

Representation:

A variable of type **char** can store a single member from the set of ASCII (American Standard Code for Information Interchange).

A variable of type **wchar_t** can store a wide character occupying 16 bits.

Characters are integer codes. Let C_8 be the set of all characters represented by type **char**.

$$C_8 = \{c \in C_8 \mid 0 \leq c \leq 2^8 - 1\}$$

Let C_{16} be the set of all characters represented by the type **wchar_t**.

$$C_{16} = \{c \in C_{16} \mid 0 \leq c \leq 2^{16} - 1\}$$

A character is distinguished from an integer only when it is printed or displayed. Instead of printing or displaying the integer code, the character face is printed or displayed.

Decimal Code	ASCII Character	Decimal Code	ASCII Character	Decimal Code	ASCII Character	Decimal Code	ASCII Character
0	NUL	32	SP	64	@	96	`
1	SOH	33	!	65	A	97	a
2	STX	34	"	66	B	98	b
3	ETX	35	#	67	C	99	c
4	EOT	36	\$	68	D	100	d
5	ENQ	37	%	69	E	101	e
6	ACK	38	&	70	F	102	f
7	BEL	39	'	71	G	103	g
8	BS	40	(	72	H	104	h
9	HT	41	)	73	I	105	i
10	LF	42	*	74	J	106	j
11	VT	43	+	75	K	107	k
12	FF	44	,	76	L	108	l
13	CR	45	-	77	M	109	m
14	SO	46	.	78	N	110	n
15	SI	47	/	79	O	111	o
16	DLE	48	0	80	P	112	p
17	DC1	49	1	81	Q	113	q
18	DC2	50	2	82	R	114	r
19	DC3	51	3	83	S	115	s
20	DC4	52	4	84	T	116	t
21	NAK	53	5	85	U	117	u
22	SYN	54	6	86	V	118	v
23	ETB	55	7	87	W	119	w
24	CAN	56	8	88	X	120	x
25	EM	57	9	89	Y	121	y
26	SUB	58	:	90	Z	122	z
27	ESC	59	;	91	[	123	{
28	FS	60	<	92	\	124	
29	GS	61	=	93	]	125	}
30	RS	62	>	94	^	126	~
31	US	63	?	95	_	127	DEL

Figure 1. ASCII Character Set

Declaration syntax:

Variables *firstinitial* and *lastinitial* in program **p01** have type `char`.

Example: Variables, *firstinitial* and *lastinitial* in the program shown in Figure 2 are character variables.

```
#include <iostream>
using namespace std;
int main()
{   cout << "What is your first initial? ";
    char firstinitial;
    cin >> firstinitial;
    cout << "What is your last initial? ";
    char lastinitial;
    cin >> lastinitial;
    cout << firstinitial << "." << lastinitial << "., you are a genius.";
    cout << endl;
    return 0;
}
```

Figure 2. Program **p01.cpp** containing character variables *firstinitial* and *lastinitial*.

Program output:

What is your first initial? **A**

What is your last initial? **E**

A. E., you are a genius.

All characters are represented as integer codes. The program in Figure 3 illustrates how the printing characters of the ASCII character set can be printed.

```
//-----
//File p02.cpp prints the printable ASCII table
//-----
#include <iostream>
#include <iomanip>
using namespace std;
int main()
{
    for (int a=32,b=64,c=96;a<64;a++,b++,c++) {
        cout << setw(4) << a;
        cout << setw(4) << (char)a;
        cout << setw(4) << b;
        cout << setw(4) << (char)b;
        cout << setw(4) << c;
        cout << setw(4) << (char)c;
        cout << endl;
    }
    return 0;
}
```

Figure 3. program **p02.cpp** illustrates how ASCII characters and their codes can be printed.

Program output:

```

32      64      @      96      `
33      !      65      A      97      a
34      "      66      B      98      b
35      #      67      C      99      c
36      $      68      D     100      d
37      %      69      E     101      e
38      &      70      F     102      f
39      '      71      G     103      g
40      (      72      H     104      h
41      )      73      I     105      i
42      *      74      J     106      j
43      +      75      K     107      k
44      ,      76      L     108      l
45      -      77      M     109      m
46      .      78      N     110      n
47      /      79      O     111      o
48      0      80      P     112      p
49      1      81      Q     113      q
50      2      82      R     114      r
51      3      83      S     115      s
52      4      84      T     116      t
53      5      85      U     117      u
54      6      86      V     118      v
55      7      87      W     119      w
56      8      88      X     120      x
57      9      89      Y     121      y
58      :      90      Z     122      z
59      ;      91      [     123      {
60      <      92      \     124      |
61      =      93      ]     125      }
62      >      94      ^     126      ~
63      ?      95      _     127      ¯

```

Constants.

character-constant:

‘ *c-char-sequence* ‘

L ‘ *c-char-sequence* ‘

c-char-sequence:

c-char

c-char-sequence c-char

c-char :

any source character except the apostrophe (‘), backslash (\), or newline

escape-character

Character Constant	Integer Code	Character Constant	Integer Code	Character Constant	Integer Code
‘a’	97	‘A’	65	‘ ‘	32
‘?’	63	‘\r’	13	‘\0’	0
‘”’	34	‘\377’	255	‘%’	37
‘8’	56	‘\\’	92	‘\’	39

Name	ASCII Name	C++ Name	Octal	Hexadecimal	Decimal
newline	NL (LF)	<code>'\n'</code>	<code>'\12'</code>	<code>'\xA'</code>	10
horizontal tab	HT	<code>'\t'</code>	<code>'\11'</code>	<code>'\x9'</code>	9
vertical tab	VT	<code>'\v'</code>	<code>'\13'</code>	<code>'\xB'</code>	11
backspace	BS	<code>'\b'</code>	<code>'\10'</code>	<code>'\x8'</code>	8
carriage return	CR	<code>'\r'</code>	<code>'\15'</code>	<code>'\xD'</code>	13
form feed	FF	<code>'\f'</code>	<code>'\14'</code>	<code>'\xC'</code>	12
alert	BEL	<code>'\a'</code>	<code>'\7'</code>	<code>'\x7'</code>	7
backslash	<code>\</code>	<code>'\\'</code>	<code>'\134'</code>	<code>'\x5C'</code>	92
question mark	<code>?</code>	<code>'\?'</code>	<code>'\77'</code>	<code>'\x3F'</code>	63
apostrophe	<code>'</code>	<code>'\''</code>	<code>'\47'</code>	<code>'\x27'</code>	39
quotation mark	<code>"</code>	<code>'\"'</code>	<code>'\42'</code>	<code>'\x22'</code>	34
octal number	<i>ooo</i>		<code>'\ooo'</code>		
hexadecimal number	<i>hhh</i>			<code>'\xhhh'</code>	

Table 1. Escape Characters