

USING THE GAME

GLOSSARY OF PROGRAMMING TERMS

Abstract class A class that includes the keyword `abstract` in its signature and cannot be instantiated; it is used during the design process to collect data members and methods common to several classes

Aggregated class A class that contains at least one data member that references an object

Aggregation The concept of referencing objects from a class's data members

Algorithm A step-by-step solution to solving a problem or performing a task that a computer system can execute

Applet A Java program that runs from within another program, which is typically a Web browser; it has restrictions placed on its instruction set consistent with this execution mode's need for enforced security

Applet container program The program, typically a Web browser, within which an applet runs; the container invokes the methods that are part of the applet's lifecycle

Applet lifecycle The period of time that begins when an applet's execution is initiated and ends when it is terminated; during this time period, the applet's container program invokes the applet methods `init`, `start`, `paint`, `stop`, and `destroy` to manage its execution

Application Programming Interface (API) A collection of packages containing interfaces and implementations of classes and data structures that can easily be incorporated into a Java program

Application software All non-operating system software, typically for use by human users

Argument A value passed to a method when it is invoked

Argument list A sequence of argument names separated by commas enclosed in a set of parentheses

Array An ordered collection of primitive or reference variables stored inside an object, which are sequentially associated with an integer beginning with zero; arrays are an implementation of the mathematical concept of subscripted variables

Array of objects An array of reference variables that contains the addresses of a set of instances of the same class

ASCII Table A specific tabulation of characters and control characters and the bit patterns used to represent them

Assignment The act of changing the contents of a variable

Atomic components Graphical user interface (GUI) components that cannot contain other components, such as text fields and buttons; most of the program user's interactions are with these components

Autoboxing A context-sensitive feature of Java in which primitive literals or variables are replaced with instances of wrapper classes that contain their values

Base case Part of the methodology of formulating recursive algorithms, which is a known or trivial solution to the problem

Base class A class that is inherited from, also known as a parent or super class

Binary numbers A number system based on two, as opposed to the decimal system, which is based on ten

Bit A single unit of storage that can assume two states, which are referred to as off and on, or zero and one, or false and true

Boolean expression An expression involving relational and logic operators that evaluates to true or false

Buffer Memory used to temporarily store data during program execution

Byte A set of eight contiguous (adjacent) bits, often used to represent a single character in the Modern Latin (English) alphabet

Byte codes The translation of a program produced by the Java language translator into intermediate code

Central processing unit (CPU) Electronic circuitry that interprets and executes instructions; the CPU can perform arithmetic and logic operations, has the ability to skip or re-execute instructions based on the truth value of a logic operation, and contains a limited amount of storage called registers

Chain inheritance When the parent class of a class extends another class

Child class A class that inherits from (extends) another class; also known as a sub- or derived class

Class A collection of variables and methods; a blueprint for an object

Class-level variable A variable defined within a class but outside of a method's code block

Cloning an object Creating a new instance of a class and (deep) copying the values of all of the data members of an existing instance of the class into the new instance

Code block A set of instructions enclosed with a set of open and close braces, { }

Collection A data structure that is accessed without specifying a key

Collections Framework of the API Part of the API that contains generically implemented data structures, methods that perform common operations on data elements, and a set of associated interfaces

Computer system A set of electronic circuits, mechanical devices and enclosures, and instructions that these devices execute to perform a task

Concatenation The act of appending one string to another

Concurrency Executing several programs, or several parts of a program, at the same time

Constructor A method in a class that is used to create an instance of a class and return its address; its name is the same as the class's name

Consumer A process that expends data

Content pane The portion of a window or other top-level container that holds the visible components added to the container

Control of flow statement A statement that overrides the default sequential execution path of a program, such as a decision statement, a repetition (loop) statement, or a subprogram invocation

Counting algorithm An algorithm that counts by adding an increment to, or subtracting a counting increment from, the current value of a counter

Data members The variables defined within a class

Data structure An organization of data within memory to facilitate its processing from a speed and memory requirements viewpoint

Deep comparison of two objects Comparing the values of one or more of the data members of an object to the corresponding data members of another instance of the class

