

**abstract** a modifier of a class or field which indicates that it is incomplete and must be fully defined in a derived class.

**abstract data type** a data type whose properties (domain and operations) are specified independently of any particular implementation

**abstraction** a model of a complex system that includes only the details essential to the perspective of the viewer of the system; the separation of the logical properties of data or actions from their implementation details

**abstract method** a method declared in a class or an interface without a method body

**acceptance tests** the process of testing the system in its real environment with real data

**activation record (stack frame)** a record used at run time to store information about a method call, including the parameters, local variables, register values, and return address

**actual parameter** a variable, constant, or expression listed in the call to a method

**adjacency list** a linked list that identifies all the vertices to which a particular vertex is connected; each vertex has its own adjacency list

**adjacency matrix** for a graph with N nodes, an NxN table that shows the existence (and weights) of all edges in the graph

**adjacent vertices** two vertices in a graph that are connected by an edge

**aggregate object** an object whose class definition includes variables that are themselves references to classes

**algorithm** a logical sequence of discrete steps that describes a complete solution to a given problem computable in a finite amount of time and space

**allocate** to assign memory space at run time for use by an object.

**assignment statement** a statement that stores the value of an expression into a variable

**asynchronous** not occurring at the same moment in time as some specific operation of the computer. In other words, not synchronized with the computer's actions.

**atomic data type** a data type whose elements are single, non-decomposable data items

**base case** the case for which the solution can be stated nonrecursively

**base class** the class being inherited from

**batch processing** a technique for entering data and executing programs without intermediate user interaction with the computer

**big-O notation** a notation that expresses computing time (complexity) as the term in a function that increases most rapidly relative to the size of a problem

**binary** expressed in terms of combinations of the numbers 1 and 0 only

**binary Operator** an operator that has two operands.

**binary search** a search algorithm for sorted lists that involves dividing the list in half and determining, by value comparison, whether the item would be in the upper or lower half; the process is performed repeatedly until either the item is found or it is determined that the item is not on the list

**binary search tree** a binary tree in which the key value in any node is greater than the key value in its left child and any of its children (the nodes in the left subtree) and less than the key value in its right child and any of its children (the nodes in the right subtree)

**binary tree** a tree in which each node is capable of having two child nodes, a left child node and a right child node

**bit** short for binary digit; a single 1 or 0

**black box testing** testing a program or method based on the possible input values, treating the code as a "black box"

**body** the statement(s) to be repeated within the loop; the executable statement(s) within a method

**Boolean** a data type consisting of only two values: true and false

**Boolean expression** an assertion that is evaluated as either true or false, the only values of the Boolean data type

**Boolean operators** operators applied to values of the type Boolean; in Java these are the special symbols &&, ||, and !

