

Glossary

A

abstract class—a class from which concrete objects cannot be instantiated but can serve as a basis for inheritance. Contrast with *concrete class*.

abstract method—a method that contains no statements and must be implemented by classes that derive from its class.

access key—a shortcut way to make a selection using the keyboard.

access specifier—a keyword that dictates which outside classes can use a method.

accessibility—describes limitations on how a method can be used.

accessors—methods in properties that specify how a class's fields are accessed. See *get accessors* and *set accessors*.

accumulated—describes totals that are added into a final sum by processing individual records one at a time in a loop.

actual parameters—arguments within a method call.

add and assign operator (+=)—an operator that adds the operand on the right

to the operand on the left and assigns the results to the operand on the left in one step.

alias—an alternative name or pseudonym.

ambiguous—describes overloaded methods between which the compiler cannot distinguish.

American Standard Code for Information Interchange (ASCII)—an eight-bit code that represents characters.

ancestors—all the superclasses from which a subclass is derived.

AND operator—an operator that determines whether two expressions are both true; it is written using two ampersands (&&). Also called the *conditional AND operator*. Contrast with *Boolean logical AND operator*.

applicable methods—the collection of potential methods that could be used by a method call.

application classes—classes that contain a `Main()` method and are executable programs.

application software—programs that allow users to complete tasks, in contrast with

system software that the computer needs to complete tasks.

argument—the expression passed to a method.

arithmetic operators are used to perform arithmetic; they include +, -, *, /, and %.

array—a list of data items that all have the same data type and the same name, but are distinguished from each other by a subscript or index.

array element—one object in an array.

assembly—a group of code modules compiled together to create an executable program.

assignment—a statement that provides a variable with a value.

assignment operator—the equal sign (=); any value to the right of the assignment operator is assigned to, or taken on by, the variable or constant to the left.

associativity—specifies the order in which a sequence of operations with the same precedence are evaluated.

at runtime—a phrase describing actions that occur during a program's execution.

attributes of an object—the characteristics of an object.

auto-implemented property—a property in which the code within the accessors is created automatically. The only action in the set accessor is to assign a value to the associated field, and the only action in the get accessor is to return the associated field value.

automatic properties—auto-implemented properties.

B

backing field—a field that has a property coded for it.

base 2 numbers—binary system numbers.

base 10 numbers—decimal system numbers.

base 16 system—a mathematical system that uses 16 symbols to represent numbers; hexadecimal.

base class—a class that is used as a basis for inheritance.

behaviors of an object—the methods associated with an object.

betterness rules—the rules that determine the best overloaded method to execute based on the arguments in a method call.

binary files—files that can store any of the 256 combinations of bits in any byte instead of just those combinations that form readable text.

binary numbering system—a numbering system that uses two digits, 0 and 1, arranged in columns to represent numbers.

binary operators—operators that use two operands: one value to the left of the operator and another value to the right of it.

binary search—an algorithm that attempts to find an item in a list by splitting the sorted list of objects in half repeatedly as the search gets closer to a match.

bit—a binary digit.

bitwise operators—operators that are used to manipulate the individual bits of values.

black box—any device that can be used without knowing how it works internally.

block—a collection of one or more statements contained within a pair of curly braces.

block comments—comments that start with a forward slash and an asterisk (`/*`) and end with an asterisk and a forward slash (`*/`). Block comments can appear on a line by themselves, on a line before executable code, or after executable code. They can also extend across as many lines as needed. Compare with *line comments*.

bool—a data type that holds a Boolean value.

Boolean logical AND operator—an operator that determines whether two expressions are both true; it is written using a single ampersand (`&`). Unlike the conditional AND operator, it does not use short-circuit evaluation.

Boolean logical inclusive OR operator—an operator that determines whether at least one of two conditions is true; it is written using a single pipe (`|`). Unlike the conditional OR operator, it does not use short-circuit evaluation.

Boolean variable—a variable that can hold only one of two values: true or false.

bugs—program errors.

byte—an integral data type that can hold an unsigned numeric value from 0 through 255.