- Deep copy of an object** Copying the values of one or more of the data members of an instance of a class into the corresponding data members of another instance of the class
- Derived class** A class that inherits from (extends) another class; also known as a child or subclass.
- Deserializing objects** The act of reassembling objects after they are read from a disk file
- Dialog box** A predefined pop-up graphical interface used to pause a program's execution until the program user acknowledges a message or performs an input
- Divide and conquer** Expressing or defining a complicated entity as a set of less complicated entities; for example, expressing the solution to a complex problem as the solutions to a set of simpler problems, or defining the data members of a complex class to be instances of less complicated classes
- Dynamic binding** Delaying the process of locating an invoked method until runtime
- Dynamic programming** A programming technique aimed at reducing execution time, which avoids repetitive processing by saving and then reusing prior processing results
- Element** One of the variables contained in an array or one object contained in a data structure
- Enumerated type** A user defined type created within a Java program by specifying its type name and allowable values within an `enum` statement
- Event** An asynchronous occurrence during a program's execution that can be used to redirect the execution path of a program
- Event handler** A method that is executed when an event occurs
- Exception class** The API class `Throwable` or a descendent of that class
- Exception error message** A string contained within an exception object that normally contains descriptive error information
- Exception object** An instance of the API class `Throwable`, or one of its decedents, which can be passed to a `catch` clause when an error is detected during the execution of a method
- Exceptions** A programming construct that promotes the reusability of methods by deferring the decision as to what action to take when an error condition is detected to the invoker of the method
- Final class** A class that cannot be a parent class; it cannot be extended
- Flow chart** A graphical representation of an algorithm
- Fractal** A mathematical or geometric object that has the property of self-similarity; that is, each part of the object is a smaller or reduced copy of itself
- General solution** Part of the methodology of formulating recursive algorithms; it is a solution to the original problem that uses the portion of the methodology known as the reduced problem
- Generic class** A class that contains generic methods and is coded in a way as to permit the type of its data members to be specified when an instance of the class is created
- Generic method** A method that can perform its algorithm on any type of object passed to it
- Generic parameter** A parameter that can be passed an object of any type and whose type is specified using a type placeholder
- Generic parameter list** A list of the type placeholders, coded within a method's signature, that are used in the method's parameter list
- Generics** A programming concept that promotes reusability by permitting the type of a method's parameters and returned value to be specified by the method's invoker and permitting the type of a class's data members to be specified when an instance of the class is created

Get method A method used to fetch the values of a class's private data members

Graphical User Interface (GUI) A means of interacting with the program user via a point-and-click mode, as opposed to a text-based mode, aimed at facilitating the I/O process

Hypertext Markup Language (HTML) A scripting language for writing instructions to be downloaded and executed by a Web browser to build and display a Web page; the script can contain instructions to download and execute a Java applet

Index The integer associated with a variable in an array

Inheritance A programming concept in which a new class can contain all of the data members and methods of an existing class by simply including an `extends` clause in its heading

Inner class A class that is defined within another class

Input method A method normally named `input` that ordinarily permits the program user to input the values of all of an object's data members

Instance of a class A specific object in the class

Integrated Development Environment (IDE) A program used by a programmer to develop a software product; it contains a collection of tools (e.g., a syntax checker, translator, editor, file-management system) that facilitate the development process

Interface A Java construct used to specify the signatures of related methods that are implicitly abstract and/or a declaration of public constants that are implicitly static and final

Iterator An object that can be used to perform time-efficient processing on all of the data elements contained in any data structure that imposes an ordering on its data elements

Java Development Kit (JDK) A set of tools used to develop Java programs; these tools include the API classes, a debugger, a compiler, an interpreter, an applet viewer, a documentation generator, a disassembler, various linking, loading, and binding tools, and a runtime environment

Java Virtual Machine A virtual computer system whose programming language is Java byte codes

Key A value associated with a data element that can be used to refer to the element

Layout manager A predefined protocol for the sizing and positioning of components added to a GUI container

Listener list An association of events and their event-handler methods that is part of a GUI component object

Local variable A variable defined within a code block whose scope is limited to the instructions within the code block

Loop A sequence of instruction that is repeated a specified number of times or until a Boolean value becomes true or false

Map A set of data structures that associate a key with each data element stored in the structure; the key can be used to specify the data element on which to operate

Menu mnemonic A menu shortcut key (hot key) associated with a terminal menu item

Methods The subprograms defined within a class, a sequence of instructions that perform a particular task

Multidimensional array An array in which each variable of the array is associated with 2, 3, ... indices, for example an array of rows and columns

Multiple inheritance When a class inherits from more than one class; this is not supported in Java