**booting the system** the process of starting up a computer by loading the operating system into its main memory

- branch** a code segment that is not always executed; for example, a switch statement has as many branches as there are case labels
- branching control structure** see *selection control structure*
- bucket** a collection of elements associated with a particular hash location
- button** a component of a frame that fires an event (called a *button event*) when the user clicks on it with the mouse.
- byte** eight bits
- bytecode** a standard machine language into which Java source code is compiled .
- catch an exception** code that is executed to handle a thrown exception is said to catch the exception
- central processing unit (CPU)** the part of the computer that executes the instructions (program) stored in memory; consists of the arithmetic/logic unit and the control unit
- chain** a linked list of elements that share the same hash location
- character set** a standard set of alphanumeric characters with a given collating sequence and binary representation
- circular linked list** a list in which every node has a successor; the "last" element is succeeded by the "first" element
- class (general sense)** a description of the behavior of a group of objects with similar properties and behaviors
- class (Java construct)** a pattern for an object
- class constructor** a special member method of a class that is implicitly invoked when a class object is defined
- class data** data that is associated with a class and accessible by all objects of that class.
- class member** a component of a class; class members may be either data or methods
- class method** a method that is associated with a class but not with a specific object. It is called by writing the name of the class followed by a period and then the name of the method and its parameter list.
- class object (class instance)** a variable of a class type
- clear (white) box testing** testing a program or method based on covering all of the branches or paths of the code
- client** software that declares and manipulates objects (instances) of a particular class
- clustering** the tendency of elements to become unevenly distributed in the hash table, with many elements clustering around a single hash location
- code** data type specifications and instructions for a computer that are written in a programming language.
- code coverage** see *clear (white) box testing*
- code walk-through** a verification process for a program in which each statement is examined to check that it faithfully implements the corresponding algorithmic step, and that the preconditions and postconditions of each method are preserved
- coding** translating an algorithm into a programming language; the process of assigning bit patterns to pieces of information
- collating sequence** the ordering of the elements of a set or series, such as the characters (values) in a character set
- collection** an object that holds other objects; typically we are interested in inserting, removing and iterating through the contents of a collection
- collision** the condition resulting when two or more keys produce the same hash location
- complete binary tree** a binary tree that is either full or full through the next-to-last level, with the leaves on the last level as far to the left as possible
- complete graph** a graph in which every vertex is directly connected to every other vertex
- complexity** a measure of the effort expended by the computer in performing a computation, relative to the size of the computation
- composite type** a data type whose elements are composed of multiple data items
- computer** a programmable device that can store, retrieve, and process data
- computer program** a list of instructions to be performed by a computer
- computer programming** the process of planning a sequence of steps for a computer to follow

- concrete class** a class that can have instantiated objects; it does not contain any abstract methods
- concrete step** a step for which the implementation details are fully specified
- constant** an item in a program whose value is fixed at compile time and cannot be changed during execution
- constant time** an algorithm whose Big-O work expression is a constant
- constructor (general)** an operation that builds new instances of an abstract data type (such as a list)
- constructor (Java)** an operation that creates a new instance of a class; a method that has the same name as the class type containing it, which is called whenever an object of that type is instantiated.
- control abstraction** the separation of the logical properties of a control structure from its implementation
- control structure** a statement used to alter the normally sequential flow of control
- control unit** the component of the central processing unit that controls the action of other components so that instructions (the program) are executed in sequence
- count-controlled loop** a loop that executes a predetermined number of times
- counter** a variable whose value is incremented to keep track of the number of times a process or event occurs
- CPU** see *central processing unit*
- crash** the cessation of a computer's operations as a result of the failure of one of its components; cessation of program execution to an error due to an error
- data** the representation of information in a manner suitable for communication or analysis by humans or machines
- data abstraction** the separation of a data type's logical properties from its implementation
- data coverage** see *black box testing*
- data encapsulation** the separation of the representation of data from the applications that use the data at a logical level; a programming language feature that enforces information hiding
- data flow** the flow of information from the calling code to a method and from the method back to the calling code
- data representation** the concrete form of data used to represent the abstract values of an abstract data type
- data structure** a collection of data elements whose logical organization reflects a relationship among the elements; a data structure is characterized by accessing operations that are used to store and retrieve the individual data elements; the implementation of the composite data members in an abstract data type
- data type** a category of data characterized by the supported elements of the category and the supported operations on those elements
- data validation** a test added to a program or a method that checks for errors in the data
- deallocate** to return the storage space for an object to the pool of free memory so that it can be reallocated to new objects
- debugging** the process by which errors are removed from a program
- decision** see *selection control structure*
- declaration** a statement that associates an identifier with a process or object so that the user can refer to that process or object by name
- deep copy** an operation that copies one class instance to another, using observer methods as necessary to eliminate nested references and copy only the primitive types that they refer to; the result is that the two instances do not contain and duplicate elements
- delimiter** a symbol or keyword that marks the beginning or end of a construct (e.g., statement, comment, declaration, or parameter list)
- deprecated** a Java construct is deprecated when the Java developers have decided that the construct might not be supported in future releases of the language; use of deprecated features is discouraged
- depth of recursion** the number of recursive calls used to complete an original call of a recursive method
- derived class** the class that inherits
- deskchecking** tracing an execution of a design or program on paper
- development environment** a single package containing all of the software required for developing a program

