C语言程序设计基础 01

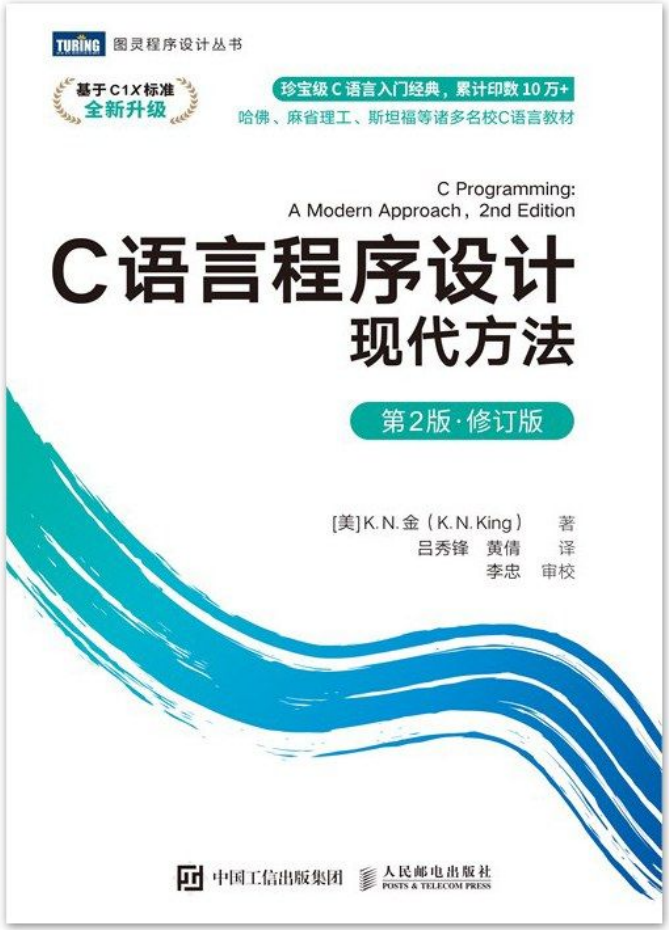
课程信息

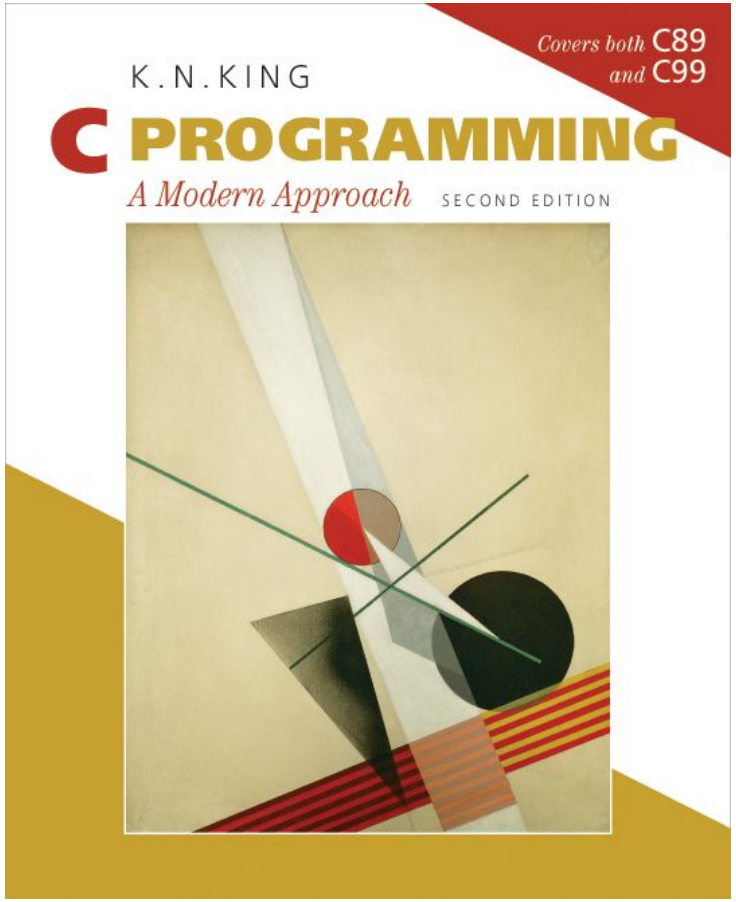
课程信息

1. 介绍计算机科学领域，将编程工具和技术（使用 C 编程语言）与严格的数学分析和推理相结合。
2. 主题包括：
   a. 数据表示；
   b. 程序控制流（条件、循环、函数）；
   c. 数理逻辑和形式化证明；
   d. 浮点数的表示和数值计算；
   e. 算法和运行时间分析；
   f. 软件工程原理（形式化规范和设计、测试和验证）。

3. 本课程不需要预先具备编程经验。

课程信息:教材





April 2008

课程信息

- 课程网站:https://zhangl.github.io/cpl.htm
- QQ群:153540807
- 问讲师:
  - zhangl@nju.edu.cn
- 问助教:
  - 李双杰: 2292847133@qq.com
  - 宋鉴清: 1149602149@qq.com
  - 周昊棣: 490902429@qq.com
- 官方网站
  - docs.cpl.icu
  - oj.cpl.icu

课程信息:成绩构成

分值	项目
10%:	平时练习, 每周一次(https://oj.cpl.icu/)
15%	平时机试
20%	平时机试
25%	期末项目(期中项目？tbd)
30%:	期末机试

课程信息:学术事务行为准则

大学与其他研究中心的区 别在于教学与学习之间的关系所占据的核心位置。正是通 过这一关系，大学才能 实现其传统使命的一个重要部分，这个使命既来自社会，也来自 历史:传承并鼓励一种具有 鉴别力的思维习惯，同时保持好奇心，一种既公平又大胆的思维习惯，重视开放、诚实和礼貌，而不是任何私人利益。

