1. C语言简介

冯洋

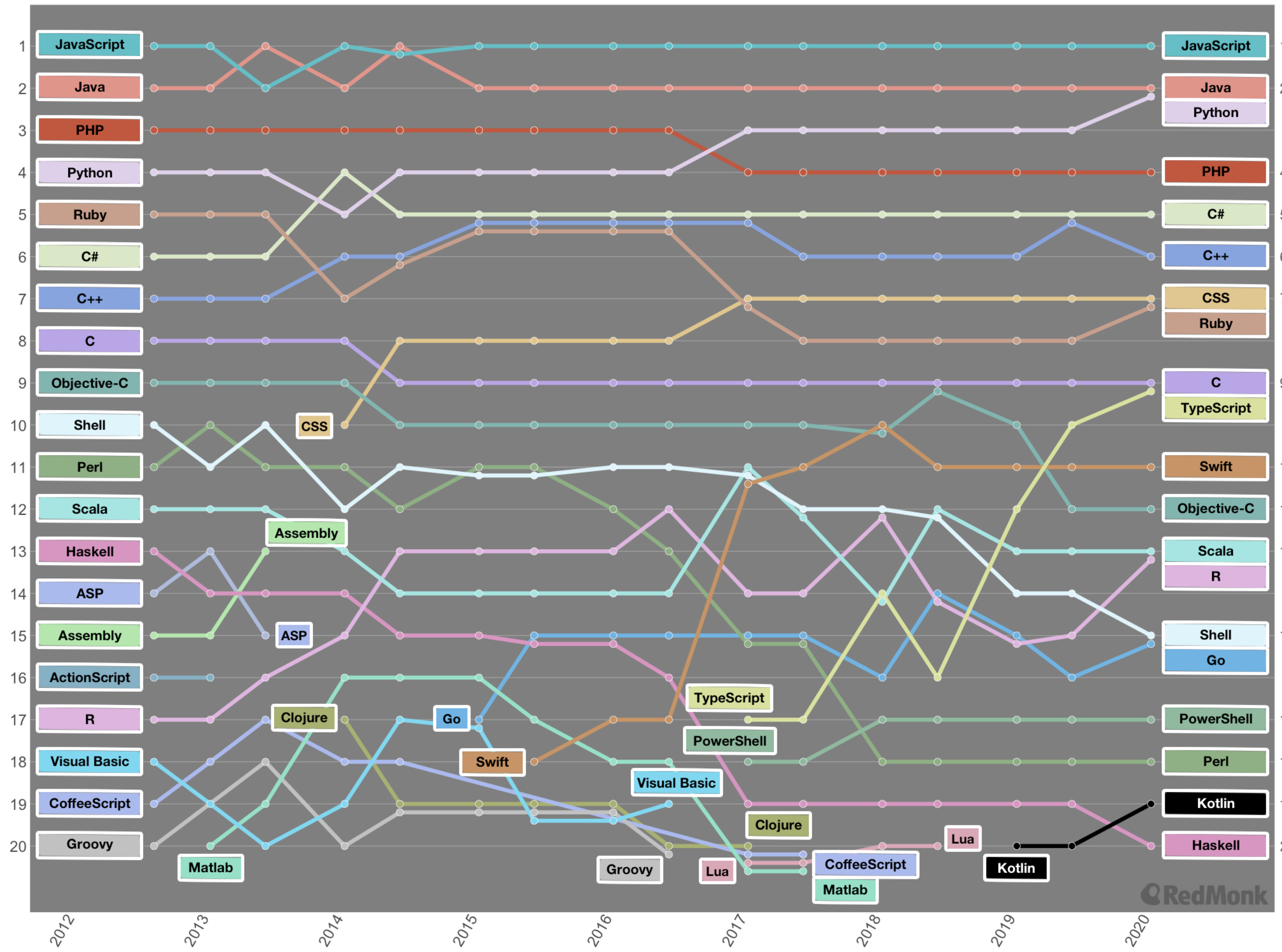
fengyang@nju.edu.cn

起源

- 1972年，由 Dennis Ritchie 和 Brian Kernighan 设计并实现
- 为了设计UNIX 操作系统而研发
- 迄今为止，仍然是世界上使用最为广泛的系统语言
- 目前已经成为一种流行的通用语言
- 1978年，由 Kernighan& Ritchie 正式公开，迅速改变了世界
- 1983年，由American National Standards Institute （ANSI）组织委员会进行了标准化制定，1988年标准版C 正式公布，即 ANSI C

