ASCII: The American Standard Code for Information Interchange. An encoding scheme that defines control characters and graphic characters for the first 128 values in a byte.

block comment: A comment beginning with /* and ending with */ that can be used to identify a comment spanning multiple lines.

Boolean: A type whose only permitted values are true and false.

Boolean literal: A 1-byte integer value in which the value 0 is interpreted as false and all other values as true.

built-in data type: A type defined by the language such as an integer.

C-string: A null-terminated array of characters.

C++ string: A string defined using the C++ string class.

character type: A 1-byte integral data type that defines a character in the ASCII encoding system.

character literal: An alphabet character enclosed in two single quotes.

class: The C++ mechanism for defining a new data type.

comment: Text in a C++ program that is ignored by the compiler.

constant: A value that cannot be changed during execution of a program.

data type: A named set of values and operations defined to manipulate them, such as character and integer.

double: The C++ type for double-precision floating-point type.

double quote: The delimiters used to enclose a string literal.

enumerated: The definition of an optional type name and a set of zero or more identifiers.

floating-point: A number that contains both an integral and a fraction.

floating-point literal: A numeric value with a fraction.

fundamental data type: One of five built-in data types in the C++ language: integer, character, boolean, floating-point, and void.

identifier: A sequence of characters that define a name.

include: A pre-processor command that specifies a library file to be inserted into the program.

integer: An integral number; a number without a fractional part.

keyword: Any specified word that has a defined syntactical meaning to a compiler.

line commend: A comment, beginning with //, that must be completed on one line.

literal: An unnamed constant coded as a part of an expression.

new line: The ASCII character that designates the end of a line of input or output.

preprocessor: A program that runs before the C++ compiler.

single quotes: The literal delimiter for a character literal.

string class: The C++ built-in type that defines the class templates and functions that support the string type.

token: In C++, a syntactical construct that represents an operation or a flag, such as the assignment token (=).

value: The contents of a variable.

variable: A named object or reference.

The C++ language is case-sensitive. We normally divide our programs into lines to improve readability.

The header file in a program includes predefined code at the top of the program. Every C++ program needs at least one function named main .

A program in C++ uses variables, which have a name and a type .

The cin and cout objects we use in a program are closely related to the ideas of variable and value. The cin object gets data from a keyboard and stores it in a variable; the cout object sends a value to the monitor.

The source code in a C++ program is usually made of two things interleaved with each other: code and comments.

The code uses the *tokens* of the C++ language;

The comments are words in English. A *token* can be an *identifier*, a *literal*, or a *symbol*.

A program written in the C++ language manipulates data. C++ recognizes different types of data: *built-in* (*fundamental and compound*) and *user-defined* (*enumerated and class*).

The fundamental data types are divided into three categories: integer, character, and floating-point type.

An integer is a number without a fraction.

A character data type defines a letter.

The Boolean data type represents a true or false value.

The floating-point type represents a number with a fraction.

The C++ language uses two string types: the first is called C-string; the second is called C++ string.

CP-1. Do we need a semicolon at the end of the main function (after the closing brace)?

No.

CP-2. Distinguish the range of a line comment and a block comment

To insert a line comment (//) token is needed. Only single line is ignored by the compiler

A block comment is one or more lines that are considered comments for the user and needs (/* ... */) tokens.

CP-3. Explain the purpose of the include directive in a C++ program.

A preprocessor directive is a command to the compiler to take some action before it compiles the program.

Note that no semicolon should be put after any include directive. The compiler may generate an error if it finds one.

```
#include <iostream>
```

CP-4. What is the use of the "using namespace" command in a program?

Last name of the object can be written at the beginning of the program. So we do not need to write it later in the code.

CP-5. Present a table of keywords.