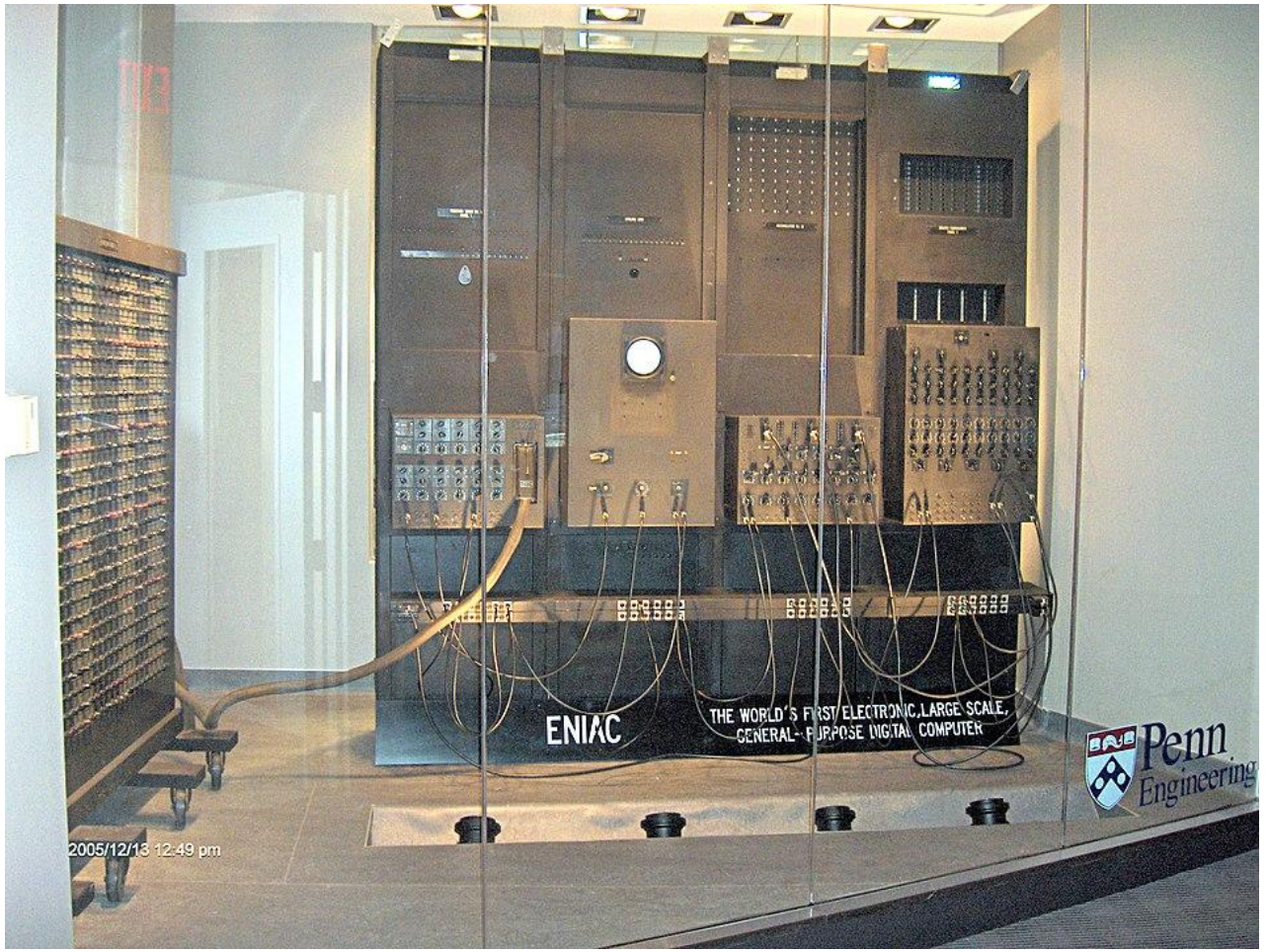
这一使命不仅仅是一种虔诚的希望。它代表了自由探究所必需的条件，这是大学的生命之源。它的 实现取决于这一关系的福祉。这种关系的双方根据 专业、知识和经验的差异，确定彼此作 为教师和学生的角色，通 过尊重、对真理的共同 热情以及对继续传承大学的原 则和理念的共同 责任而相互联系。

因此，这一准则关注的是教职员工和学生的 责任，不是他 们作为行政、专业或社会团体的成员的责任，而是他 们在教学和学习关系的所有合作阶段中的责任。

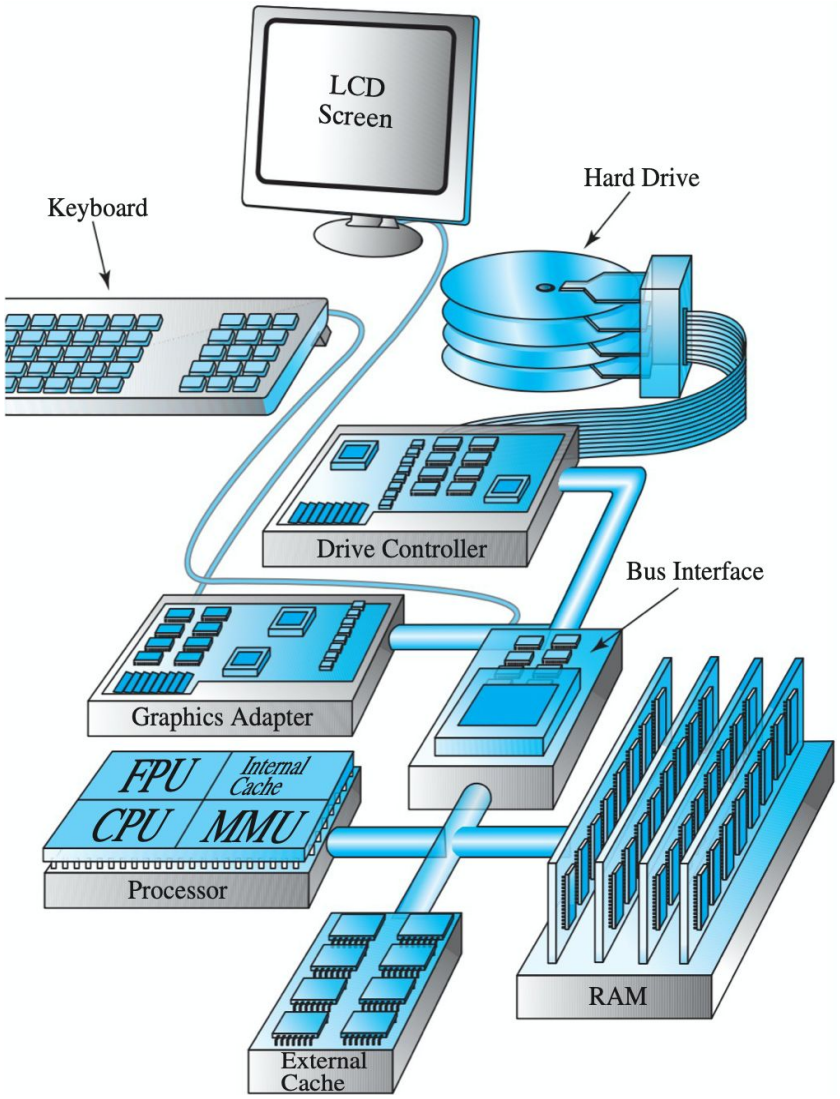
当教师或学生抛弃 对对方以及参与学习的其他人的尊重，支持自身利益，当真理被 权宜之计绑架时，这种合作受到威胁。大学有责任代表教师和学生，并根据自身原 则和理念履行职责，确保学术成就不会因作弊或虚假 陈述而被掩盖或破坏，确保 评估过程符合最高的公平和 诚实标准，确保恶意甚至恶作剧性的干扰不会威胁教育进程。