RedMonk Language Rankings

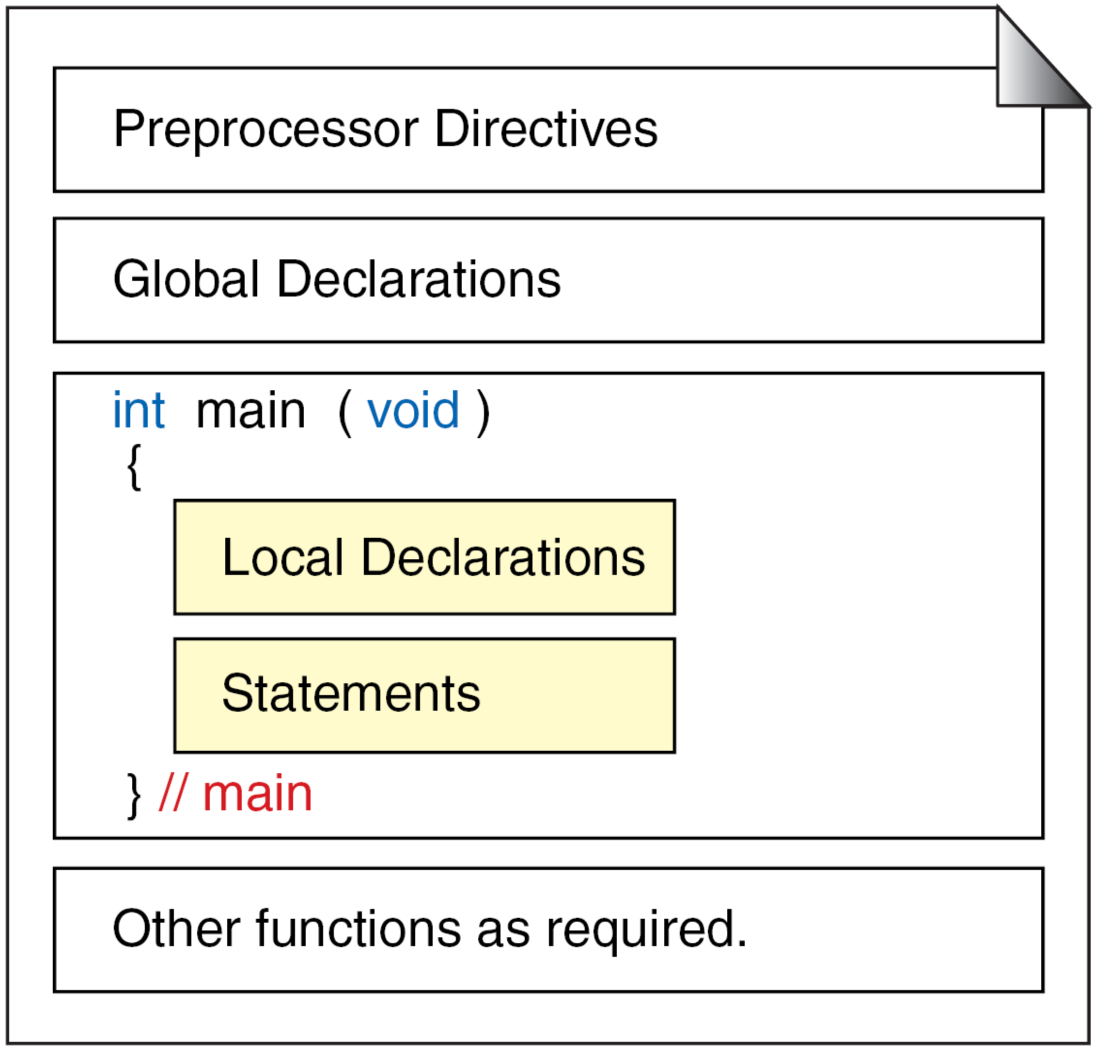
September 2012 - January 2020



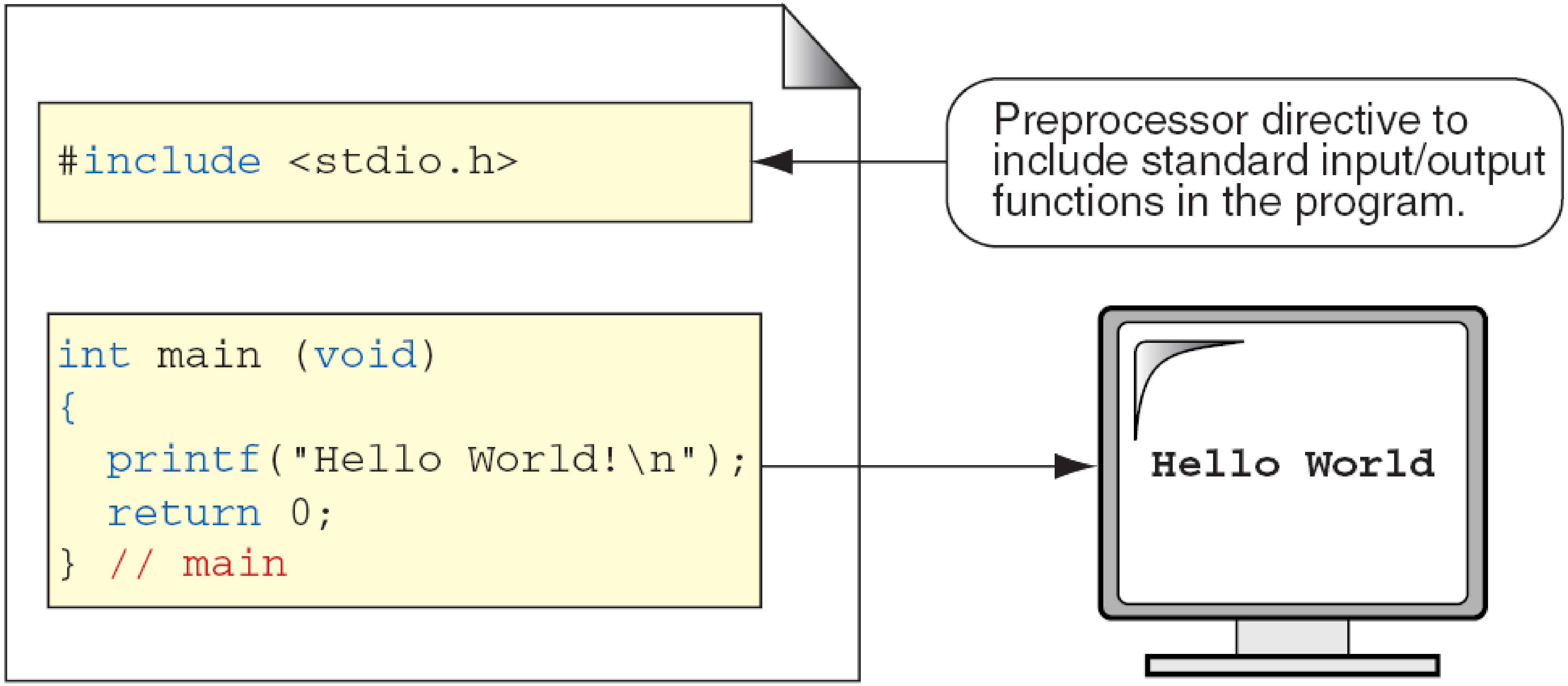
起源

- 为什么流行而且重要？ （好问题！ ）
  - 运行速度非常快，近似于直接用汇编语言编写的
  - 发展很早，具有非常多的遗留代码块
    - 操作系统
    - 编译器
    - 各类驱动
    - 数据库
  - 社区极为活跃且优秀

常见的 C 语言程序结构


Preprocessor Directives
Global Declarations
int main (void)
{
Local Declarations
Statements
} // main
Other functions as required.

C的第一个程序



```
#include <stdio.h>

int main (void)
{
   printf("Hello World!\n");
   return 0;
} // main
```

Preprocessor directive to include standard input/output functions in the program.
Hello World

常见的 C 语言结构

```c
/* The greeting program. This program demonstrates
   some of the components of a simple C program.
      Written by:  your name here
      Date:        date program written
*/
#include <stdio.h>

int main (void)
{
// Local Declarations

// Statements

   printf("Hello World!\n");

   return 0;
} // main
```

C语言注释示例

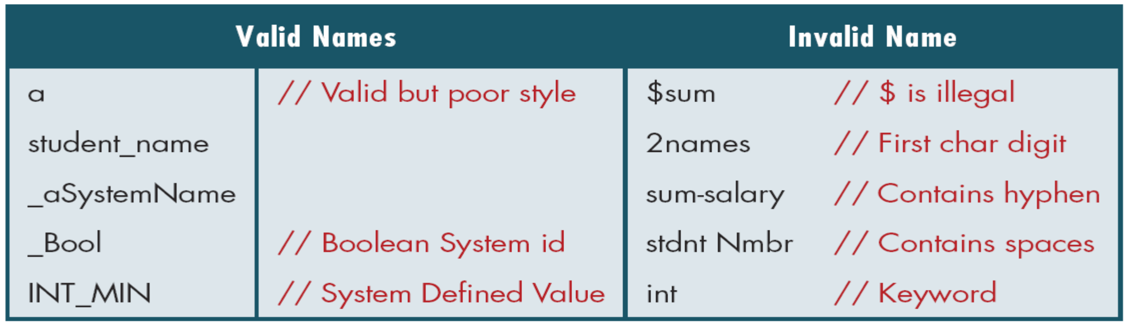
思考：
1. 块注释与行注释的用法？
2. 我们为什么要写注释？

```
/* This is a block comment that
   covers two lines.                */

/*
** It is a very common style to put the opening token
** on a line by itself, followed by the documentation
** and then the closing token on a separate line. Some
** programmers also like to put asterisks at the beginning
** of each line to clearly mark the comment.
*/


// This is a whole line comment


a = 5;            // This is a partial line comment
```

标识符（Identifier)

- 标识符用于命名程序中的数据与对象
- 每个标识符存储于计算机内存中的一个独有的位置
- C 语言中标识符的命名规则
  - 标识符必须以字母a~z、 A~Z或下划线开头 标识符区分大小写字母
  - 标识符的长度，c89规定31个字符以内，c99规定63个字符以内
  - C语言中的关键字，有特殊意义，不能作为标识符
  - 定义标识符最好取具有一定意义的字符串，便于记忆和理解

标识符（Identifier）



Valid Names		Invalid Name	
a	// Valid but poor style	$sum	// $ is illegal
student_name		2names	// First char digit
_aSystemName		sum-salary	// Contains hyphen
_Bool	// Boolean System id	stdnt Nmbr	// Contains spaces
INT_MIN	// System Defined Value	int	// Keyword

