

KEYWORDS

asm*	Identifies following statement as inline assembly language.
auto	Default function variable declaration (not normally used).
break	Skips to end of current loop. Usually used with 'switch'.
case	Control statement used with 'switch'.
catch*	Control statement used with 'try' and 'throw' keywords to provide exception handling.
char	Character variable type (1 byte).
class*	Identifies a class definition.
const	Qualifier that establishes a variable whose value cannot be modified.
continue	Control statement used with 'while', 'do', and 'for'. Causes next iteration of loop to execute immediately.
default	Control statement used with 'switch'.
delete*	Deallocates memory assigned to object.
direct**	Used with some compilers to identify a class referenced by a pointer.
do	Control statement.
double	Floating point variable type. Minimum of 10 digits (decimal) precision.
else	Control statement used with 'if'.
enum	Variable type. Lets you declare symbolic names to represent integer constants. Used to improve readability.
extern	Qualifier that identifies a variable as being defined elsewhere in program.
float	Floating point variable type. Minimum of 6 digits (decimal) precision.
for	Control statement.
friend*	Function and class specifier that allows access to objects private data.
goto	Control statement (not normally used).
if	Control statement.
indirect**	Used with some compilers to identify a class referenced by a handle.
inherited**	Used with some compilers to access methods of the immediate superclass.

inline*	Function identifier that tells compiler to treat function as an inline function.
int	Integer variable type (minimum range is +/-32767).
long	Integer variable type (minimum range is +/-2147483647).
new*	Allocates memory for an object and returns a pointer.
operator*	Function used to overload existing operators.
private*	Default access specifier for class members. Prevents access outside of class.
protected*	Access specifier which allows access within class as well as derived classes.
public*	Access specifier which allows access by functions outside of class.
register	Register variable qualifier. Requests compiler to use CPU registers to store variable.
return	Return variable to calling function.
short	Integer variable type (minimum range is +/-32767).
signed	Integer variable type (or qualifier when used in conjunction with an integer other than an 'int'). Identifies that variable can store both positive and negative numbers.
sizeof	Returns size of variable in bytes.
static	Function variable qualifier. Maintains values between function calls.
struct	Identifies a structure definition.
switch	Control statement. Uses 'case', 'break', and 'default'.
template*	Used to define template functions and classes which use multiple data types.
this*	Pointer sent to member function that contains the address of the object.
throw*	Control statement used with 'catch' and 'try' keywords to provide exception handling.
try*	Control statement used with 'catch' and 'throw' keywords to provide exception handling.
typedef	Defines creation of new data type (or assign new name to existing data type).
union	Identifies a union definition. Similar definition as structure, except different data types are stored in same memory space (but not simultaneously).
unsigned	Integer variable type (or qualifier when used in conjunction with an integer other than an 'int'). Identifies that

variable can only store positive numbers.

virtual* Method qualifier that tells compiler to bind function at runtime. Also used to identify virtual classes when using multiple inheritance.

void Null data type.

volatile Variable qualifier that identifies data which can be altered by agencies other than the program.

while Control statement used with 'do'.

* C++ specific keywords (additions to straight C).

** Additional keywords available with Symantec THINK C.