在这些领域，教师和学生必然具有共同的 兴趣和共同的 责任。

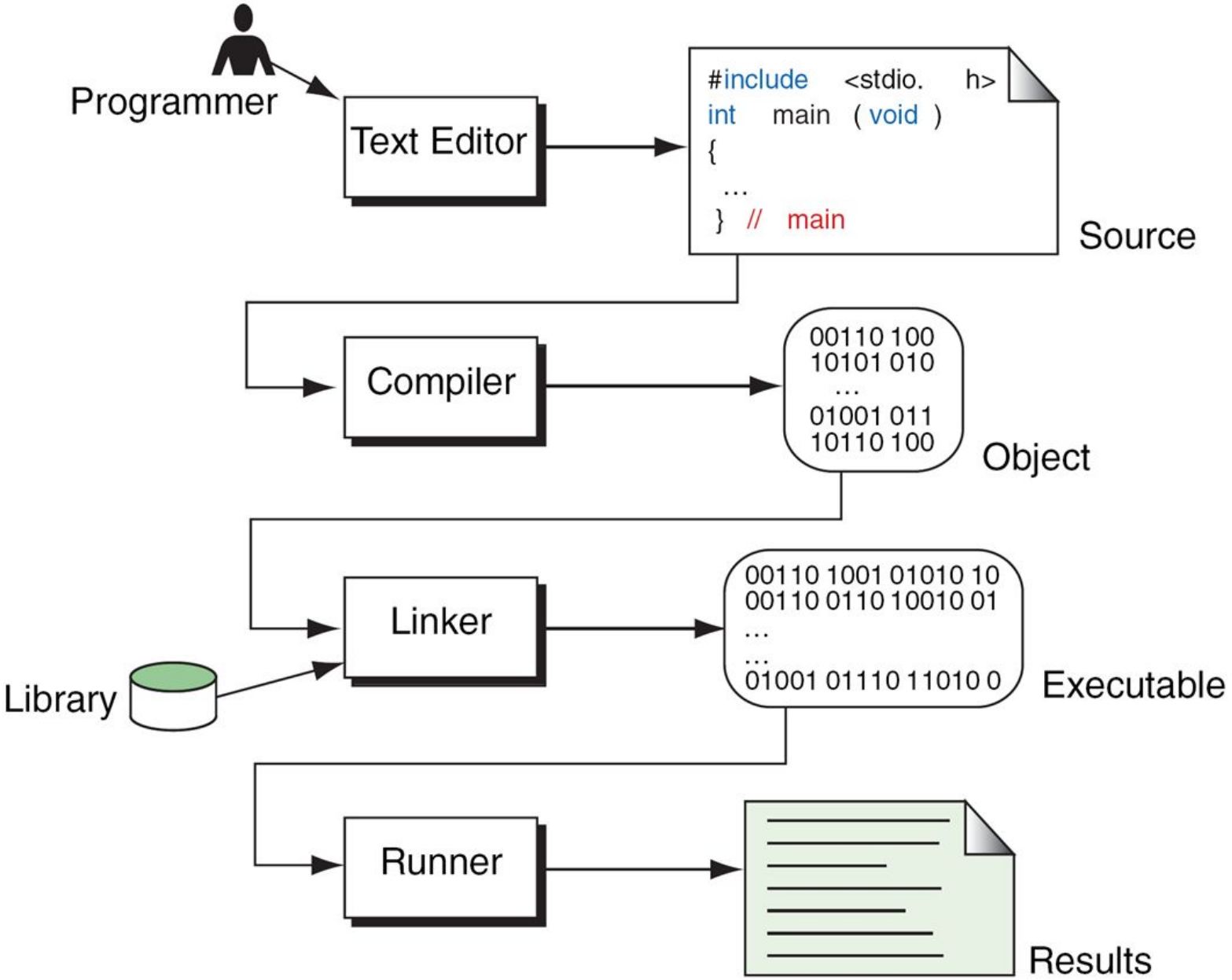
第一台计算机ENIAC



现代计算机结构


LCD
Screen
Keyboard
Hard Drive
Drive Controller
Bus Interface
Graphics Adapter
FPU
Internal
Cache
CPU
MMU
Processor
RAM
External
Cache