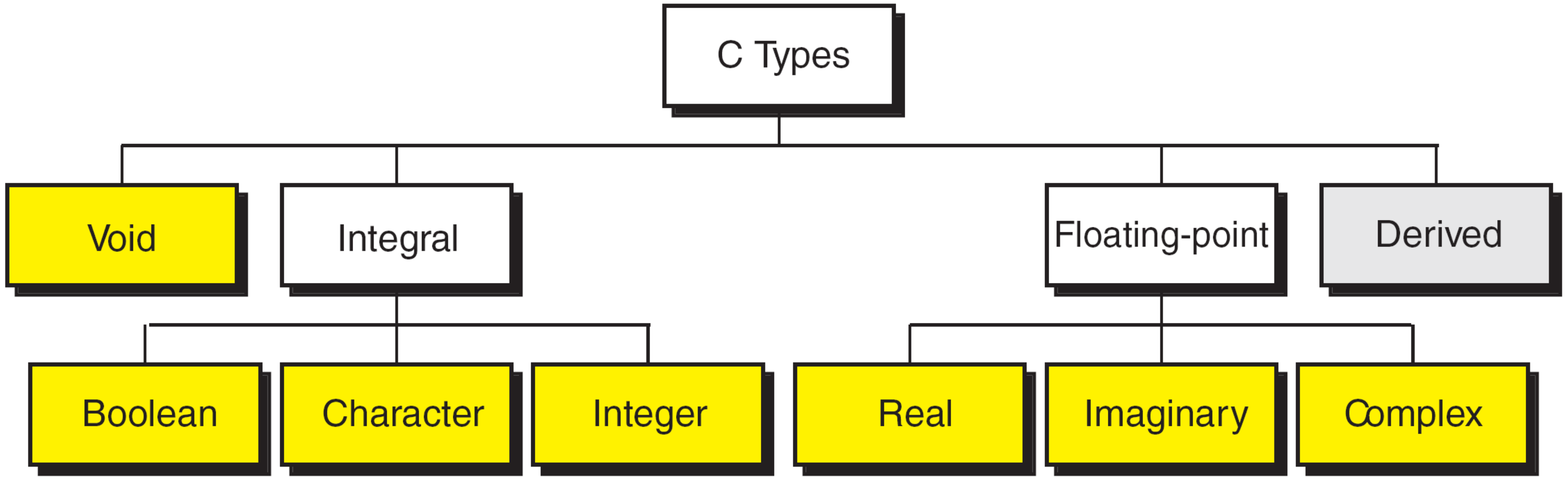
- 标识符用于命名程序中的数据与对象
- 每个标识符存储于计算机内存中的一个独有的位置
- C 语言中标识符的命名规则
  - 标识符必须以字母a~z、 A~Z或下划线开头 标识符区分大小写字母
  - 标识符的长度，c89规定31个字符以内，c99规定63个字符以内
  - C语言中的关键字，有特殊意义，不能作为标识符
  - 定义标识符最好取具有一定意义的字符串，便于记忆和理解

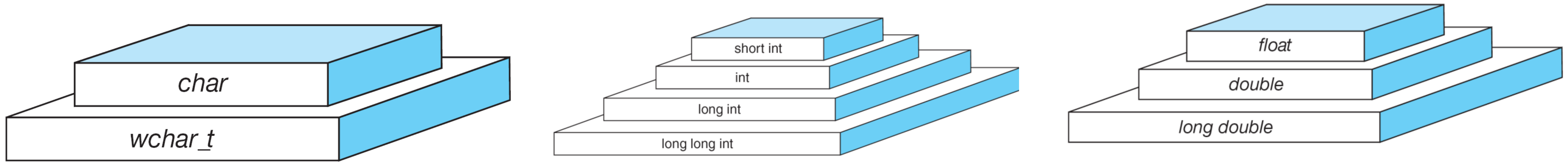
类型（Types）

- 类型的定义：A type defines a set of values and a set of operations that can be applied on those values.

类型（Types）

- 类型的定义：A type defines a set of values and a set of operations that can be applied on those values.

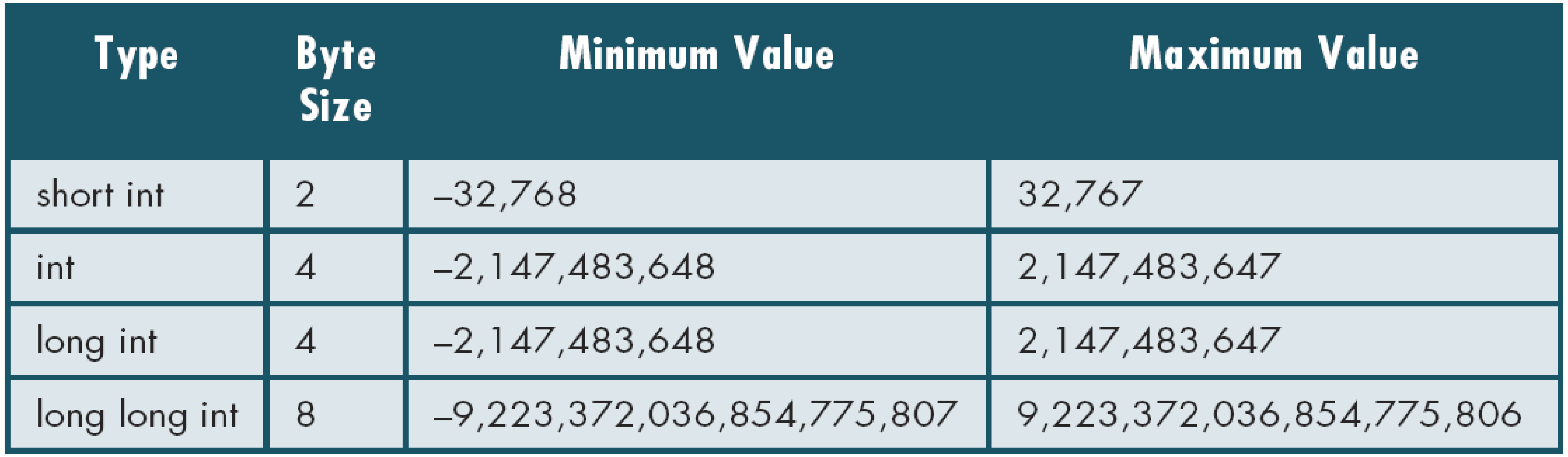




类型（Types）

- 类型的定义：A type defines a set of values and a set of operations that can be applied on those values.

sizeof (short) ≤ sizeof (int) ≤ sizeof (long) ≤ sizeof (long long)