C

C# programming language—a computer programming language developed as an object-oriented and component-oriented language. It exists as part of Visual Studio, a package used for developing applications for the Windows family of operating systems.

call—to invoke a method.

call stack—the memory location where the computer stores the list of locations to which the system must return after method calls.

called—describes a method that has been invoked.

called method—a method that has been invoked by another method.

calling method—a method that calls another method.

camel casing—a style of creating identifiers in which the first letter is not capitalized, but each new word is. Contrast with *Pascal casing*.

case label—identifies a course of action in a `switch` structure.

catch block—a block of code that can catch one type of `Exception`.

char—an integral data type that can store a character such as 'A', '4', or '\$'.

character—any one of the letters, numbers, or other special symbols (such as punctuation marks) that comprise data.

character set—the group of all the characters used to represent data on a particular computer.

check box—a GUI widget that a user can click to select or deselect an option.

child class—a derived class; a subclass; a class that has inherited from a base class.

class—a category of objects or a type of object.

class access specifier—describes access to a class.

class client—a program or class that instantiates objects of another prewritten class. Also called a *class user*.

class definition—the first class line that describes a class; it contains an optional access specifier, the keyword `class`, and any legal identifier for the name of the class. Also known as *class definition*.

class header—the first class line that describes a class; it contains an optional access specifier, the keyword `class`, and any legal identifier for the name of the class. Also known as *class definition*.

class user—a program or class that instantiates objects of another prewritten class. Also called a *class client*.

click event—an action fired when a user clicks a button in a GUI environment.

client—a method that uses another method.

closing a file—the process of making a file no longer available to an application.

code bloat—describes unnecessarily long or repetitive program statements.

code refactoring—the process of changing a program's internal structure without changing the way the program works.

code snippets—small units of commonly used C# source code that are generated in the IDE using a few keystrokes.

combo box—a GUI element that is a combination of a list box and an editing control that allows a user to select from the list or enter new text.

command line—the operating system's text interface.

command prompt—a request for input that appears at the beginning of the command line.

comment out—to turn a statement into a comment so that the compiler will not execute its command.

comparison operator—an operator that compares two values.

compiler—a computer program that translates high-level language statements into machine code.

completion mode—an IDE editing mode in which suggestions are automatically provided based on context.

composed delegate—a delegate that calls the delegates from which it is built.

composition—the technique of using an object within another object.

computer file—a collection of information stored on a nonvolatile device in a computer system.

computer simulations—programs that attempt to mimic real-world activities to foster a better understanding of them.

concatenate—to join strings together in a chain.

concrete class—a nonabstract class from which objects can be instantiated. Contrast with *abstract class*.

conditional AND operator—an operator that determines whether two expressions are both true; it is written using two ampersands (`&&`). Also called the *AND operator*. Contrast with *Boolean logical AND operator*.

conditional operator—a ternary operator that is used as an abbreviated version of the `if-else` statement; it requires three expressions separated by a question mark and a colon.

conditional OR operator—an operator that determines whether at least one of two conditions is true; it is written using

two pipes (`||`). Also called the *OR operator*. Contrast with *Boolean logical inclusive OR operator*.

constant—describes a data item when it cannot be changed after a program is compiled; in other words, when it cannot vary.

constructor—a method that instantiates (creates an instance of) an object.

constructor initializer—a clause that indicates another instance of a class constructor should be executed before any statements in the current constructor body.

contextual keywords—identifiers that act like keywords in specific circumstances.

control—a GUI component such as a text field, button, or check box that users can manipulate to interact with a program.

control statement—the part of a structure that determines whether the subsequent block of statements executes.

counted loop—a definite loop.

CSV file—a file that contains comma-separated values.

culture—a set of rules that determines how culturally dependent values such as money and dates are formatted.