编程语言

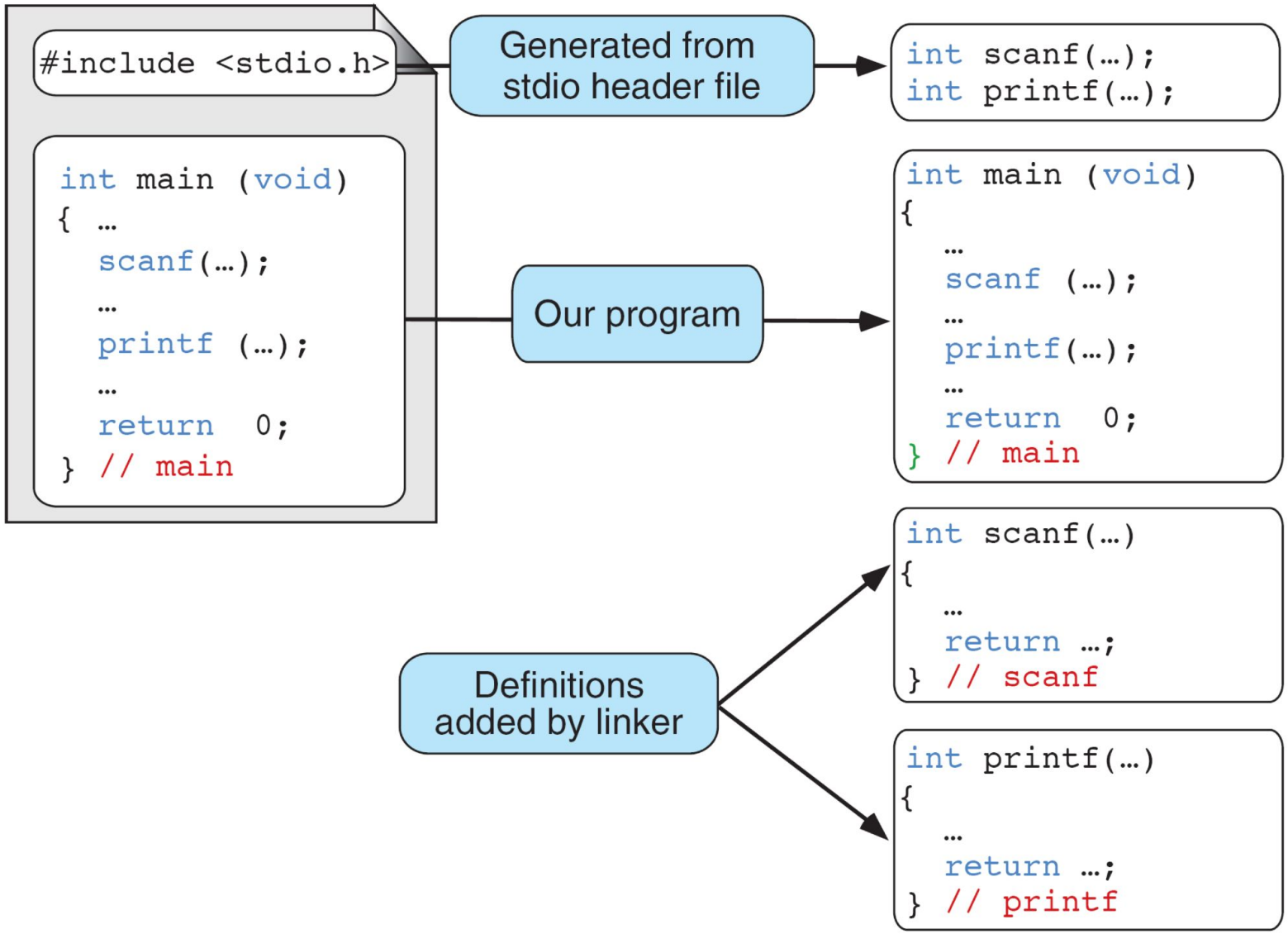

Programmer
Text Editor
#include <stdio. h>
int main (void)
{
…
} // main
Source
Compiler
00110 100
10101 010
…
01001 011
10110 100
Object
Linker
Library
00110 1001 01010 10
00110 0110 10010 01
…
…
01001 01110 11010 0
Executable
Runner
Results

hello world

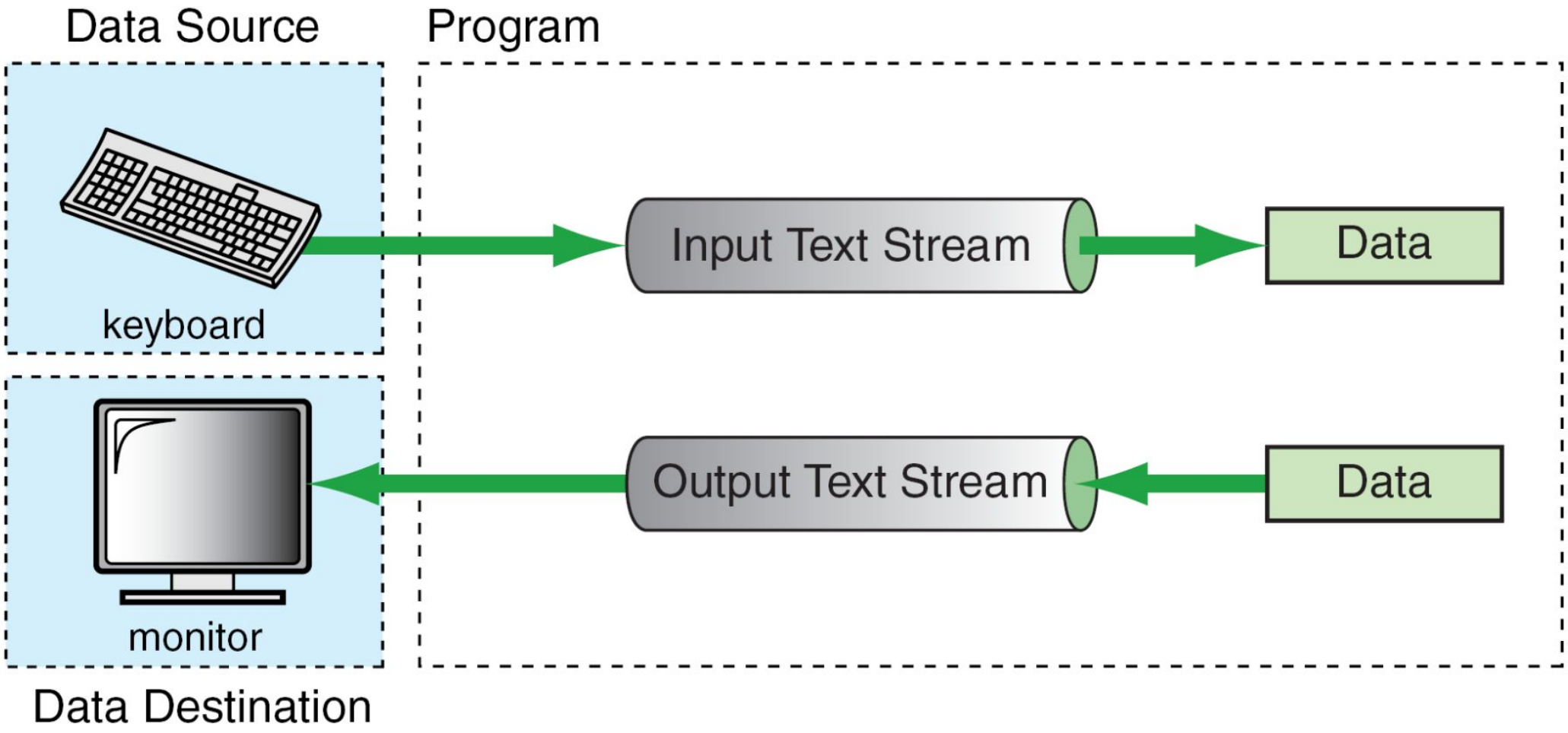
```c
/* pun.c (Chapter 2, page 10) */

#include <stdio.h>

int main(void)
{
  printf("To C, or not to C: that is the question.\n");
  return 0;
}
```