- dialog** a style of user interface in which the user enters data and then performs a separate action (such as clicking a button) when the entered values are ready to be processed by the program.
- direct execution** the process by which a computer performs the actions specified in a machine language program.
- directed graph (digraph)** a graph in which each edge is directed from one vertex to another (or the same) vertex
- direct recursion** recursion in which a method directly calls itself
- documentation** the written text and comments that make a program easier for others to understand, use, and modify
- doubly linked list** a linked list in which each node is linked to both its successor and its predecessor
- down** a descriptive term applied to a computer when it is not in a usable condition
- driver** a simple dummy main program that is used to call a method being tested; a main method in an object-oriented program
- dynamic allocation** allocation of memory space for a variable at run time (as opposed to static allocation at compile time)
- dynamic binding** determining at run time which form of a polymorphic method to call.
- dynamic data structure** a data structure that can expand and contract during program execution
- dynamic memory management** the allocation and deallocation of storage space as needed while an application is executing.
- echo printing** printing the data values input to a program to verify that they are correct
- edge (arc)** a pair of vertices representing a connection between two nodes in a graph
- editor** an interactive program used to create and modify source programs or data
- encapsulation** (in OOD) the bundling of data and actions in such a way that the logical properties of the data and actions are separated from the implementation details
- event** an action, such as a mouse click, that takes place asynchronously with respect to the execution of the program.
- event counter** a variable that is incremented each time a particular event occurs
- event handler** a method that is part of an event listener and is invoked when the listener receives a corresponding event.
- event handling** the process of responding to events that can occur at any time during execution of the program.
- event listener** an object that is waiting for one or more events to occur.
- event-controlled loop** a loop that terminates when something happens inside the loop body to signal that the loop should be exited
- exception** a situation associated with an unusual, often unpredictable event, detectable by software or hardware, which requires special processing
- exception handler** a section of a program that is executed when an exception occurs. In Java, an exception handler appears within a catch clause of a *try-catch-finally* control structure.
- executing** the action of a computer performing as instructed by a given program
- external file** a file that is used to communicate with people or programs and is stored externally to the program
- field** a component of a frame in which the user can type a value. The user must first place the cursor in the field by clicking inside the field; a named place in memory that holds a data or a reference to an object.
- file** a named area in secondary storage that is used to hold a collection of data; the collection of data itself
- firing an event** an event source generates an event.
- flag** a Boolean variable that is set in one part of the program and tested in another to control the logical flow of a program
- flow of control** the order of execution of the statements in a program
- folding** a hash method that breaks the key into several pieces and concatenates or exclusive-ORs some of them to form the hash value
- formal parameter** a variable declared in a method heading