D

data files—files that contain facts and figures; persistent collections of related records.

data hierarchy—the relationship of characters, fields, records, and files.

data type—a description of the format and size of a data item as well as the operations that can be performed on it.

dead code—statements that can never execute under any circumstances because the program logic “can’t get there.” Also see *unreachable*.

debugging—the process of removing all syntax and logical errors from a program.

decimal—a floating-point data type that has a greater precision and a smaller range than a `float` or `double`, which makes it suitable for financial and monetary calculations.

decimal numbering system—the system that uses 10 digits, 0 through 9, arranged in columns to represent numbers.

decision structure—a unit of program logic that involves choosing between alternative courses of action based on some value.

decrement operator (`--`)—an operator that reduces a variable’s value by 1; there is a prefix and a postfix version.

decrementing—the act of decreasing the value of a variable, often by 1.

default constructor—a constructor that requires no parameters; it can be automatically supplied or written by a programmer.

default event—for a `Control`, the event or method generated when it is double-clicked in the IDE, as well as the event that users expect to generate when they encounter the `Control` in a working application.

default value of an object—the value initialized with a default constructor.

definite loop—a loop in which the number of iterations is predetermined. Also a *counted loop*. Contrast with *indefinite loop*.

delegate—an object that contains a reference to a method.

delimiter—a character used to specify the boundary between characters in text files.

derived class—a subclass; a class that has inherited from a base class.

deserialization—the process of converting streams of bytes back into objects.

design time—the period of time during which a programmer designs a program's interface and writes the code.

destructor—a method that contains the actions performed when an instance of a class is destroyed.

directories—structures used to organize files on a storage device; folders.

do loop—a type of posttest loop; a loop that is tested at the bottom of the loop after one repetition has occurred.

double—a data type that can hold a floating-point number with 15 or 16 significant digits of accuracy.

dual-alternative decisions—decisions that have two possible outcomes.

E

empty body—a block that has no statements in it.

encapsulation—the technique of packaging an object's attributes and methods into a cohesive unit that can be used as an undivided entity.

enhanced for loop—a version of the for loop that cycles through an array without specifying the starting and ending points for the loop control variable.

enumeration—a list of values in which names are substituted for numeric values.

Equals()—a method that determines equivalency. In the `String` class, it is the method that determines if two strings have the same value.

escape sequence—a single character composed of two symbols beginning with a backslash that represents a nonprinting character such as a tab.

event—an object generated when a user interacts with a GUI object, causing the program to perform a task.

event-driven programs—programs that contain code that causes an event such as a button click to perform a task.

event handler—a method that performs a task in response to an event; an event receiver.

event receiver—a method that performs a task in response to an event; an event handler.

event sender—the control that generates an event.

event wiring—the act of connecting an event to its resulting actions.

exception—an error condition or unexpected behavior in an executing program.

exception handling—the set of object-oriented techniques used to manage unexpected errors.

explicit cast—purposefully assigns a value to a different data type; it involves placing the desired result type in parentheses followed by the variable or constant to be cast.

explicitly—purposefully. Contrast with *implicitly*.

exposes—associates a `FileStream` with a file.

Extended Binary Coded Decimal Interchange Code (EBCDIC)—an eight-bit code that is used in IBM mainframe computers.

extended class—a derived class; a child class; a subclass; a class that has inherited from a base class.

extension methods—static methods that act like instance methods.

F

fault-tolerant—describes applications that are designed so that they continue to operate, possibly at a reduced level, when some part of the system fails.

field—in a class, an instance variable. In a file or database, a character or group of characters that has some meaning.

file position pointer—a variable that holds the byte number of the next byte to be read from a file.

finally block—a block of code that optionally follows a `try` block; the code within a `finally` block executes whether the preceding `try` block identifies any `Exceptions` or not.

fires an event—causes an event to occur. Also see *raises an event* and *triggers an event*.

fixed-pitch font—a font in which each character occupies the same width.

float—a data type that can hold a floating-point number with as many as seven significant digits of accuracy.

floating-point—describes a number that contains decimal positions.

flowchart—a tool that helps programmers plan a program's logic by writing program steps in diagram form, as a series of shapes connected by arrows.