每个符号什么含义:看看去掉之后会发生什么

计算机程序


#include <stdio.h>
int main (void)
{ …
scanf(…);
…
printf (…);
…
return 0;
} // main
Generated from stdio header file
int scanf(…);
int printf(…);
Our program
int main (void)
{
…
scanf (…);
…
printf(…);
…
return 0;
} // main
Definitions added by linker
int scanf(…)
{
…
return …;
} // scanf
int printf(…)
{
…
return …;
} // printf

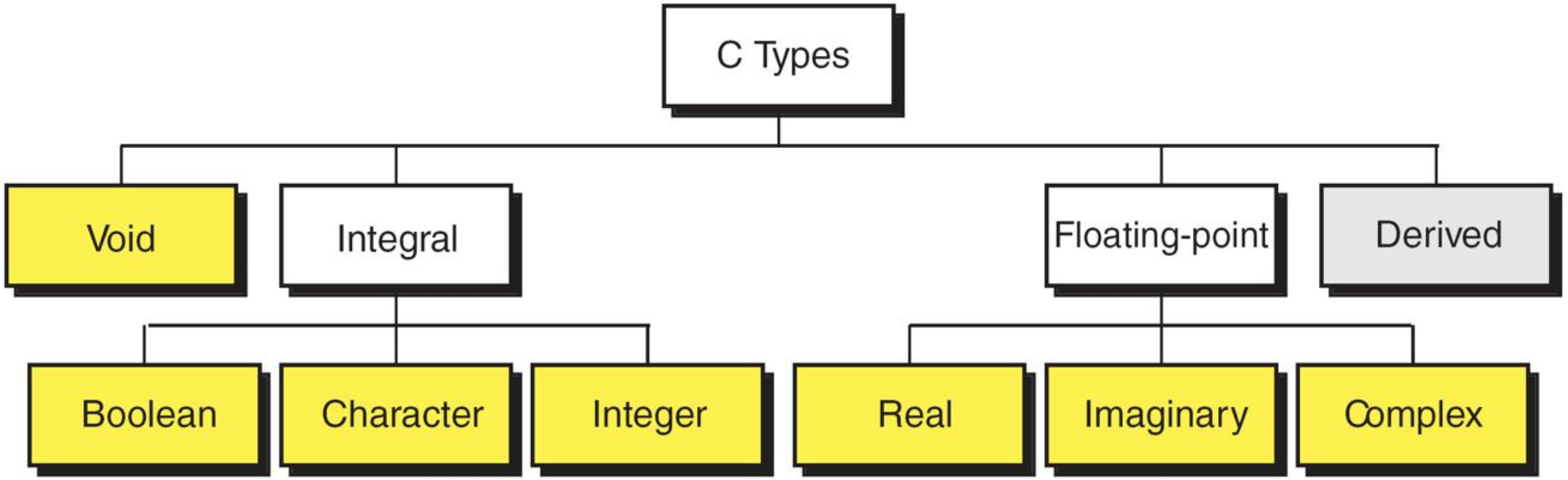
计算机程序


Data Source
Program
keyboard
Input Text Stream
Data
monitor
Output Text Stream
Data
Data Destination

数据类型

- 数据的类型独立于编程语言存在
  - 数字的历史比编程语言更悠久
  - 这些独立于编程语言的类型通常被称为抽象数据类型(abstract data types)
- 数据类型的定义
- 数值数据(numerical data)
  - 整数, 有理数, 实数
- 符号数据(symbolic data)
  - 布尔数据
  - 字符数据
  - 文本数据(字符串)

C数据类型


C Types
Void
Integral
Floating-point
Derived
Boolean
Character
Integer
Real
Imaginary
Complex

ASCII

发音(ass-key)