- formatting** the planned positioning of statements or declarations and blanks on a line of a program; the arranging of program output so that it is neatly spaced and aligned
- free pool (heap)** an area of memory, managed by the JVM, that is used to provide storage space for objects.
- full binary tree** a binary tree in which all of the leaves are on the same level and every nonleaf node has two children
- functional decomposition** a technique for developing software in which the problem is divided into more easily handled subproblems, the solutions of which create a solution to the overall problem.
- functional domain** the set of valid input data for a program or method
- functional problem description** a description that clearly states what a program is to do
- garbage** the set of currently unreachable objects.
- garbage collection** the process of finding all unreachable objects and deallocating their storage space.
- general (recursive) case** the case for which the solution is expressed in terms of a smaller version of itself; also known as *recursive case*.
- generic data type** a type for which the operations are defined but the types of the items being manipulated are not
- graph** a data structure that consists of a set of nodes and a set of edges that relate the nodes to each other
- hardware** the physical components of a computer
- hash function** a function used to manipulate the key of an element in a list to identify its location in the list
- hashing** the technique used for ordering and accessing elements in a list in a relatively constant amount of time by manipulating the key to identify its location in the list
- hash table** the data structure used to store and retrieve elements using hashing
- header node** a placeholder node at the beginning of a list; used to simplify list processing
- heap** an implementation of a Priority Queue based on a complete binary tree, each of whose elements contains a value that is greater than or equal to the value of each of its children; see also *free pool*
- heuristics** assorted problem-solving strategies
- hide** to provide a field in a derived class that has the same name as a field in its superclass; to provide a class method that has the same form of heading as a class method in its superclass; the field or class method is said to hide the corresponding component of the superclass.
- hierarchy** (in OOD) structuring of abstractions in which a descendant object inherits the characteristics of its ancestors
- high-level programming language** any programming language in which a single statement translates into one or more machine language instructions
- homogeneous** a descriptive term applied to structures in which all components are of the same data type (such as an array)
- identifier** a name associated with a process or object and used to refer to that process or object
- immutable object** an object whose state cannot be changed once it is created.
- implementation phase** the second set of steps in programming a computer; translating (coding) the algorithm into a programming language; testing the resulting program by running it on a computer, checking for accuracy, and making any necessary corrections; using the program
- implementing** coding and testing an algorithm
- implementing a test plan** running the program with the test cases listed in the test plan
- in place** describes a kind of sorting algorithm in which the components in an array are sorted without the use of a second array
- index** a value that selects a component of an array
- indirect recursion** recursion in which a chain of two or more method calls returns to the method that originated the chain
- infinite loop** a loop whose termination condition is never reached and which therefore is never exited without intervention from outside of the program
- infinite recursion** the situation in which a method calls itself over and over continuously
- information** any knowledge that can be communicated

- information hiding** the practice of hiding the details of a module with the goal of controlling access to the details from the rest of the system
- inheritance** a design technique used with a hierarchy of classes by which each descendant class inherits the properties (data and operations) of its ancestor class; the language mechanism by which one class acquires the properties—data and operations—of another class; a mechanism for automatically sharing data and methods among members of a class and its subclasses
- inheritance of interfaces** a Java interface can extend another Java interface, inheriting its requirements; if interface B extends interface A, then classes that implement interface B must also implement interface A.
- inner class** a class defined as a member of another class; the outer class can access the private variables of the inner class
- inorder traversal** a systematic way of visiting all the nodes in a binary tree that visits the nodes in the left subtree of a node, then visits the node, and then visits the nodes in the right subtree of the node
- input** the process of placing values from an outside data set into variables in a program; the data may come from either an input device (keyboard) or an auxiliary storage device (disk or tape)
- input prompts** messages printed by an interactive program, explaining what data is to be entered
- input/output (I/O) devices** the parts of a computer that accept data to be processed (input) and present the results of that processing (output)
- insertion sort** a sorting algorithm in which values are placed one at a time into their proper position within a list that was originally empty
- inspection** a verification method in which one member of a team reads the program or design line by line and the others point out errors
- instance data** data that is associated with a specific object.
- instance method** A method that is associated with an object of a given type. It is called by writing the name of the object followed by a period and then the name of the method and its parameter list.
- instantiate** to create an object based on the description supplied by a class.
- instantiation** creating an object, an instance of a class
- integer number** a positive or negative whole number made up of a sign and digits (when the sign is omitted, a positive sign is assumed)
- integration testing** testing performed to integrate program modules that have already been independently unit tested
- interactive system** a system that allows direct communication between the user and the computer
- interface** a connecting link (such as a computer terminal) at a shared boundary that allows independent systems (such as the user and the computer) to meet and act on or communicate with each other; in Java, the formal definition of the behavior of a class and the mechanism for communicating with it
- interpretation** the translation, while a program is running, of non-machine-language instructions (such as Bytecode) into executable operations.
- interpreter** a program that inputs a program in a high-level language and directs the computer to perform the actions specified in each statement; unlike a compiler, an interpreter does not produce a machine language version of the entire program
- invoke** to call on a method, causing the method to execute before control is returned to the statement following the call
- iteration** an individual pass through, or repetition of, the body of a loop
- iteration counter** a counter variable that is incremented with each iteration of a loop
- iterator** an operation that allows us to process – one at a time – all the components in an object.
- key** the attributes that are used to determine the logical order of the items on a list
- layout manager** a method in the Frame class that automatically manages the placement of display elements within this particular style of window on the screen.
- leaf node** a tree node that has no children
- length** the number of items in a list; the length can vary over time
- linear probing** resolving a hash collision by sequentially searching a hash table beginning at the location returned by the hash function