focus—the state of a GUI component when the user's attention is drawn to it visually. When a component has focus, its action can be executed by pressing the Enter key.

folders—structures used to organize files on a storage device; directories.

for loop—a loop that contains the starting value for the loop control variable, the test condition that controls loop entry, and the expression that alters the loop control variable, all in one statement.

form—a GUI interface for collecting, displaying, and delivering information.

formal parameter—a parameter within a method header that accepts a value.

format specifier—one of nine built-in format characters in a format string that defines the most commonly used numeric format types.

format string—a string of characters that controls the appearance of output.

fragile—describes classes that depend on field names from parent classes because they are prone to errors; that is, they are easy to “break.”

fully qualified—describes a name that includes the class name as well as the identifier.

G

garbage—an unknown memory value.

get accessors—methods in properties that allow retrieval of a field value by using a property name.

getter—another term for a class property's get accessor.

gigabyte—approximately a billion bytes.

governing type—in a switch statement, the type that is established by the switch expression. The governing type can be sbyte, byte, short, ushort, int, uint, long, ulong, char, string, or enum.

graphical control elements—the components through which user interacts with a GUI program.

graphical user interface (GUI)—an interface that employs graphical images representing controls the user manipulates.

group box—a GUI element that can be used to group controls on a form, it is similar to a panel but it can contain a caption and does not have a scroll bar.

H

hardware—the physical devices associated with a computer.

has-a relationship—the relationship created using composition, so called because one class “has an” instance of another.

hash code—a number that should uniquely identify an object.

hexadecimal numbering system—a mathematical system that uses 16 symbols to represent numbers; base 16.

hide—to override a parent class member in a child class, making the parent class member invisible.

high-level programming language—a language that uses a vocabulary of reasonable terms such as *read*, *write*, or *add* instead of referencing the sequence of on/off switches that perform these tasks.

I

identifier—the name of a program component such as a variable, class, or method.

if statement—a program statement used to make a single-alternative decision.

if-else statement—a statement that performs a dual-alternative decision.

immutable—unchangeable.

implementation hiding—the technique of keeping the details of a method's operations hidden.

implicit cast—the automatic transformation that occurs when a value is assigned to a type with higher precedence.

implicit conversion—the conversion that occurs when a type is automatically changed to another upon assignment.

implicit parameter—an undeclared parameter that gets its value automatically.

implicit reference conversion—the type of conversion that occurs when a derived class object is assigned to its ancestor's data type.

implicitly—automatically. Contrast with *explicitly*.

incrementing—the act of increasing the value of a variable, often by 1.

indefinite loop—a loop in which the number of iterations is not predetermined. Contrast with *definite loop*.

index—an integer contained within square brackets that indicates the position of one of an array's elements. Also see *subscript*.

infinite loop—a loop that (theoretically) never ends.

information hiding—a feature found in all object-oriented languages, in which a class's data is private and changed or manipulated only by its own methods.

inheritance—the ability to extend a class so as to create a more specific class that contains all the attributes and methods of a more general class; the extended class usually contains new attributes or methods as well.

initialization—an assignment made when a variable is declared.

initializer list—the list of values provided for an array.

inner loop—the loop in a pair of nested loops that is entirely contained within another loop.

instance methods—methods that are used with object instantiations.

instance of a class—an object; a tangible example of a class.

instance variables—data components of a class that exist separately for each instantiation. Also called *fields*.

instantiate—to create an object.

instantiation—a created object.

int—an integral data type that can hold a signed numeric value in four bytes.

integers—whole numbers.

integral data types—data types that store whole numbers; the nine integral types are `byte`, `sbyte`, `short`, `ushort`, `int`, `uint`, `long`, `ulong`, and `char`.

Integrated Development Environment (IDE)—a program development environment in which programmers select options from menus or by clicking buttons; an IDE provides helpful features such as color coding and automatic statement completion.