ASCII Table

Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char
0	0	0		32	20	40	[space]	64	40	100	@	96	60	140	`
1	1	1		33	21	41	!	65	41	101	A	97	61	141	a
2	2	2		34	22	42	"	66	42	102	B	98	62	142	b
3	3	3		35	23	43	#	67	43	103	C	99	63	143	c
4	4	4		36	24	44	$	68	44	104	D	100	64	144	d
5	5	5		37	25	45	%	69	45	105	E	101	65	145	e
6	6	6		38	26	46	&	70	46	106	F	102	66	146	f
7	7	7		39	27	47	'	71	47	107	G	103	67	147	g
8	8	10		40	28	50	(	72	48	110	H	104	68	150	h
9	9	11		41	29	51	)	73	49	111	I	105	69	151	i
10	A	12		42	2A	52	*	74	4A	112	J	106	6A	152	j
11	B	13		43	2B	53	+	75	4B	113	K	107	6B	153	k
12	C	14		44	2C	54	,	76	4C	114	L	108	6C	154	l
13	D	15		45	2D	55	-	77	4D	115	M	109	6D	155	m
14	E	16		46	2E	56	.	78	4E	116	N	110	6E	156	n
15	F	17		47	2F	57	/	79	4F	117	O	111	6F	157	o
16	10	20		48	30	60	0	80	50	120	P	112	70	160	p
17	11	21		49	31	61	1	81	51	121	Q	113	71	161	q
18	12	22		50	32	62	2	82	52	122	R	114	72	162	r
19	13	23		51	33	63	3	83	53	123	S	115	73	163	s
20	14	24		52	34	64	4	84	54	124	T	116	74	164	t
21	15	25		53	35	65	5	85	55	125	U	117	75	165	u
22	16	26		54	36	66	6	86	56	126	V	118	76	166	v
23	17	27		55	37	67	7	87	57	127	W	119	77	167	w
24	18	30		56	38	70	8	88	58	130	X	120	78	170	x
25	19	31		57	39	71	9	89	59	131	Y	121	79	171	y
26	1A	32		58	3A	72	:	90	5A	132	Z	122	7A	172	z
27	1B	33		59	3B	73	;	91	5B	133	[	123	7B	173	{
28	1C	34		60	3C	74	<	92	5C	134	\	124	7C	174	|
29	1D	35		61	3D	75	=	93	5D	135	]	125	7D	175	}
30	1E	36		62	3E	76	>	94	5E	136	^	126	7E	176	~
31	1F	37		63	3F	77	?	95	5F	137	_	127	7F	177	

转义序列(escape sequence)

由两个单引号(')括起来的一个特殊字符序列, 其中的字符序列以\开头, 后面是一个特殊字符或ASCII码

特殊转义序列

'\n' (换行符), '\a' (响铃符), '\t' (制表符), '\b' (退格符),

'\\' (反斜杠本身), '\'' (单引号本身), '\"' (双引号本身), '\%' (百分号本身) , ...

ASCII转义序列

八进制:'\ddd', 如:'\101'(表示'A')

十六进制:'\xdd'(或'\Xdd'), 如:'\x47'(表示'G')

空字符 Null

空字符（Null character）又称结束符，缩写NUL，是一个数值为0的控制字符。在C语言及其派生语言非常重要，在C语言中空字符是字符串的结束码。因此字符串的长度可以为任意自然数，但需多增加一个字符的空间存储空字符。

转义符\0表示空字符。\0不是一个单独的转义序列，而是一个以八进制表示常量，而常量的数值为0，\0后面不能接0至7的数字，不然会视为是一个八进制的数字。

在变量中存储数据

随着我们想要执行的计算变得越来越复杂，仅依赖数字和运算符非常麻烦。

我们可以编写非常复杂的嵌套表达式，但这会使我们的代码非常难以理解。

我们可以通过将这个问题分解为中间步骤来改进我们的代码。

我们将**中间步骤的数据值储存到变量**中，在后续的计算中就可以引用这些值。

声明变量：`int high;`

赋值语句：`high = 12345;`

在变量中存储数据

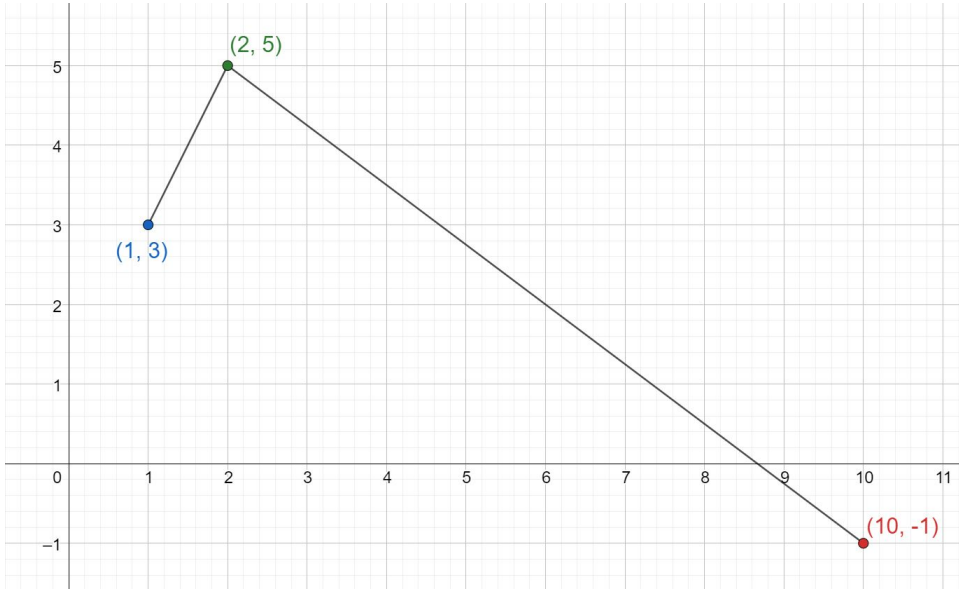
计算右图三点之间路径的距离

代码1:

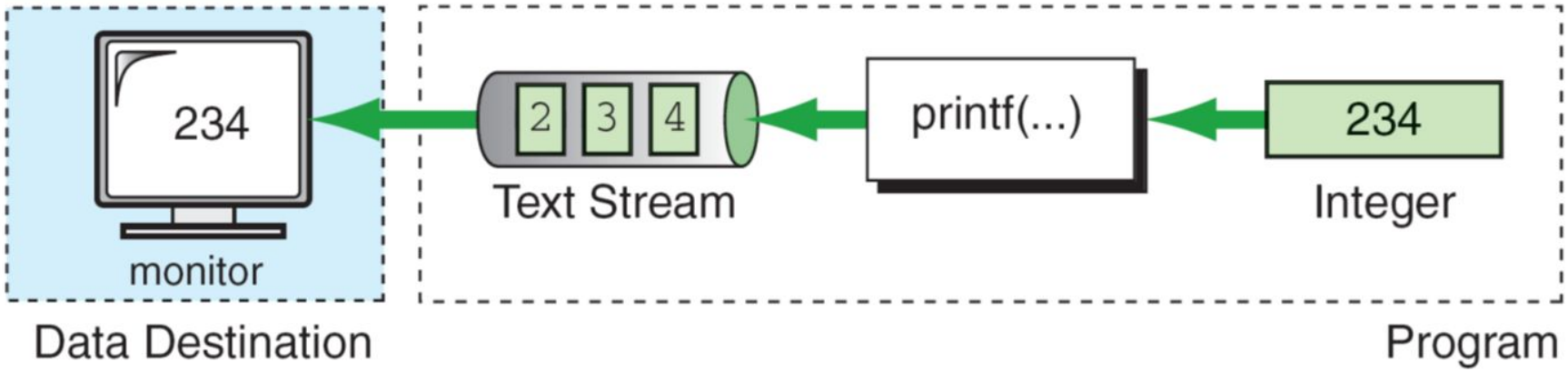
```
printf("%f\n", sqrt((2 - 1) * (2 - 1) + (5 - 3) * (5 - 3)) + sqrt((10 - 2)*(10 - 2) + ((-1) - 5) * ((-1) - 5)));
```

代码2:

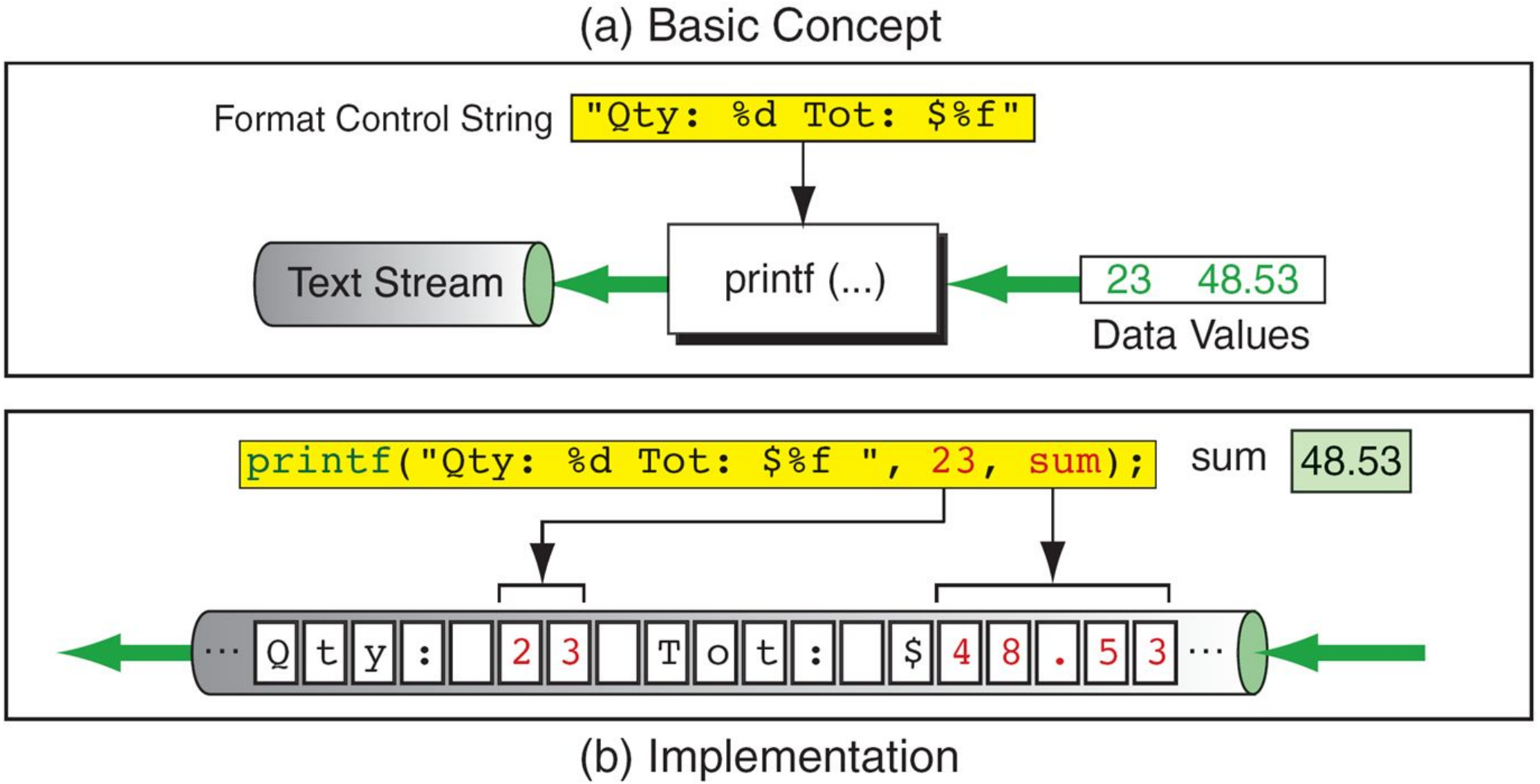
```
double distance1 = sqrt((2 - 1) * (2 - 1) + (5 - 3) * (5 - 3));
double distance2 = sqrt((10 - 2)*(10 - 2) + ((-1) - 5) * ((-1) - 5) ) ;
total_distance = distance1 + distance2
printf("%f\n", total_distance);
```