- linear relationship** Each element except the first has a unique predecessor, and each element except the last has a unique successor
- linear time** for an algorithm, when the Big-O work expression can be expressed in terms of a constant times  $N$ , where  $N$  is the number of values in a data set
- linked list** a list in which the order of the components is determined by an explicit link field in each node, rather than by the sequential order of the components in memory
- listing** a copy of a source program, output by a compiler, containing messages to the programmer
- literal value** any constant value written in a program
- local data** data that is associated with a specific call to a method.
- local variable** a variable declared within a block; it is not accessible outside of that block
- logarithmic order** for an algorithm, when the Big-O work expression can be expressed in terms of the logarithm of  $N$ , where  $N$  is the number of values in a data set
- logging on** taking the preliminary steps necessary to identify yourself to a computer so that it accept your commands
- logging off** informing a computer—usually through a simple command—that no further commands follow
- logical order** the order in which the programmer wants the statements in the program to be executed, which may differ from the physical order in which they appear
- loop** a method of structuring statements so that they are repeated while certain conditions are met
- loop control variable (LCV)** a variable whose value is used to determine whether the loop executes another iteration or exits
- loop entry** the point at which the flow of control first passes to a statement inside a loop
- loop exit** that point when the repetition of the loop body ends and control passes to the first statement following the loop
- loop invariant** assertions about the characteristics of a loop that must always be true for a loop to execute properly; the assertions are true on loop entry, at the start of each loop iteration, and on exit from the loop, but are not necessarily true at each point in the body of the loop
- loop test** the point at which the loop expression is evaluated and the decision is made either to begin a new iteration or skip to the statement immediately following the loop
- machine language** the language, made up of binary-coded instructions, that is used directly by the computer
- mainframe** a large computing system designed for high-volume processing or for use by many people at once
- maintenance** the modification of a program, after it has been completed, in order to meet changing requirements or to take care of any errors that show up
- maintenance phase** period during which maintenance occurs
- member** a field (variable) or method declaration within a class.
- member selector** the expression used to access components of an object; it is formed by using the variable name and the member name, separated by a dot (period)
- memory unit** internal data storage in a computer
- metalanguage** a language that is used to write the syntax rules for another language
- metric based testing** testing based on measurable factors
- minicomputer** a computer system larger than a personal computer but smaller than a mainframe; sometimes called an entry-level mainframe
- modifiability** the property of an encapsulated class definition that allows the implementation to be changed without having an effect on code that uses it (except in terms of speed or memory space).
- module** a cohesive system subunit that performs a share of the work; in Java, the primary module mechanism is the *class*.
- multiple inheritance of interfaces** unlike for classes, Java does support multiple inheritance of interfaces; if interface  $C$  extends both interface  $A$  and interface  $B$ , then classes that implement interface  $C$  must also implement both interface  $A$  and interface  $B$
- mutable** an object whose state can be changed after it is created.
- named constant** a location in memory, referenced by an identifier, where a data value that cannot be changed is stored