IntelliSense—Microsoft's name for the set of features designed to minimize program development time in the IDE.

interactive program—a program that allows user input.

interface—a collection of abstract methods (and perhaps other members) that can be used by any class as long as the class provides a definition to override the interface's abstract definitions..

intermediate language (IL)—the language into which source code statements are compiled.

internal—a class access modifier that means access is limited to the assembly to which the class belongs.

internal access—a level of method accessibility that limits method access to the containing program.

intrinsic types—basic, built-in data types; C# provides 15 intrinsic types.

invoke—to call a method.

invoking object—the object referenced by `this` in an instance method.

invoking the event—calling an event method.

iteration—one execution of any loop.

iteration variable—a temporary variable that holds each array value in turn in a `foreach` statement.

J

jagged array—a one-dimensional array in which each element is another array.

just in time (JIT)—the C# compiler that translates intermediate code into executable code.

K

key—a value that uniquely identifies a record.

key events—keyboard events that occur when a user presses and releases keyboard keys.

key field—the field used to uniquely identify records and to control the order of records in a sequential file.

keywords—predefined and reserved identifiers that have special meaning to the compiler.

kilobyte—approximately a thousand bytes.

L

label—a control that typically provides descriptive text for another control or displays other text information on a `Form`.

left-associative—describes operators whose operations are performed from left to right.

Length field—a field that contains the number of elements in an array.

lexically—alphabetically.

line comments—comments that start with two forward slashes (`//`) and continue to the end of the current line. Line comments can appear on a line by themselves, or at the end of a line following executable code. Compare with *block comments*.

link label—a GUI control with text and that links the user to other sources, such as Web pages or files.

list box—a GUI element that displays a list of items the user can select by clicking.

literal constant—a value that is taken literally at each use.

literal string—a series of characters that is used exactly as entered.

local—describes a variable that is declared in the current method.

logic—the sequence of statements and methods that produce the desired results in a computer program.

Long—an integral data type that can hold a signed numeric value in eight bytes.

loop—a structure that allows repeated execution of a block of statements.

loop body—the block of statements executed in a loop.

loop control variable—a variable that determines whether loop execution will continue on each iteration.

loop fusion—the technique of combining two loops into one.

lower camel casing—a style of creating identifiers in which the first letter is not capitalized, but each new word is. Also called *camel casing*. Contrast with *Pascal casing*.

M

machine language—the most basic circuitry-level language.

magic number—a hard-coded number.

menu bar—in the Visual Studio IDE, the list of choices that run horizontally across the top of the screen.

mandatory parameters—method parameters for which an argument is required in every method call.

megabyte—approximately a million bytes.

menu strip—a horizontal list of general options that appears under the title bar of a Form or Window.

method—an encapsulated series of statements that carry out a task.

method body—all the instructions contained within a pair of curly braces (`{ }`) following a method header.

method declaration—a method header or definition; it precedes a method and includes a return type, identifier, and an optional parameter list.

method definition—a method header or declaration; it precedes a method and includes a return type, identifier, and an optional parameter list.

method header—the first line of a method, which includes the method name and information about what will pass into and be returned from a method.

method's type—a method's return type.

mission critical—describes any process that is crucial to an organization.

modal window—a secondary window that takes control from a primary window and that a user must deal with before proceeding.

multidimensional arrays—arrays that require multiple subscripts to access the array elements.

multifile assembly—a group of files containing methods that work together to create an application.

multiple inheritance—the ability to inherit from more than one class.

N

named argument—a method argument that is preceded with the name of the called method's parameter to which it will be assigned.

named constant—an identifier whose contents cannot change.

namespace—a scheme that provides a way to group similar classes.

nested if—a statement in which one decision structure is contained within another.

nested method calls—method calls placed inside other method calls.

nibble—four bits; half a byte.

node—In the Visual Studio IDE, a box that appears on a vertical tree to the left of a list or a section of code and that can be expanded or condensed.

non-application classes—classes that do not contain a `Main()` method and therefore are not runnable programs; they provide support for other classes.

nonstatic—describes a method that requires an object reference.

nonvolatile—the type of computer storage that is permanent; it is not lost when a computer loses power.