Table 2.1 Keywords

alignas	alignof	and	and_eq	asm
auto	bitand	bitor	bool	break
case	catch	char	char16_t	char32_t
class	compl	const	const_cast	constexpr
continue	decltype	default	delete	do
double	dynamic_cast	else	enum	explicit
export	extern	false	float	for
friend	goto	if	inline	int
long	mutable	namespace	new	noexcept
not	not_eq	nullptr	operator	or
or_eq	private	protected	public	register
reinterpret_cast	return	short	signed	sizeof
static	static_assert	static_cast	struct	switch
template	this	thread_local	throw	true
try	typedef	typeid	typename	union
unsigned	using	virtual	void	volatile
wchar_t	while	xor	xor_eq	

CP-6. Present a table of symbols.

Table 2.2 Symbols in C++

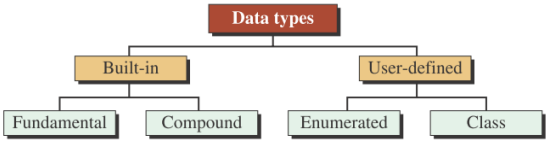
{	}	[	]	#	##	(	)	<:	:>
<%	%>	%:	%:%:	;	:	...	,	.	+
–	*	/	%	^	&	|	?	::	.*
–>	–>*	~	!	=	==	<	>	<=	>=
!=	<<	>>	<<=	>>=	-=	+=	*=	/=	%=
&=	!=	^=	++	– –					

CP-7. Examples of predefined identifiers

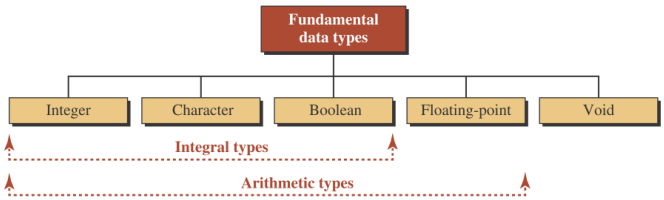
include, iostream, main, std, cout, endl

CP-8. What is the difference between built-in types and user-defined types?

The **built-in** data types are those defined by the language



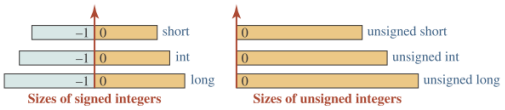
CP-9. List the fundamental data types.



CP-10. Define three different sizes of the integer type.

Table 2.3 Ranges of integers in a typical machine

Type	Sign	Range	
short int	signed	–32,768	+32,767
	unsigned	0	65,536
int	signed	–2,147,483,648	+2,147,483,647
	unsigned	0	4,294,967,295
long int	signed	–2,147,483,648	+2,147,483,647
	unsigned	0	4,294,967,295



CP-11. Which of the integer sizes cannot be used as a literal?

short int is not used for literal

CP-12. What suffixes should be used to declare integer literal?

Table 2.4 Suffixes for explicit size and sign of an integer literal

Size	*Suffixes*	*Sign*	*Suffixes*
int	None	signed int or signed long	None
long	l or L	unsigned int or unsigned long	u or U

To explicitly define the size or sign of an integer literal, we use suffixes.

CP-13. Is there such a thing as unsigned floating-point types?

No. All floating-point numbers are signed. The default size of a floating-point literal is double.

CP-14. What suffixes do we need to add to a number to declare a float, a double, or a long double?

Table 2.6 Suffixes to define the size of a literal

Floating-point type	*Suffixes*	*Example*
float	f or F	12.23F, 12345.45F, –1436F
double	None	1425.36, 1234.34, 123454
long double	l or L	2456.23L, 143679.00004 L, –0.02345L

CP-15. Can we declare a variable of type void?

No. The void type has no value. Used to show the lack of any value.

CP-16. Distinguish between a C string and a C++ string.

C string need null character at the end, whereas C++ does not. You can use operations on C++ string objects (like concatenating)

CP-17. Distinguish between the two literals 'a' and "a".

A character literal is always enclosed in a pair of single quotes.

A string is form by a pair of double quotes.

CP-18. What header do we need to add to our program to use C++ strings?

```
#include <string>
```