Language Specification – C++

Table of Contents

Language Specification – C++

Data types

Data type	Bytes	Range
		Integer types
char	1	-128 to 127
unsigned char	1	0 to 255
short	2	-32,768 to 32,767
unsigned short	2	0 to 65,535
int	4	-2,147,483,648 to 2,147,483,647
unsigned int	4	0 to 4,294,967,295
long	4	-2,147,483,648 to 2,147,483,647
unsigned long	4	0 to 4,294,967,295
long long	8	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
unsigned long long	8	0 to 18,446,744,073,709,551,615
		Real-number types
float	4	-3.4^{38} to -1.2^{-38}, 0, 1.2^{-38} to 3.4^{38}
double	8	-1.8^{308} to -2.2^{-308}, 0, 2.2^{-308} to 1.8^{308}
long double	8	-1.8^{308} to -2.2^{-308}, 0, 2.2^{-308} to 1.8^{308}
		Other types
bool	1	true or false
char	1	A single character, enclosed by single quotes
string	28	A sequence of characters, enclosed by double quotes

String and character literals

String literal

- A string literal is a sequence of zero or more characters enclosed in double quotes.

Character literal

- A character literal is one character enclosed in single quotes.

Identifier rules

- An identifier:
  - ✓ Begins with a letter or underscore.
  - ✓ Is followed by zero or more letters, digits, and underscores.

Variable declarations

```
<data-type> <variable-name>;
OR
<data-type> <variable-name> = <initial-value>;
```

Constant declaration

```
const <data-type> <constant-name> = <initial-value>;
```

Array declarations

```
<data-type> <array-name>[<integer-expression>];
OR
<data-type> <array-name>[] = {<initializer-list};
```

Other declarations

Pointer variable

```
<data-type> *<variable-name>;
OR
<data-type> *<variable-name> = <initial-value>;
```

struct type

```
struct <type-name>
{
     <variable-or-array-declaration-1>;
     <variable-or-array-declaration-2>;
     …
     <variable-or-array-declaration-n>;
};
```

Keyboard input

```
cin >> <variable>;
OR
cin.get(<char-variable>);
OR
cin.getline(<char-array-variable>, <integer-expression>);
OR
getline(cin, <string-variable>);
```

Screen output

```
cout << <expression-1> << <expression-2> << … << <expression-n>;
```

Escape sequences

Sequence	Purpose
\?	Output a question mark; used to prevent trigraphs from being interpreted inside string literals.
\\	Output a backslash.
\’	Output a single quote.
\”	Output a double quote.
\a	Beep user.
\b	Backspace cursor.
\n	Move cursor to start of next line.
\r	Move cursor to start of current line.
\t	Tab on the current line.

Assignment statement

```
<variable> = <expression>;
```

Arithmetic operators

Operator	Purpose
+	Addition
–	Subtraction
*	Multiplication
/	Division
%	Modulo (integer remainder)

Relational operators

Operator	Purpose
==	Equal to
!=	Not equal to
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to

Logical operators

Operator	Purpose
!	Not
&&	And
||	Or

Conditionals

if statement

```
if (<condition>)
{
      <block>
}
else if (<condition>)
{
      <block>
}
else
{
      <block>
}
```

<block> is one or more statements. If there is only one statement, then the curly braces may be omitted.

switch statement

```
switch (<expression>)
{
      case <literal-1>:
            <block>;
            break;
      case <literal-2>:
            <block>;
            break;
      …
      case <literal-n>:
            <block>;
            break;
      default:
            <block>;
}
```

<block> is one or more statements.

Iteration

while statement

```
while (<condition>)
{
	<block>
}
```

do-while statement

```
do
{
	<block>
}
while (<condition>);
```

for statement

```
for (<initialization>; <condition>; <update>)
{
	<block>
}
```

<block> is one or more statements. If there is only one statement, then the curly braces may be omitted.

Reserved words

alignas
alignof
and
and_eq
asm
atomic_cancel
atomic_commit
atomic_noexcept
auto
bitand
bitor
bool
break
case
catch
char
char16_t
char32_t
class
compl
concept
const
constexpr
const_cast
continue
decltype
default
delete
do
double
dynamic_cast
else
enum
explicit
export
extern
false
float
for
friend
goto
if
import
inline
int

Language Specification – C++

long
module
mutable
namespace
new
noexcept
not
not_eq
nullptr
operator
or
or_eq
private
protected
public
register
reinterpret_cast
requires
return
short
signed
sizeof
static
static_assert
static_cast
struct
switch
synchronized
template
this
thread_local
throw
true
try
typedef
typeid
typename
union
unsigned
using
virtual
void
volatile
wchar_t
while
xor
xor_eq