NOT operator (!)—an operator that negates the result of any Boolean expression.

O

object—a concrete entity that has attributes and behaviors; an object is an instance of a class.

object (or object)—a class type in the `System` namespace that is the ultimate base class for all other types.

object initializer—a clause that assigns values to any accessible members or properties of a class at the time of instantiation without calling a constructor with parameters.

object-oriented approach—an approach to a problem that involves defining the objects needed to accomplish a task and developing classes that describe the objects so that each maintains its own data and carries out tasks when another object requests them.

object-oriented programming (OOP)—a programming technique that features objects, classes, encapsulation, interfaces, polymorphism, and inheritance.

one-dimensional array—an array whose elements are accessed using a single subscript. Also see *single-dimensional array*.

opening a file—the process of creating an object and associating a stream of bytes with it.

operands—the values that operators use in expressions.

operator precedence—rules that determine the order in which parts of a mathematical expression are evaluated. Also called *order of operation*.

optional parameter—a parameter to a method for which a default value is automatically supplied when no argument is supplied; a parameter is optional when it is given a value in the method declaration.

OR operator—an operator that determines whether at least one of two conditions is true; it is written using two pipes (`||`). Also called the *conditional OR operator*. Contrast with *Boolean logical inclusive OR operator*.

order of operation—rules that determine the order in which parts of a mathematical expression are evaluated. Also called *operator precedence*.

orphaned method—a method that never executes in an application, and thus serves no purpose.

out of bounds—describes a subscript that is not within the allowed range for an array.

out of scope—describes a variable that is not usable because it has ceased to exist.

outer loop—the loop in a pair of nested loops that contains another loop.

output parameter—a parameter to a method that receives the argument's address; it is not required to have an initial value. Contrast with *value parameter* and *reference parameter*.

overload resolution—the process of determining which of multiple applicable methods is the best match for a method call.

overloading—using one term to indicate diverse meanings. In C#, methods are overloaded when there are multiple versions with the same name but different parameter lists.

override—a keyword used in method headers when a derived class inherits an abstract method from a parent.

override—to take precedence over another method version.

P

panel—a GUI element that can be used to group controls on a *Form*; similar to a group box, but it does not have a caption and can have a scroll bar.

parallel array—an array that has the same number of elements as another array and holds corresponding data.

parameter array—a local array declared within a method header that can accept any number of elements of the same data type.

parameter list—the data types and parameter names that appear between parentheses in a method header.

parameter to a method—an object or reference that is declared in a method definition.

parameterless constructor—a constructor that takes no parameters; one that is called using no arguments.

parent class—a base class; a superclass; a class that is used as a basis for inheritance.

parse—to break an item into component parts.

Pascal casing—a style of creating identifiers in which the first letter of all new words in a variable name, even the first one, is capitalized. Contrast with *camel casing*.

passed by reference—describes how an argument is passed to a method when the method receives its memory address.

passed by value—describes how data is passed to a method when the method receives a copy of the argument passed to it.

path—the disk drive in which a file resides plus the complete hierarchy of directories.

permanent storage devices—hardware such as hard disks, USB drives, and compact discs that are used to store files.

persistent—describes storage that is nonvolatile.

picture box—a GUI element that displays graphics

polymorphism—the ability to create methods that act appropriately depending on the context.

populating an array—the act of providing values for all the elements in an array.

positional argument—an unnamed method argument that is assigned to a parameter list based on its position in the method call.

postfix increment operator—an operator that evaluates a variable and then adds 1 to it. This operator is represented by ++ after a variable.

posttest loop—a loop in which the loop control variable is tested after the loop body executes. Contrast with *pretest loop*.

precedence—a feature of operators that controls the order in which they are evaluated in an expression.

precision specifier—a specifier that controls the number of significant digits or zeros to the right of the decimal point in a format string.

prefix increment operator—an operator that increases the variable's value by 1 and then evaluates it. This operator is represented by ++ before a variable.

pretest loop—a loop in which the loop control variable is tested before the loop body executes. Contrast with *posttest loop*.