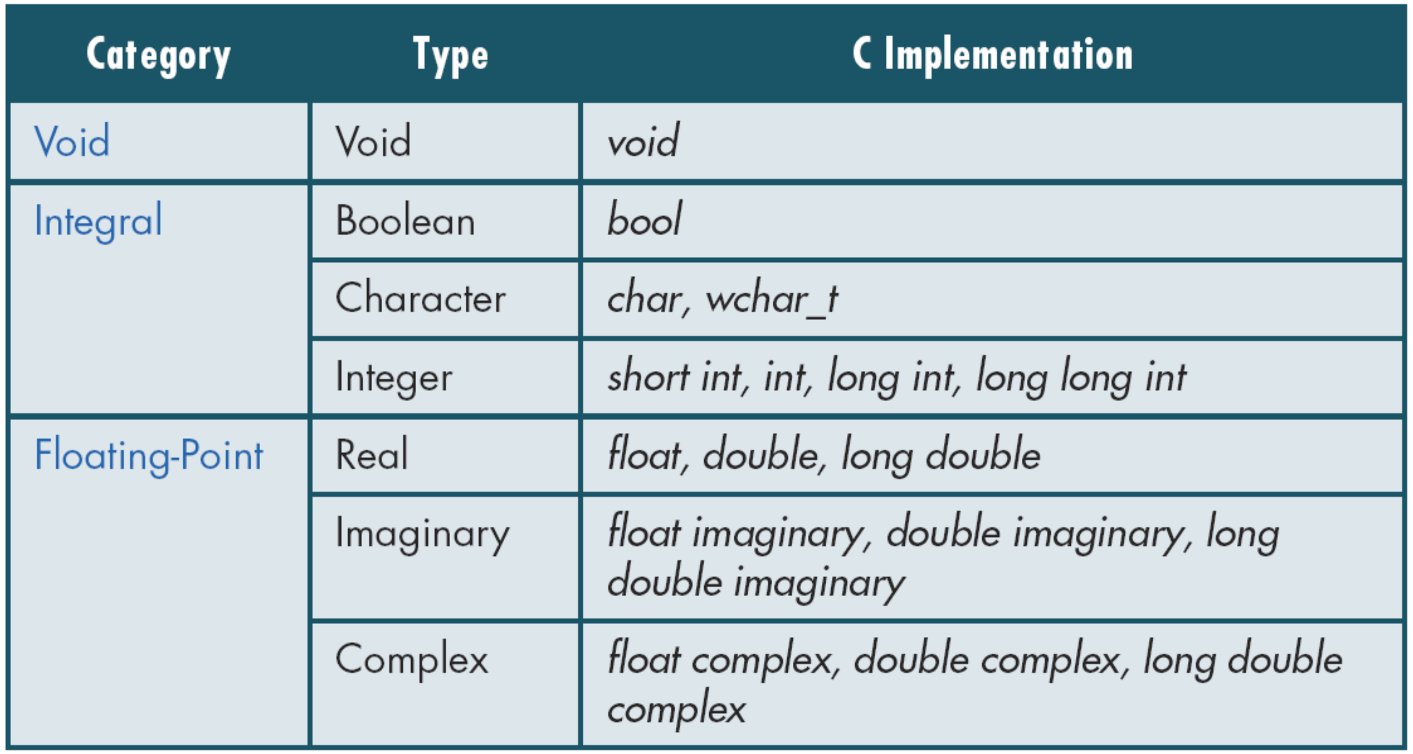


Type	Byte Size	Minimum Value	Maximum Value
short int	2	–32,768	32,767
int	4	–2,147,483,648	2,147,483,647
long int	4	–2,147,483,648	2,147,483,647
long long int	8	–9,223,372,036,854,775,807	9,223,372,036,854,775,806

类型（Types）

- 类型的定义：A type defines a set of values and a set of operations that can be applied on those values.

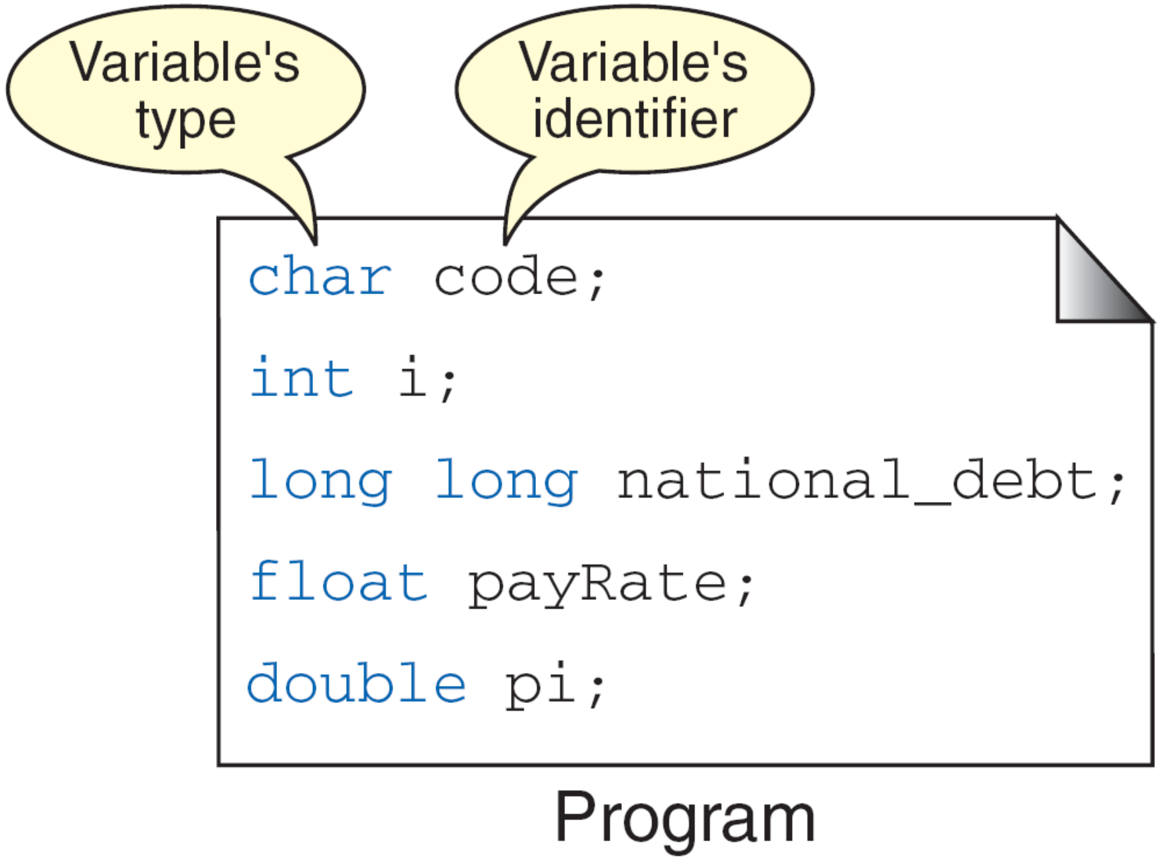
sizeof (float) ≤ sizeof (double) ≤ sizeof (long double)



Category	Type	C Implementation
Void	Void	*void*
Integral	Boolean	*bool*
	Character	*char, wchar_t*
	Integer	*short int, int, long int, long long int*
Floating-Point	Real	*float, double, long double*
	Imaginary	*float imaginary, double imaginary, long double imaginary*
	Complex	*float complex, double complex, long double complex*

变量（Variables）

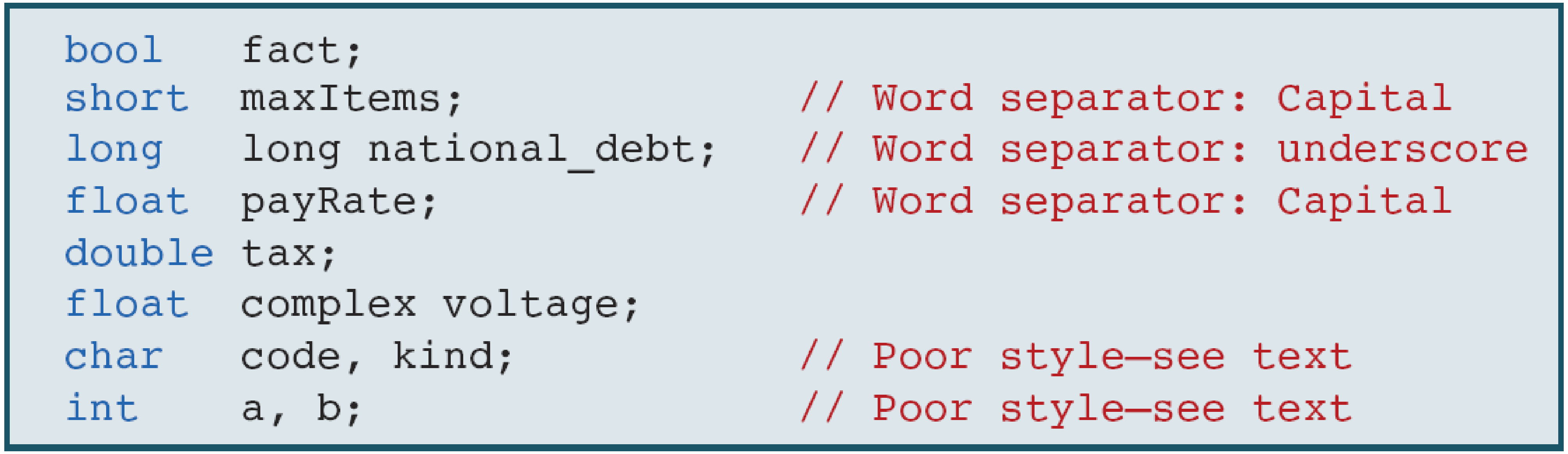
- 变量的定义：变量，即有名字的类型化内存地址。类型定义了变量支持的合法操作。



```
char code;
int i;
long long national_debt;
float payRate;
double pi;
```

