
   B. Design and development
   C. Documentation
   D. Maintenance

1.7 Summary

THE C PROGRAMMING LANGUAGE

7. Steps of the design and development phase are:

A. Analyze the problem
B. Select an overall solution algorithm
C. Write the program
D. Test and correct the program

8. Writing a program consists of translating the solution algorithm into a computer language

9. Fundamental programming **control structures**

—**Sequence**, **selection**, **iteration** and **invocation**

10. You always need at least one backup of a program

5. Summary

1. The physical components used in constructing a computer are called **hardware**
2. The programs used to operate a computer are referred to as **software**
3. Programming languages come in a variety of forms and types
4. **Algorithm**: step-by-step sequence of instructions that must terminate and describes how to perform an operation to produce a desired output
5. The **software development (软件开发)** procedure consists of the four phases.

Reference

- BOOK
- Some part of this PPT given by Prof 欧阳城添
  (**Prof: Chengtian Ouyang**)
- with special thank
- https://www.codingunit.com/c-tutorial-first-c-program-hello-world