priming read—an input statement that gets a first data item or record.

private—an access specifier that indicates other classes may not use the method or variable that it modifies. When used as a class access specifier, it means access is limited to another class to which the class belongs.

private access—a level of method accessibility that limits method access to the containing class.

procedural program—a program created by writing a series of steps or operations to manipulate values.

program—a set of written computer instructions.

program comments—nonexecuting statements that document a program.

prompt—an instruction to the user to enter data.

propagating an exception—the act of transmitting an exception object unchanged through the call stack.

property—a member of a class that provides access to a field of a class; properties define how fields will be set and retrieved. The value of an object.

proportional font—a font in which different characters have different pitches or widths.

protected—a keyword that provides an intermediate level of security between **public** and **private** access.

protected access—a level of method accessibility that limits method access to the containing class or types derived from the containing class.

pseudocode—a tool that helps programmers plan a program's logic by writing plain English statements.

public—an access specifier that indicates other classes may use the method or variable it modifies.

public access—a level of method accessibility that allows unlimited access to a method by any class.

R

radio buttons—GUI widgets, similar to check boxes, except that when they are placed on a form, only one can be selected at a time and selecting any one automatically deselects the others.

raises an event—causes an event to occur. Also see *fires an event* and *triggers an event*.

random access file—a file from which records can be accessed in any order.

random access memory (RAM)—temporary storage in a computer.

range check—a series of statements that determine whether a value falls within a specified range.

range match—a process that determines whether a value falls between a pair of limiting values.

read from the file—to copy data from a file on a storage device into RAM.

read-only—describes a value, property, or file that can be accessed but not altered.

read-only property—a property that has only a get accessor, and not a set accessor.

record—a collection of fields that contain data about an entity.

rectangular array—an array in which each row has the same number of columns.

recursive—describes a method that calls itself.

reference equality—the type of equality that occurs when two reference type objects refer to the same object.

reference parameter—a parameter to a method that receives the argument's address; it is required to have an initial value. Contrast with *value parameter* and *output parameter*.

reference type—a data type that holds a memory address. Contrast with *value types*.

rethrow an Exception—to throw a caught `Exception` instead of handling it.

return statement—a method statement that causes a value to be sent back from a method to its calling method.

return type—the type of value a method will return to any other method that calls it.

right-associative—describes operators whose operations are performed from right to left.

robustness—describes the degree to which a system is resilient to stress, maintaining correct functioning even in the presence of errors.

root class—the ultimate or first base class in a hierarchical ancestry tree.

root directory—the main directory of a storage device.

run time—the period of time during which a program executes.

runnable—describes files that are executable.

S

sbyte—an integral data type that can hold a signed numeric value from -128 through 127 .

scientific notation—a numeric expression format that includes an *E* (for exponent) and that specifies a number of implied decimal places.

scope—the area where a variable or constant is known and can be used.

sealed class—a class that cannot be extended.

searching an array—the process of comparing a value to a list of values in an array, looking for a match.

self-documenting—describes a program element that is self-explanatory.

sentinel value—a value that a user must supply to stop a loop.

sequence structure—a unit of program logic in which one step follows another unconditionally.

sequential access file—a data file in which each record is stored in order based on the value in some field.

sequential search—a search that is conducted by examining a list item-by-item in sequence.

serialization—the process of converting objects into streams of bytes.

set accessors—methods in properties that allow use of the assignment operator with a property name.

setter—another term for a class property's set accessor.

short—an integral data type that can hold a signed numeric value in two bytes.

short-circuit evaluation—the C# feature in which parts of an AND or OR expression are evaluated only as far as necessary to determine whether the entire expression is true or false.

side effect—an unintended consequence of an operation.

signature—a method's name and parameter list.

significant digits—specifies the mathematical accuracy of a value.

simple type—describes one of the following in C#: `byte`, `sbyte`, `short`, `ushort`, `int`, `uint`, `long`, `ulong`, `float`, `double`, `decimal`, `char`, and `bool`.