变量（Variables）

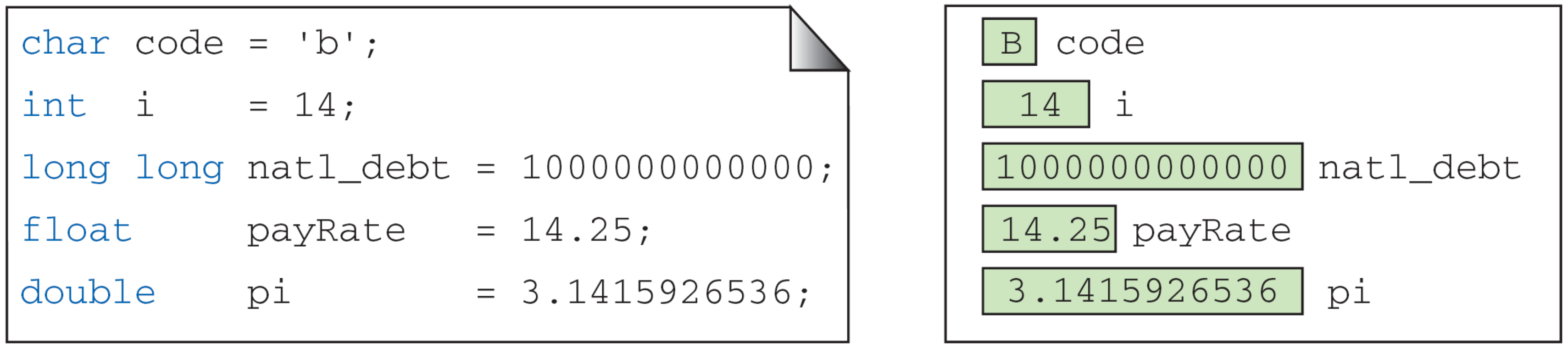
- 变量的定义：变量，即有名字的类型化内存地址。类型定义了变量支持的合法操作。



```
bool   fact;
short  maxItems;                // Word separator: Capital
long   long national_debt;     // Word separator: underscore
float  payRate;                 // Word separator: Capital
double tax;
float  complex voltage;
char   code, kind;              // Poor style—see text
int    a, b;                    // Poor style—see text
```

变量（Variables）

- 变量的定义：变量，即有名字的类型化内存地址。类型定义了变量支持的合法操作。



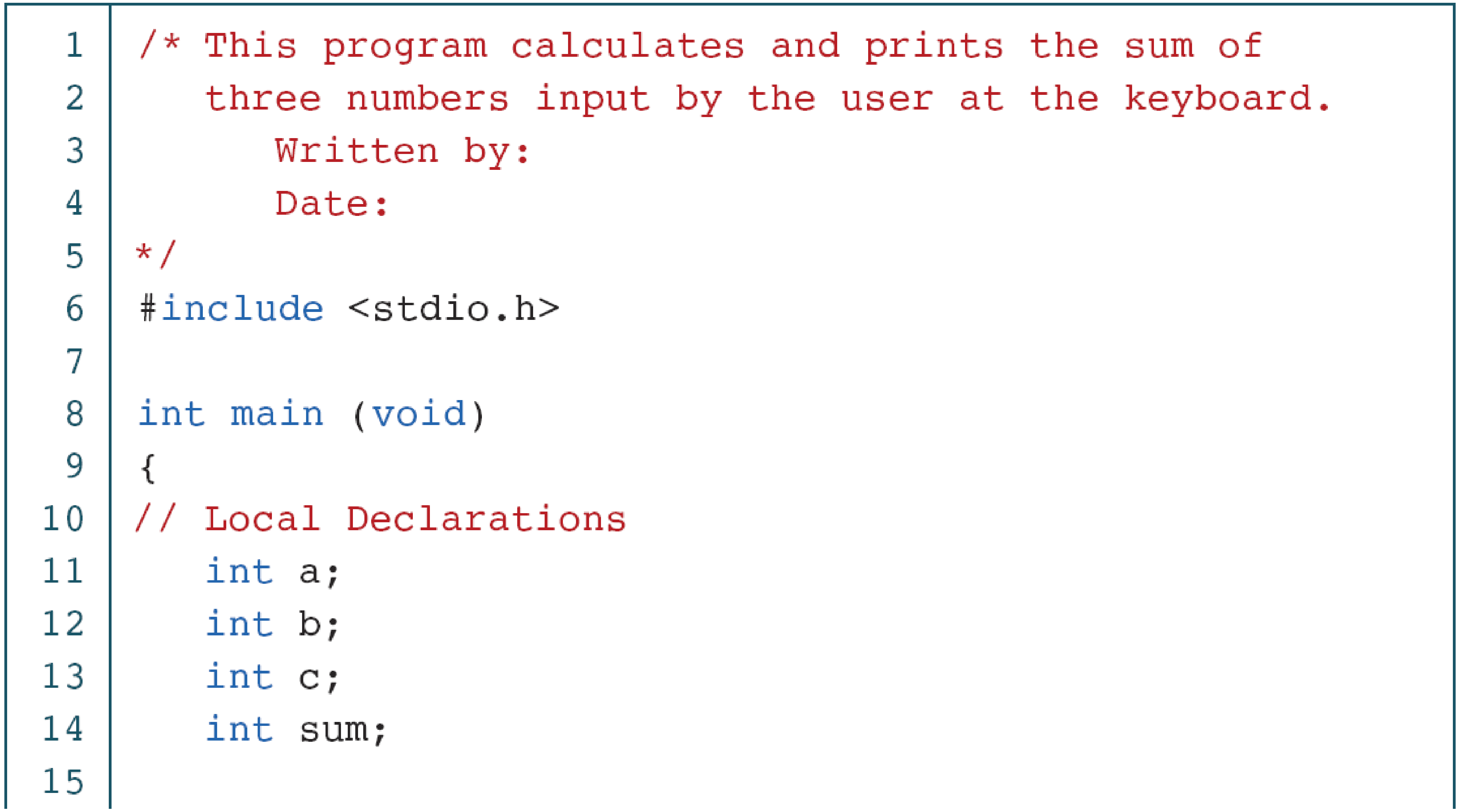
```
char code = 'b';
int  i    = 14;
long long natl_debt = 1000000000000;
float     payRate   = 14.25;
double    pi        = 3.1415926536;
```

