COMPILER GENERATED ERROR MESSAGES

Below is a list of the error messages that the C compiler generates, and, if applicable, probable causes and the K & R Appendix A section number (in parenthesis) to see for more specific information.

already a local variable
: Variable has already been declared at the current block level. (8.1, 9.2)

argument : <text>
: Error from preprocessor. Self-explanatory. Most common cause of this error is not being able to find an include file.

argument error
: Function argument declared as type struct, union, or function. Pointers to such types, however, are allowed. (10.1)

argument storage
: Function arguments may only be declared as storage class register. (10.1)

bad character
: A character not in the C character set (probably a control char) was encountered in the source file. (2)

both must be integral
: >> and << operands cannot be FLOAT or DOUBLE. (7.5)

break error
: The break statement is allowed only inside a while, do, for, or switch block. (9.8)

can't take address
: & operator is not allowed on a register variable. Operand must otherwise be an lvalue. (7.2)

cannot cast
: Type result of cast cannot be FUNCTION or ARRAY. (7.2, 8.7)

cannot evaluate size
: Could not determine size from declaration or initializer. (8.6, 14.3)

cannot initialize
: Storage class or type does not allow variable to be initialized. (8.6)

compiler trouble
: Compiler detected something it couldn't handle. Try compiling the program again. If this error still occurs, contact Radio Shack.

condition needed
: While, do, for, switch, and if statements require a condition expression. (9.3)

constant expression required
: Initializer expressions for static or external variables cannot reference variables. They may, however, refer to the address of a previously declared variable. This installation allows no initializer expressions unless all operands are of type INT or CHAR (8.6)

constant overflow
: Input numeric constant was too large for the implied or explicit type. (2.6, [PDP-11])

constant required
: Variables are not allowed for array dimensions or cases. (8.3, 8.7, 9.7)

continue error
: The continue statement is allowed only inside a while, do, or for block. (9.9)

declaration mismatch
: This declaration conflicts with a previous one. This is typically caused by declaring a function to return a non-integer type after a reference has been made to the function. Depending on the line structure of the declaration block, this error may be reported on the line following the erroneous declaration. (11, 11.1 11.2)

divide by zero
: Divide by zero occurred when evaluating a constant expression.

? expected
: ? is any character that was expected to appear here. Missing semicolons or braces cause this error.

expression missing
: An expression is required here.

function header missing
: Statement or expression encountered outside a function. Typically caused by mismatched braces. (10.1)

function type error
: A function cannot be declared as returning an array, function, struct, or union. (8.4, 10.1)

function unfinished
: End-of-file encountered before the end of function definition. (10.1)

identifier missing
: Identifier name required here but none was found.

illegal declaration
Declarations are allowed only at the beginning of a block. (9.2)

label required
Label name required on goto statement. (9.11)

label undefined
Goto to label not defined in the current function. (9.12)

lvalue required
Left side of assignment must be able to be "stored into". Array names, functions, structs, etc. are not lvalues. (7.1)

multiple defaults
Only one default statement is allowed in a switch block. (9.7)

multiple definition
Identifier name was declared more than once in the same block level (9.2, 11.1)

must be integral
Type of object required here must be type int, char, or pointer.

name clash
Struct-union member and tag names must be mutually distinct. (8.5)

name in a cast
Identifier name found in a cast. Only types are allowed. (7.2, 8.7)

named twice
Names in a function parameter list may appear only once. (10.1)

no 'if' for 'else'
Else statement found with no matching if. This is typically caused by extra or missing braces and/or semicolons. (9.3)

no switch statement
Case statements can only appear within a switch block. (9.7)

not a function
Primary in expression is not type "function returning...". If this is really a function call, the function name was declared differently elsewhere. (7.1)

not an argument
Name does not appear in the function parameter list. (10.1)

operand expected
Unary operators require one operand, binary operators two. This is typically caused by misplaced parenthesis, casts or

operators. (7.1)

out of memory
Compiler dynamic memory overflow. The compiler requires dynamic memory for symbol table entries, block level declarations and code generation. Three major factors affect this memory usage. Permanent declarations (those appearing on the outer block level (used in include files)) must be reserved from the dynamic memory for the duration of the compilation of the file. Each { causes the compiler to perform a block-level recursion which may involve "pushing down" previous declarations which consume memory. Auto class initializers require saving expression trees until past the declarations which may be very memory-expensive if they exist. Avoiding excessive declarations, both permanent and inside compound statement blocks, conserve memory. If this error occurs on an auto initializer, try initializing the value in the code body.

pointer mismatch
Pointers refer to different types. Use a cast if required. (7.1)

pointer or integer required
A pointer (of any type) or integer is required to the left of the '->' operator. (7.1)

pointer required
Pointer operand required with unary * operator. (7.1)

primary expected
Primary expression required here. (7.1)

should be NULL
Second and third expression of ?: conditional operator cannot be pointers to different types. If both are pointers, they must be of the same type or one of the two must be null. (7.13)

**** STACK OVERFLOW ****
Compiler stack has overflowed. Most likely cause is very deep lock-level nesting or hundreds of switch cases.

storage error
Reg and auto storage classes may only be used within functions. (8.1)

struct member mismatch
Identical member names in two different structures must have the same type and offset in both. (8.5)

struct member required
Identifier used with . and -> operators must be a structure member name. (7.1)

struct syntax
Brace, comma, etc. is missing in a struct declaration. (8.5)

struct or union inappropriate
Struct or union cannot be used in this context.

syntax error
Expression, declaration, or statement is incorrectly formed.

third expression missing
? must be followed by a : with expression. This error may be caused by unmatched parenthesis or other errors in the expression. (7.13)

too long
Too many characters provided in a string initializing a character array. (8.6)

too many brackets
Unmatched or unexpected brackets encountered processing an initializer. (8.6)

too many elements
More data items supplied for aggregate level in initializer than members of the aggregate. (8.6)

type error
Compiler type matching error. Should never happen.

type mismatch
Types and/or operators in expression do not correspond. (6)

typedef - not a variable
Typedef type name cannot be used in this manner. (8.8)

undeclared variable
No declaration exists at any block level for this identifier.

undefined structure
Union or struct declaration refers to an undefined structure name. (8.5)

unions not allowed
Cannot initialize union members. (8.6)

unterminated character constant
Unmatched ' character delimiters. (2.4.3)

unterminated string
Unmatched " string delimiters. (2.5)

while expected
No while found for do statement. (9.5)