- nested control structure** a program structure consisting of one control statement (selection, iteration, etc.) embedded within another control statement
- nested If** an If statement that is nested within another If statement
- nested loop** a loop that is within another loop
- new** a Java operator that returns the address of new space allocated for a dynamic variable
- nodes** the building blocks of dynamic structures, each made up of a component (the data) and a reference (the link) to the next node
- null statement** an empty statement
- object (general sense)** an entity or thing that is relevant in the context of a problem
- object (Java)** an instance of a class
- object class (class)** the description of a group of objects with similar properties and behaviors; a pattern for creating individual objects
- object Code** a machine language version of a source code.
- object-based programming language** a programming language that supports abstraction and encapsulation, but not inheritance
- object-oriented Design** a technique for developing software in which the solution is expressed in terms of objects—self-contained entities composed of data and operations on that data that interact by sending messages to one another.
- object-oriented programming** a method of implementation in which programs are organized as cooperative collections of objects, each of which represents an instance of some class, and whose classes are all members of a hierarchy of classes united via inheritance relationships
- observer** a method that returns an observation on the state of an object without changing it.
- one-dimensional array** a structured collection of components of the same type given a single name; each component is accessed by an index that indicates its position within the collection
- operating system** a set of programs that manages all of the computer's resources
- out-of-bounds array index** an index value that is either less than 0 or greater than the array size minus 1.
- overflow** the condition that arises when the value of a calculation is too large to be represented
- overloading** the repeated use of a method name with a different signature.
- override** to provide an instance method in a derived class that has the same form of heading as an instance method in its superclass. The method in the derived class redefines (overrides) the method in its superclass. We cannot override class methods.
- package** a set of related classes, grouped together to provide efficient access and use
- parameter** a literal, constant, variable, or expression used for communicating values to or from a method
- parameter list** a mechanism by which methods communicate with each other
- parameter passing** the transfer of data between the arguments and parameters in a method call.
- parameter** an expression used for communicating values to a method.
- path** a combination of branches that might be traversed when a program or method is executed; a sequence of vertices that connects two nodes in a graph
- path testing** a testing technique whereby the tester tries to execute all possible paths in a program or method
- peripheral device** an input, output, or auxiliary storage device attached to a computer
- polymorphic operation** an operation that has multiple meanings depending on the type of the object to which it is bound at run time
- polymorphism** the ability to determine which of several operations with the same name is appropriate; a combination of static and dynamic binding
- positional matching** a method of matching actual and formal parameters by their relative positions in the two parameter lists; also called *relative* or *implicit* matching
- postconditions** statements that describe what results are to be expected at the exit of an operation or method, assuming that the preconditions are true
- postfix operator** an operator that follows its operand(s)

- postorder traversal** a systematic way of visiting all the nodes in a binary tree that visits the nodes in the left subtree of a node, then visits the nodes in the right subtree of the node, and then visits the node
- precision** a maximum number of significant digits
- preconditions** assumptions that must be true on entry into an operation or method for the postconditions to be guaranteed
- prefix operator** an operator that precedes its operand(s)
- preorder traversal** a systematic way of visiting all the nodes in a binary tree that visits a node, then visits all the nodes in the left subtree of the node, and then visits the nodes in the right subtree of the node
- primitive type** a data type whose elements are single, non-decomposable data items
- problem-solving phase** the first set of steps in programming a computer: analyzing the problem; developing an algorithm; testing the algorithm for accuracy
- procedural abstraction** the separation of the logical properties of an action from its implementation
- program validation** the process of determining the degree to which software fulfills its intended purpose
- program verification** the process of determining the degree to which a software product fulfills its specifications
- programming** planning, scheduling, or performing a task or an event; see also *computer programming*
- programming language** a set of rules, symbols, and special words used to construct a computer program.
- pseudocode** a mixture of English statements and Java-like control structures that can easily be translated into a programming language
- public interface** the members of a class that can be accessed outside of the class, together with the modes of access that are specified by other modifiers.
- quadratic probing** resolving a hash collision by using the rehashing formula  $(\text{HashValue} + I^2) \% \text{array-size}$ , where  $I$  is the number of times that the rehash function has been applied
- queue** a structure in which elements are added to the rear and removed from the front; a "first in, first out" (FIFO) structure
- radix** the number of possibilities for each position; the digits in a number system
- random probing** resolving a hash collision by generating pseudorandom hash values in successive applications of the rehash function
- range of values** the interval within which values must fall, specified in terms of the largest and smallest allowable values
- real number** a number that has a whole and a fractional part and no imaginary part
- recursion** the situation in which a method calls itself
- recursive algorithm** a solution that is expressed in terms of (a) smaller instances of itself and (b) a base case.
- recursive call** a method call in which the method being called is the same as the one making the call
- recursive case** see *general case*
- recursive definition** a definition in which something is defined in terms of a smaller version of itself; a definition that is recursive
- reference type** a simple data type consisting of an unbounded set of values, each of which is the address of a variable of a given type. The only operation defined on a reference variable is initialization, after which every appearance of the variable is implicitly dereferenced
- refinement** in top-down design, the expansion of a module specification to form a new module that solves a major step in the computer solution of a problem
- registering the listener** adding the listener to an event source object's list of interested listeners.
- regression testing** re-execution of program tests after modifications have been made in order to ensure that the program still works correctly
- rehashing** resolving a collision by computing a new hash location from a hash function that manipulates the original location rather than the element's key
- relational operators** operators that state that a relationship exists between two values; in Java, symbols that cause the computer to perform operations to verify whether or not the indicated relationship exists