Program　　　　　　　　Memory

变量（Variables）

- 区分变量的声明与初始化
  - 变量在声明或定义的时候并未初始化
  - 初始化明确了变量指向的内存地址
  - 应当在变量定义时刻即初始化变量

变量（Variables）



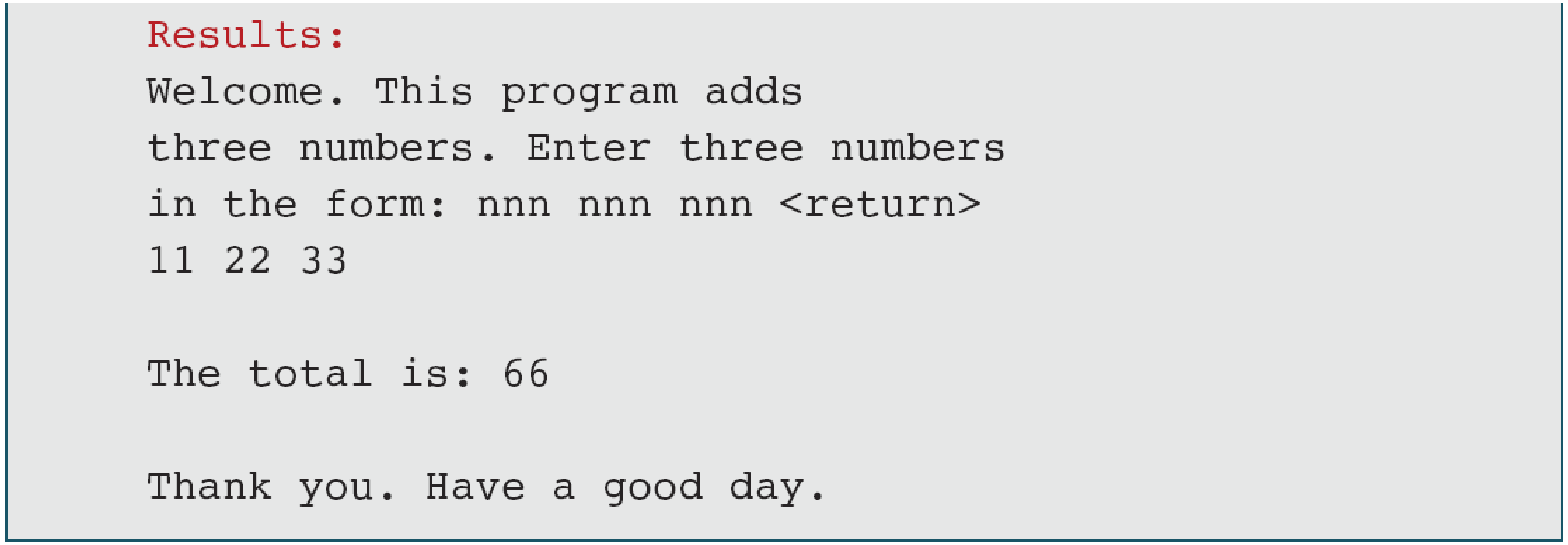
```c
/* This program calculates and prints the sum of
   three numbers input by the user at the keyboard.
      Written by:
      Date:
*/
#include <stdio.h>

int main (void)
{
// Local Declarations
   int a;
   int b;
   int c;
   int sum;

```

变量（Variables）

```c
// Statements
   printf("\nWelcome. This program adds\n");
   printf("three numbers. Enter three numbers\n");
   printf("in the form: nnn nnn nnn <return>\n");
   scanf("%d %d %d", &a, &b, &c);

   // Numbers are now in a, b, and c. Add them.
   sum = a + b + c;

   printf("The total is: %d\n\n", sum);

   printf("Thank you. Have a good day.\n");
   return 0;
}  //  main
```

思考：这个程序写得好吗？

变量（Variables）



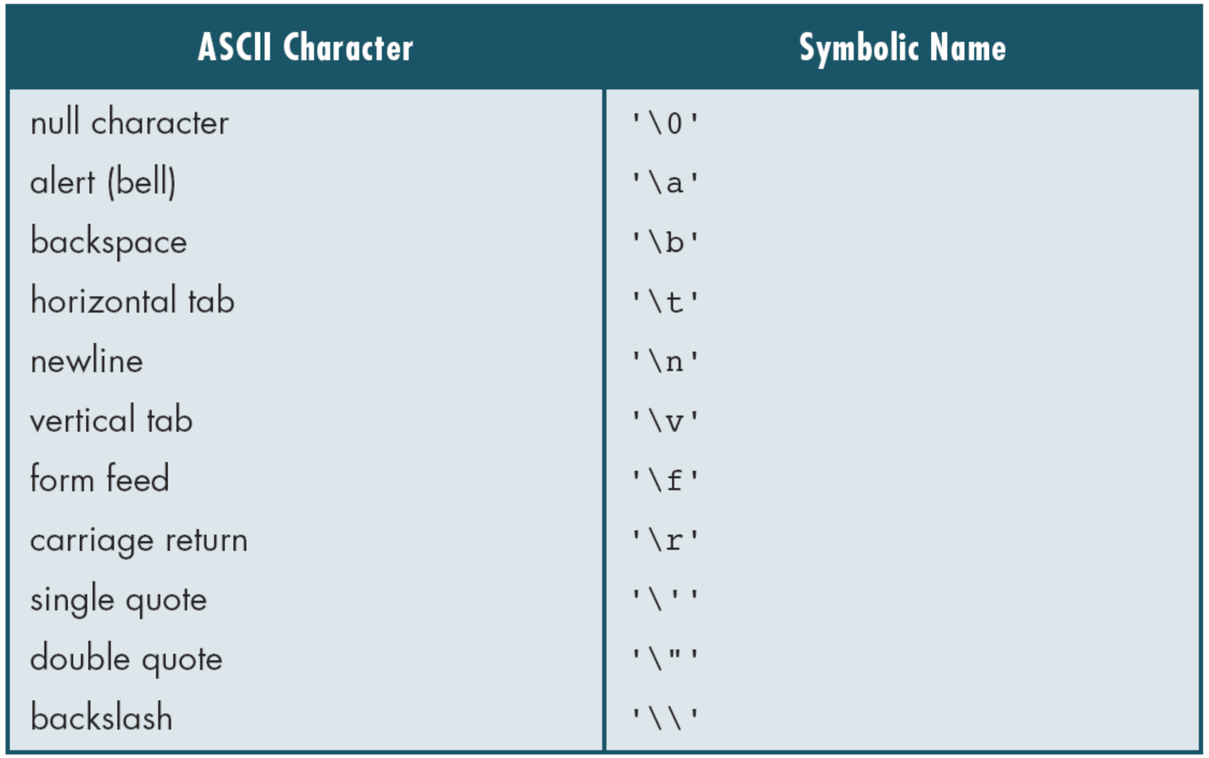
```
Results:
Welcome. This program adds
three numbers. Enter three numbers
in the form: nnn nnn nnn <return>
11 22 33

The total is: 66

Thank you. Have a good day.
```