数据输出函数printf

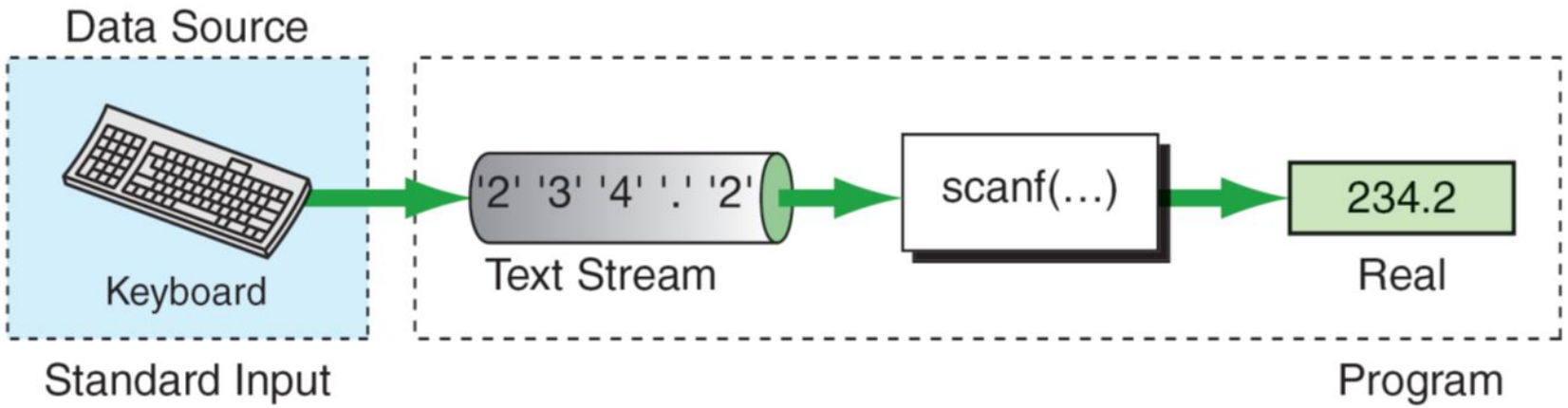


数据输出函数printf

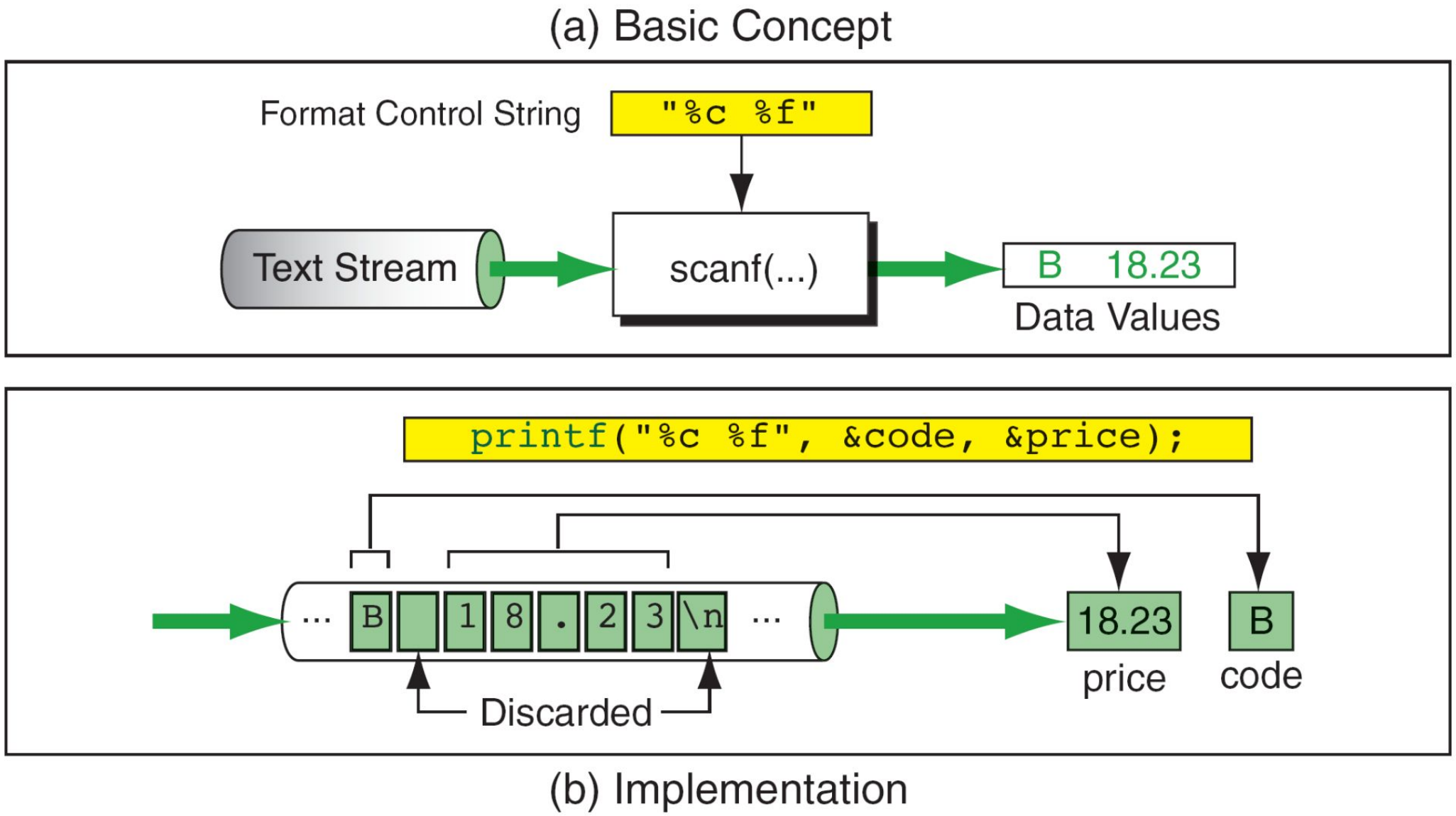


数据输出函数printf

数据输入函数scanf



数据输入函数scanf



数据输入函数scanf

总结

- 计算机程序
- 数值数据
- 符号数据
- 变量

谢谢！