**relative matching** see *positional matching*

**representational error** arithmetic error caused when the precision of the true result of arithmetic operations is greater than the precision of the machine

**requirements** a statement of what is to be provided by a computer system or software product

**reserved word** a word that has special meaning in a programming language; it cannot be used as an identifier

**return** the point at which the computer comes back from executing a method

**reuse** the ability to import a class into any program without additional modification to either the class or the program; the ability to extend the definition of a class.

**right-justified** placed as far to the right as possible within a fixed number of character positions

**robust** a descriptive term for a program that can recover from erroneous inputs and keep running

**robustness** the ability of a program to recover following an error; the ability of a program to continue to operate within its environment

**root** the top node of a tree structure; a node with no parent

**run-time stack** a data structure that keeps track of activation records during the execution of a program

**scope** the region of program code where it is legal to reference (use) an identifier

**scope rules** the rules that determine where in a program a given identifier may be accessed, given the point at which the identifier is declared

**secondary storage device** see *auxiliary storage device*

**selection control structure** a form of program structure allowing the computer to select one among possible actions to perform based on given circumstances; also called a *branching control structure*

**self-documenting code** a program containing meaningful identifiers as well as judiciously used clarifying comments

**self-referential class** a class that includes an instance variable or variables that can hold a reference to an object of the same class

**semantics** the set of rules that gives the meaning of instruction written in a programming language

**sentinel** a special data value used in certain event-controlled loops as a signal that the loop should be exited

**sequence** a structure in which statements are executed one after another

**shallow copy** an operation that copies a source class instance to a destination class instance, simply copying all references so that the destination instance contains duplicate references to values that are also referred to by the source

**short-circuit (conditional) evaluation** evaluation of a logical expression in left-to-right order with evaluation stopping as soon as the final Boolean value can be determined.

**side effect** any effect of one method on another that is not part of the explicitly defined interface between them

**signature** the distinguishing features of a method heading. The combination of a method name with the number and type(s) of its parameters in their given order.

**significant digits** those digits from the first nonzero digit on the left to the last nonzero digit on the right (plus any zero digits that are exact)

**simulation** a problem solution that has been arrived at through the application of an algorithm designed to model the behavior of physical systems, materials, or processes

**size (of an array)** the physical space reserved for an array

**software** computer programs; the set of all programs available on a computer

**software engineering** the discipline devoted to the design, production and maintenance of computer programs that are developed on time and within cost estimates, using tools that help to manage the size and complexity of the resulting software products

**software life cycle** the phases in the life of a large software project including requirements analysis, specification, design, implementation, testing, and maintenance

**software piracy** the unauthorized copying of software for either personal use or use by others

**software process** a standard, integrated set of software engineering tools and techniques used on a project or by an organization