常量（Constant）

- 定义：
  - 不能在运行时改变值的数据
  - 同样具有类型

常量（Constant）

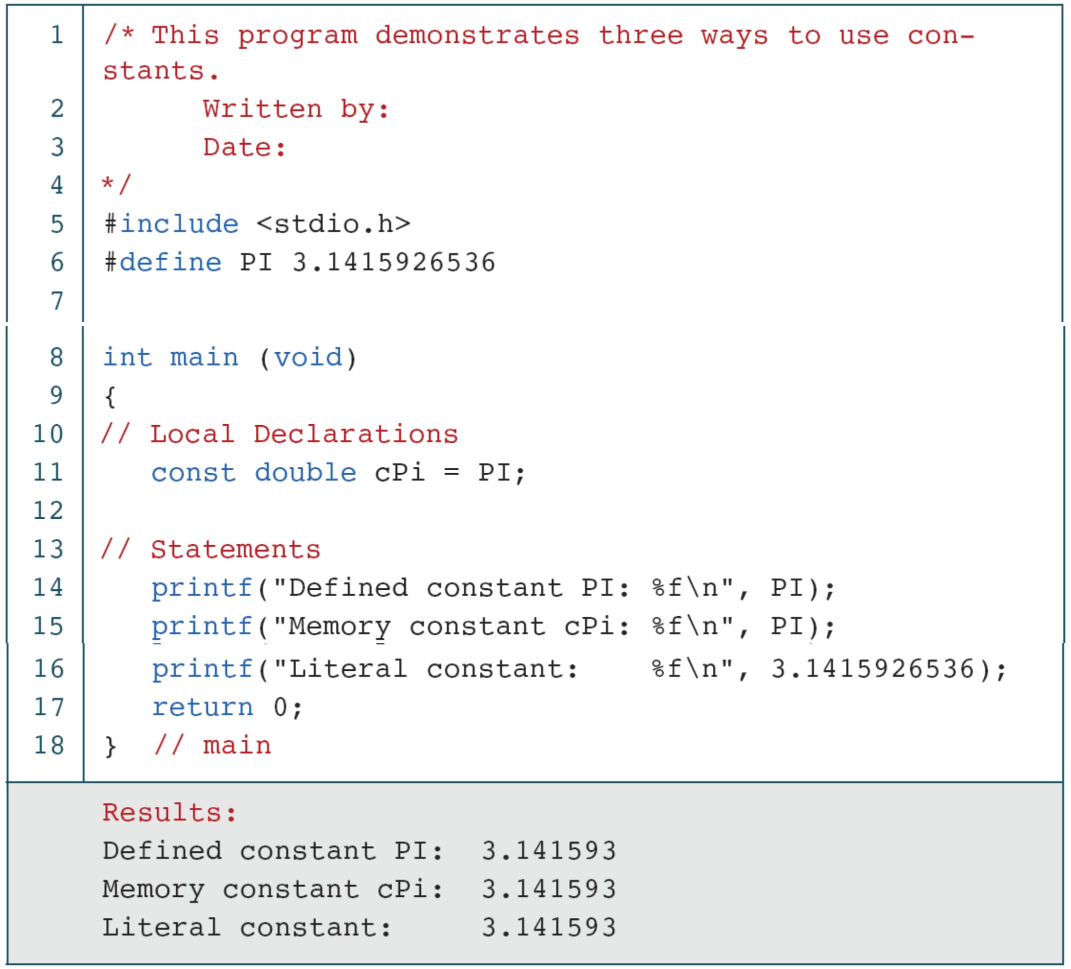
- C语言中，字符型常量通过单引号标示，而字符串则使用双引号标示



ASCII Character	Symbolic Name
null character	`'\0'`
alert (bell)	`'\a'`
backspace	`'\b'`
horizontal tab	`'\t'`
newline	`'\n'`
vertical tab	`'\v'`
form feed	`'\f'`
carriage return	`'\r'`
single quote	`'\''`
double quote	`'\"'`
backslash	`'\\'`

常量（Constant）

- C语言中，字符型常量通过单引号标示，而字符串则使用双引号标示

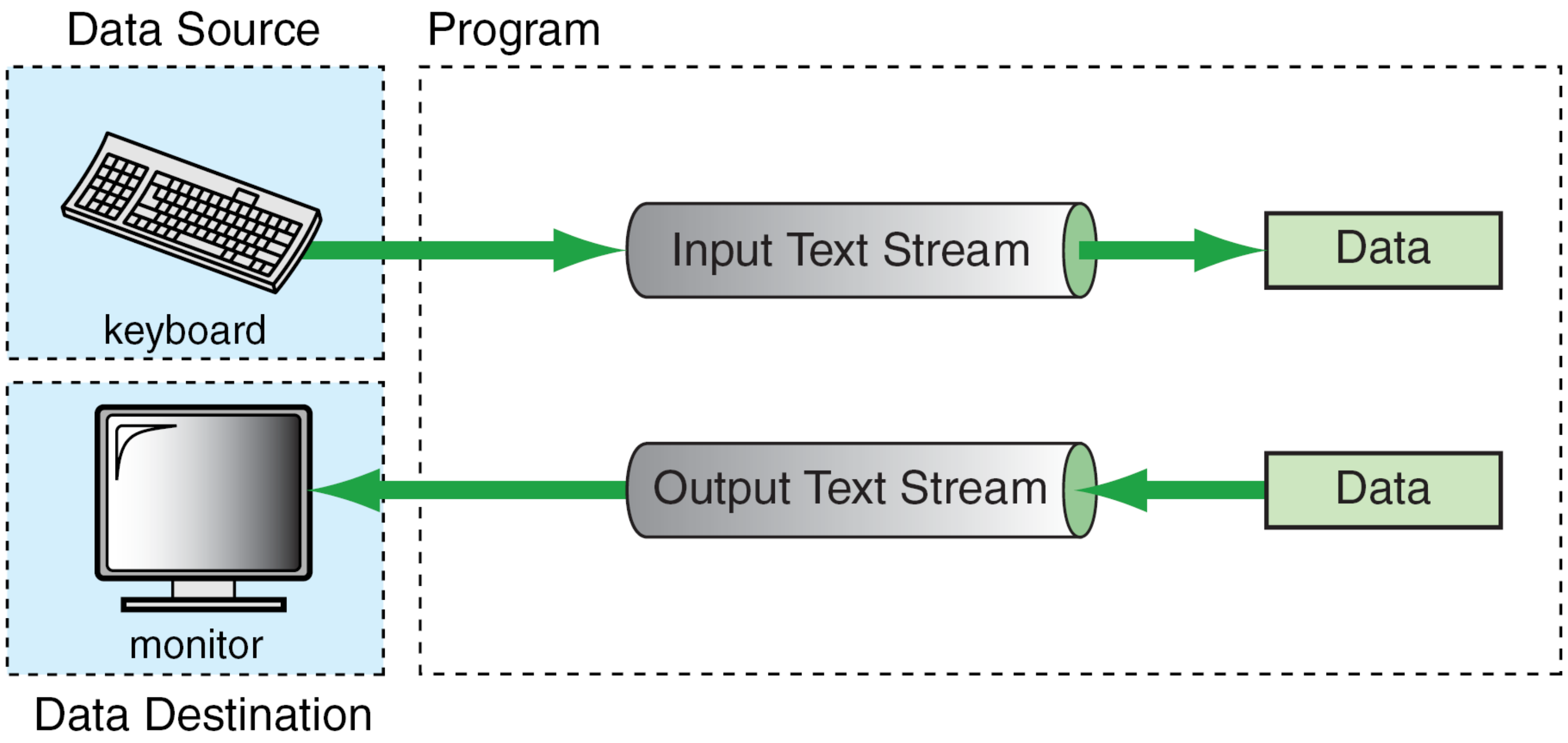


```c
/* This program demonstrates three ways to use con-
stants.
      Written by:
      Date:
*/
#include <stdio.h>
#define PI 3.1415926536

int main (void)
{
// Local Declarations
   const double cPi = PI;

// Statements
   printf("Defined constant PI: %f\n", PI);
   printf("Memory constant cPi: %f\n", PI);
   printf("Literal constant:    %f\n", 3.1415926536);
   return 0;
}  // main
```

```
Results:
Defined constant PI:   3.141593
Memory constant cPi:   3.141593
Literal constant:      3.141593
```

