Summary

1 Tokens are the basic syntactic units of C. They include keywords, identifiers, constants, string constants, operators, and punctuators. White space, along with operators and punctuators, can serve to separate tokens. For this reason, white space, operators, and punctuators are collectively called separators. White space, other than serving to separate tokens, is ignored by the compiler.

2 Comments are enclosed by the bracket pair /* and */ and are treated as white space by the compiler. They are critical for good program documentation. Comments should assist the reader to both use and understand the program.

3 A keyword, also called a reserved word, has a strict meaning. There are 32 keywords in C. They cannot be redefined.

4 Identifiers are tokens that the programmer uses chiefly to name variables and functions. They begin with a letter or underscore and are chosen to be meaningful to the human reader.

5 Some identifiers are already known to the system because they are the names of functions in the standard library. These include the input/output functions scanf() and printf() and mathematical functions such as sqrt(), sin(), cos(), and tan().

6 Constants include various kinds of integer and floating constants, character constants such as 'a' and '#', and string constants such as "abc". All constants are collected by the compiler as tokens.

7 String constants such as "deep blue sea" are arbitrary sequences of characters, including white-space characters, that are placed inside double quotes. A string constant is stored as an array of characters, but it is collected by the compiler as a single token. The compiler provides the space in memory needed to store a string constant. Character constants and string constants are treated differently. For example, 'x' and "x" are not the same.

8 Operators and punctuators are numerous in C. The parentheses that follow main in the code

```
int main(void)
{
```

constitute an operator; they tell the compiler that main is a function. The parentheses in the expression a * (b + c) are punctuators. The operations inside the parentheses are done first.

9 In C, the rules of precedence and associativity for operators determine how an expression gets evaluated. The programmer needs to know them.

IO The increment operator ++ and the decrement operator - have a side effect. In addition to having a value, an expression such as ++i causes the stored value of i in memory to be incremented by 1.

11 The operators ++ and - can be used in both prefix and postfix positions, possibly with different effects. The expression ++i causes i to be incremented in memory, and the new value of i is the value of the expression. The expression i++ has as its value the current value of i, and then the expression causes i to be incremented in memory.

12 In C, the assignment symbol is an operator. An expression such as a= b + c assigns the value of b + c to a, and the expression as a whole takes on this value. Although the assignment operator in C and the equal sign in mathematics look alike, they are not comparable.

13 The standard library contains many useful functions. If the programmer uses a function from the standard library, then the corresponding standard header file should be included. The standard header file provides the appropriate function prototype.