- software specification** a detailed description of the function, inputs, processing, outputs, and special requirements of a software product; it provides the information needed to design and implement the system
- sorted list** a list that is sorted by the value in the key; there is a semantic relationship among the keys of the items in the list
- sorting** arranging the components of a list in order (for instance, words in alphabetical order, numbers in ascending or descending order)
- source code** data type specifications and instructions written in a high-level programming language.
- stable sort** a sorting algorithm that preserves the order of duplicates
- stack** a structure in which elements are added and removed from only one end; a "last in, first out" (LIFO) structure
- stack frame** see *activation record*
- stack overflow** the condition resulting from trying to push an element onto a full stack
- stack underflow** the condition resulting from trying to pop an empty stack
- standard (built-in) type** a data type that is automatically available for use in every Java program.
- standardized** made uniform; most high-level languages are standardized, as official descriptions of them exist
- state** the current values contained within an object.
- static binding** determining at compile time which form of a polymorphic method to call.
- stepwise design** see *top-down design*
- stepwise refinement** see *top-down design*
- structured data type** an organized collection of components; the organization determines the method used to access individual components.
- stub** a dummy method that assists in testing part of a program; it has the same method that would actually be called by the part of the program being tested, but is usually much simpler
- style** the individual manner in which computer programmers translate algorithms into a programming language
- subinterface** an interface that extends another interface; it must list all the abstract methods in the interface it extends, plus it can add more abstract methods of its own
- supercomputer** the most powerful class of computers
- switch expression** the expression in a Switch statement whose value determines which case label is selected; it cannot be a floating point expression
- syntax** the formal rules governing how valid instructions (constructs) are written in a programming language
- system software** a set of programs—including the compiler, the operating system, and the editor—that improves the efficiency and convenience of the computer's processing
- tail recursion** the case in which a method contains only a single recursive invocation and it is the last statement to be executed in the method
- team programming** the use of two or more programmers to design a program that would take one programmer too long to complete
- termination condition** the condition that causes a loop to be exited
- test driver** see *driver*
- test harness** a standalone program designed to facilitate testing of the implementations of algorithms
- test plan** a document showing the test cases planned for a program or module, their purposes, inputs, expected outputs, and criteria for success
- test plan implementation** using the test cases specified in a test plan to verify that a program outputs the predicted results
- testing** the process of executing a program with data sets designed to discover errors
- text file** a file in which each component is a character; each numeric digit is represented by its code in the collating sequence
- throw** interrupt the normal processing of a program to raise an exception that must be handled or rethrown by the surrounding block of code

- top-down design** a technique for developing a program in which the problem is divided into more easily handled subproblems, the solutions of which create a solution to the overall problem; also called stepwise refinement and modular programming
- trailer node** a placeholder node at the end of a list; used to simplify list processing
- transformer** a method that changes the internal state of an object.
- traverse a list** to access the components of a list one at a time from the beginning of the list to the end
- tree** a structure with a unique starting node (the root), in which each node is capable of having many child nodes, and in which a unique path exists from the root to every other node
- two-dimensional array** a collection of components, all of the same type, structured in two dimensions; each component is accessed by a pair of indices that represent the component's position within each dimension
- unary operator** an operator that has just one operand
- unchecked exception** a Java exception of the RuntimeException class, it does not have to be explicitly handled by the method within which it might be raised
- underflow** the condition that arises when the value of a calculation is too small to be represented
- undirected graph** a graph in which the edges have no direction
- unit testing** testing a class or method by itself
- unreachable** A condition of an object wherein there is no way to refer to it.
- unsorted list** A list in which data items are placed in no particular order with respect to their content; the only relationship between data elements is the list predecessor and successor relationships
- unstructured data type** a collection consisting of components that are not organized with respect to one another
- user name** the name by which a computer recognizes the user, and which must be entered to log on to a machine
- value parameter** a formal parameter that receives a copy of the contents of the corresponding actual parameter
- variable** a location in memory, referenced by an identifier, in which a data value that can be changed is stored
- vertex** a node in a graph
- visible** accessible; a term used in describing a scope of access
- void method** a method that is called as a separate statement. The method does not return a value.
- walk-through** a verification method in which a *team* performs a manual simulation of the program or design
- weighted graph** a graph in which each edge carries a value
- work** a measure of the effort expended by the computer in performing a computation
- workstation** a minicomputer or powerful microcomputer designed to be used primarily by one person at a time
- wrapper class** a Java class that wraps a primitive type, letting it be manipulated as an object, and providing some useful utility methods related to the type