

References

A

- **Algorithm**
An unambiguous, executable, and terminating specification of a way to solve a problem.
- **Accessor method**
A method that accesses an object but does not change it.
- **API documentation**
Information about each class in the Java library.
- **Applets**
A graphical Java program that executes inside a web browser or applet viewer.
- **Access specifier**
A keyword that indicates the accessibility of a feature, such as `private` or `public`.

B

- **Block**
A group of statements bracketed by `{}`.
- **Boolean type**
A type with two possible values: `true` and `false`.
- **Boolean operator**
An operator that can be applied to Boolean values. Java has three logical operators: `&&`, `||`, and `!`.
- **Black-box testing**
Testing a method without knowing its implementation.

- **Breakpoint**

A program that lets a user run another program one or a few steps at a time, stop execution, and inspect the variables in order to analyze it for bugs.

C

- **Cast**

Explicitly converting a value from one type to a different type. For example, the cast from a floating-point number x to an integer is expressed in Java by the cast notation `(int) x`.

- **Central processing unit (CPU)**

The part of a computer that executes the machine instructions.

- **Compiler**

A program that translates code in a high-level language (such as Java) to machine instructions (such as byte code for the Java virtual machine).

- **Class**

A programmer-defined data type.

- **Comment**

An explanation to help the human reader understand a section of a program; ignored by the compiler.

- **Component**

A building block for a graphical user interface, such as a button or a text field. User-interface components are used to present information to the user and allow the user to enter information to the program.

- **Case sensitive**

Distinguishing upper- and lowercase characters.

- **Compile-time error**

An error that is detected when a program is compiled.

- **Construction**

Setting a newly allocated object to an initial state.

- **Constructors**

A method that initializes a newly instantiated object.

- **Constants**

A value that cannot be changed by a program. In Java, constants are defined with the reserved word `final`.

- **Concatenation**

Placing one string after another to form a new string.

- **Code coverage**

A measure of the amount of source code that has been executed during testing.

D

- **Directories**

A structure on a disk that can hold files or other directories; also called a folder.

- **Debuggers**

A program that lets a user run another program one or a few steps at a time, stop execution, and inspect the variables in order to analyze it for bugs.

- **Documentation comments**

A comment in a source file that can be automatically extracted into the program documentation by a program such as `javadoc`.

E

- **Explicit parameter**

A parameter of a method other than the object on which the method is invoked.

- **Encapsulation**

The hiding of implementation details.

- **Enumeration types**

A type with a finite number of values, each of which has its own symbolic name.

F

- **Floating-point numbers**

A number that can have a fractional part.

- **Frame**

A window with a border and a title bar.

G

- **Garbage collector**

Automatic reclamation of memory occupied by objects that are no longer referenced.

H

I

- **Integers**

A number that cannot have a fractional part.

- **Implicit parameter**

The object on which a method is invoked. For example, in the call $x.f(y)$, the object x is the implicit parameter of the method f .

- **Inner class**

A class that is defined inside another class.

- **Instance variables**

A variable defined in a class for which every object of the class has its own value.

- **Initialization**

Setting a variable to a well-defined value when it is created.

J

K

L

- **Library**

A set of precompiled classes that can be included in programs.

- **Local variable**

A variable whose scope is a block.

- **Loop invariant**

A statement about the program state that is preserved when the statements in the loop are executed once.

M

- **Method**

A sequence of statements that has a name, may have formal parameters, and may return a value. A method can be invoked any number of times, with different values for its parameters.

- **Mutator method**

A method that changes the state of an object.

N

- **Number literal**

A constant value in a program this is explicitly written as a number, such as -2 or 6.02214115E23.

- **Nested**

A loop that is contained in another loop.

O

- **Object**

A value of a class type.

- **Overloaded**

Giving more than one meaning to a method name.

- **Object reference**

A value that denotes the location of an object in memory. In Java, a variable whose type is a class contains a reference to an object of that class.

P

- **Parameter**

An item of information that is specified to a method when the method is called. For example, in the call `System.out.println("Hello, World!")`, the parameters are the implicit parameter `System.out` and the explicit parameter `"Hello, World!"`.

- **Pseudocode**

A high-level description of the actions of a program or algorithm, using a mixture of English and informal programming language syntax.

- **Primitive types**

In Java, a number type or `boolean`.

- **Packages**

A collection of related classes. The `import` statement is used to access one or more classes in a package.

- **Parameter variables**

A variable of a method that is initialized with a parameter value when the method is called.

- **Public interface**

The features (methods, fields, and nested types) of a class that are accessible to all clients.

- **Prompt**

A string that tells the user to provide input.

- **Predicate method**

A method that returns a Boolean value.

- **Pseudorandom numbers**

A number that appears to be random but is generated by a mathematical formula.

Q

R

- **Run-time error**

An error in a syntactically correct program that causes it to act differently from its specification.

- **Reserved words**

A word that has a special meaning in a programming language and therefore cannot be used as a name by the programmer.

- **Relational operator**

An operator that compares two values, yielding a Boolean result.

S

- **Statements**

A syntactical unit in a program. In Java a statement is either a simple statement, a compound statement, or a block.

- **String**

A sequence of characters.

- **Source code**

Instructions in a programming language that need to be translated before execution on a computer.

- **Side effect**

An effect of a method other than returning a value.

- **Symmetric**

Bounds that include the starting index and the ending index.

- **Sentinel**

A value in input that is not to be used as an actual input value but to signal the end of input.

T

- **Type**

A named set of values and the operations that can be carried out with them.

U

- **Unit testing**

A test of a method by itself, isolated from the remainder of the program.

V

- **Variables**

A symbol in a program that identifies a storage location that can hold different values.

W

- **White space**

Any sequence of only space, tab, and newline characters.

- **White-box testing**

Testing methods by taking their implementations into account, in contrast to black-box testing; for

example, by selecting boundary test cases and ensuring that all branches of the code are covered by some test case.

X

Y

Z