single-dimensional array—an array whose elements are accessed using a single subscript. Also see *one-dimensional array*.

snap lines—lines that appear in a design environment to help designers align new Controls with others that are already in place.

software—computer programs

source code—program statements.

standard numeric format strings—strings of characters expressed within double quotation marks that indicate a format for output.

state of an object—the collective value of all an object's attributes at any point in time.

static—a keyword that indicates that a method will be executed through a class and not by an object.

step value—the amount by which a loop control variable is altered on each iteration, especially in a `for` loop.

stream—a pipeline or channel through which bytes are input from and output to a file.

string—a data type that can hold a series of characters.

subclass—a derived class; a child class; a class that has inherited from a base class.

subscript—an integer contained within square brackets that indicates the position of one of an array's elements. Also see *index*.

suggestion mode—an IDE editing mode in which completion mode is turned off.

superclass—a base class; a parent class; a class that is the basis for inheritance.

switch structure—a structure that tests a single variable against a series of exact matches.

syntax—the set of grammar rules in a programming language.

syntax error—an error that occurs when a programming language is used incorrectly.

T

tab order—describes the sequence of controls selected when the user presses the Tab key in a program with a GUI interface.

ternary—describes an operator that requires three arguments.

TextBoxes—controls through which a user can enter input data in a GUI application.

text files—files that contain data that can be read in a text editor because the data has been encoded using a scheme such as ASCII or Unicode.

this reference—the reference to an object that is implicitly passed to an instance method of its class.

token—a block of text within a string that represents an entity or field.

transitive—inheriting all the members of one's ancestors.

triggers an event—causes an event to occur. Also see *fires an event* and *raises an event*.

truth tables—diagrams used in mathematics and logic to help describe the truth of an entire expression based on the truth of its parts.

try block—a block of code that might create exceptions that will be handled.

two-dimensional arrays—multidimensional arrays that have two or more columns of values for each row.

type precedence—a hierarchy of data types used to determine the unifying type in arithmetic expressions containing dissimilar data types.

U

uint—an integral data type that can hold an unsigned numeric value in four bytes.

ulong—an integral data type that can hold an unsigned numeric value in eight bytes.

unary operators—operators used with one operand.

Unicode—a 16-bit coding scheme for characters.

Unified Modeling Language (UML) diagrams—graphical tools that programmers and analysts use to describe systems.

unifying type—the type chosen for an arithmetic result when operands are of dissimilar types.

unreachable—describes program statements that can never execute under any circumstances because the program logic “can't get there.” Also see *dead code*.

upper camel casing—a style of creating identifiers in which the first letter of all new words in a name is capitalized, even the first one; Pascal casing.

ushort—an integral data type that can hold an unsigned numeric value in two bytes.

using clause, or using directive—code that declares a namespace.

V

value parameter—a parameter to a method that receives a copy of the value passed to it. Contrast with *reference parameter* and *output parameter*.

value types—data types that hold a value; they are predefined types such as `int`, `double`, and `char`. Contrast with *reference type*.

variable—a named location in computer memory that can hold different values at different points in time.

variable declaration—the statement that names a variable; it includes a data type, an identifier, an optional assignment operator and assigned value, and an ending semicolon.

verbatim identifier—an identifier with an `@` prefix.

virtual method—a method whose behavior is determined by the implementation in a child class.

visible—describes a class member that has not been hidden.

void—a keyword that indicates that a method does not return any value when called.

volatile—the type of computer storage that is lost when power is lost.

W

while loop—a structure that executes a body of statements continuously while some condition continues to be true; it uses the keyword `while`.

whitespace—any combination of spaces, tabs, and carriage returns (blank lines) in a program.

widgets—interactive controls such as labels, scroll bars, check boxes, and radio buttons.

write to a file—to store data in a computer file on a permanent storage device.

X

XML—an abbreviation of eXtensible Markup Language, which is a standard for exchanging data over the Internet.

XML-documentation format

comments—comments that use a special set of tags within angle brackets to create documentation within a program.