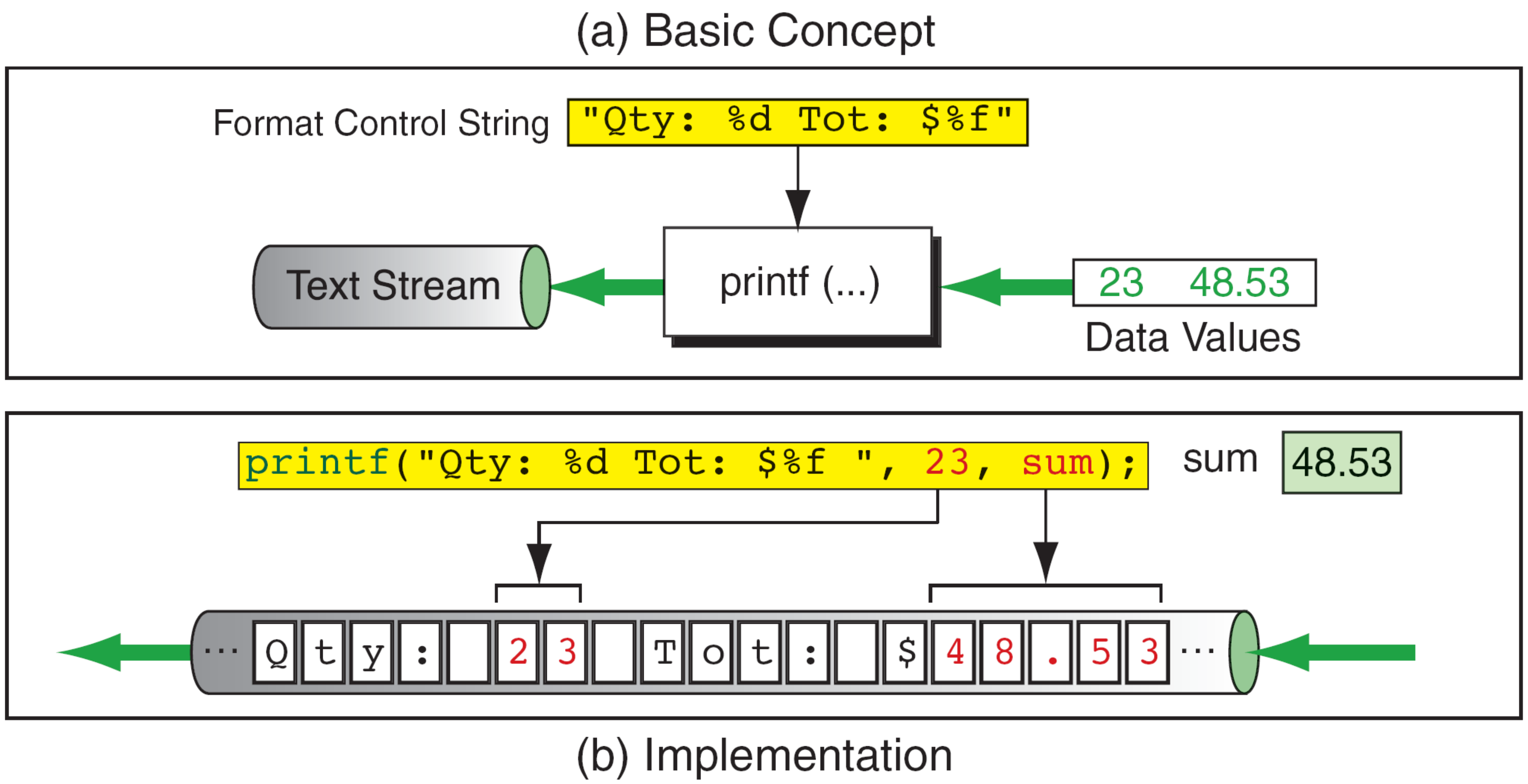
输入输出（Input/Output）

- 回顾我们前面的介绍“C语言是目前最为流行的系统编程语言”



格式化输出

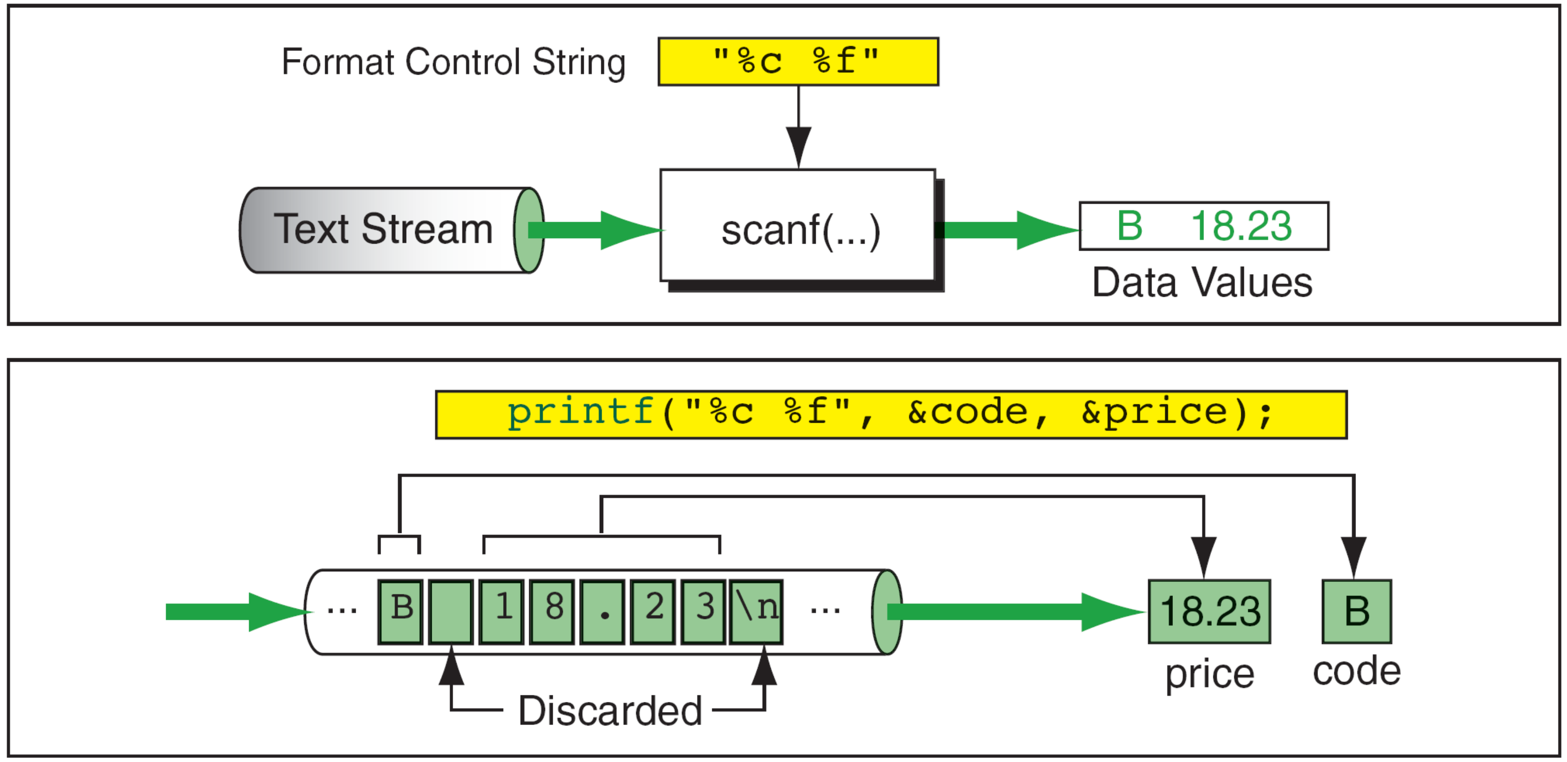
- C语言支持通过 printf() 函数格式化输出



格式化输入

- C语言支持通过 scanf() 函数格式化输出

(a) Basic Concept



(b) Implementation