Multitasking Executing several threads of an application at the same time or giving the impression that they are executing at the same time

Nested loops Coding loops inside of loops

Nested statements Statements that are contained within another statement or another statement's statement block

Non-void method A method that returns a value, whose type is specified in the method's signature

Object A particular occurrence of a class that contains all of the class's non-static data members

Object oriented programming (OOP) An approach to programming (a programming paradigm) aimed at facilitating the development of programs that deal with objects, such as starships, people, or Web pages

One-dimensional array An array in which each variable of the array is associated with one index

Operating system software A program to manage the resources of a computer system and to permit a user of the system to interact with it, usually via a point-and-click interface

Overloading methods The act of writing two or more methods in the same class that have the same name but different parameter lists

Overriding a method Rewriting an inherited method using the exact same signature of the inherited method

Parallel arrays A use of multiple one-dimensional arrays in which the *i*th element of each array is associated with the same entity; for example, if Mary's age was stored in the second element of one array, then the rest of Mary's information would be stored in the second element of the other arrays

Parameter A variable that can receive a value (an argument) passed to the method when it is invoked

Parameter list A sequence of parameter names, each preceded by its type, separated by commas, and enclosed in a set of parentheses

Parent class A class that is inherited from, also known as a super or base class

Parsing The act of changing a string into a numeric; also the act of separating a string into its component parts that are separated by a specified delimiter

Platform A particular CPU model and operating system

Platform independence The concept that the programmer's translation of a program can be transmitted to, and then run on, any computer system

Polymorphism The ability of one invocation to morph itself into an invocation of a parent's version of a method or any of its children's versions of the method; rooted in the fact that a parent reference variable can refer to an instance of a child class

Pop-up menu A space-saving alternative to a menu-bar-based drop-down menu that remains invisible until the user performs a platform-dependent mouse or keyboard action on a GUI component

Precedence rules A specification of the order in which to perform a set of operations

Primitive variable A variable that can store a numeric value, a Boolean value, or one character; the type used in its declaration is one of the primitive types

Primitive type The Java types `byte`, `short`, `int`, `long`, `float`, `double`, `char`, and `boolean`

Priority queue A queue that associates a priority with each of its data elements; the elements assigned the highest priority are fetched and deleted (on a first-in-first-out basis) before those of lower priority

Private data member A data member of a class that cannot be directly accessed by methods that are not part of the class; `get` and `set` methods are used to fetch and change their values

Producer A process that generates data

Pseudorandom numbers Apparent, but not truly, random numbers

Public data members Data members of a class that can be directly accessed by methods that are not part of the class; they are accessed by coding their name preceded by either the name of an instance of the class or the class name, followed by a dot

Queue A data structure in which the data elements are fetched and deleted on a first-in-first-out basis

Random access memory (RAM) High-speed, high-cost storage physically located in close proximity to the central processing unit

Recursion The act of defining something in terms of itself

Recursive method A method that invokes itself or initiates a sequence of method invocations that eventually leads to an invocation of itself

Reduced problem Part of the methodology of formulating recursive algorithms, it is a problem similar to the original problem, usually between the original problem and the base case, usually closer to the original problem, and (when progressively reduced) becomes the base case for all versions of the original problem

Reference variable A variable that can store a memory address; the type used in its declaration is the name of a class

Registering an event handler The act of associating an event-handler method with a particular event that could be performed on a GUI component

Runtime the time during which the program is in execution

Scope of a variable or a method The range of a program's instructions within which a variable can be used or a method can be invoked

Sentinel loop A loop that ends on a particular value of the data it is processing or on a particular user input; for example, a negative deposit

Serializing objects The act of disassembling objects before writing them to a disk file so they can be recreated when they are read from the disk

Set methods Methods used to change the values of a class's private data members

Shallow comparison Comparing the contents of one variable to the contents of another variable using the equality (`==`) operator

Shallow copy Copying the contents of one variable into another using the assignment (`=`) operator

Shared buffer Memory used to temporarily share a data item among one or more threads

Show method A method named `show` that ordinarily outputs all of the data members of an object or draws the object

Signature of a method The first line of a method's code

Software engineer A computer professional that produces programs that are error free, within budget, on schedule, and satisfy the customers' current and future needs

Stack A data structure in which the data elements are fetched and deleted on a last-in-first-out basis

States of a thread The six statuses a thread can assume from the time it is created to the time it is terminated

Static data member A class's data member that is designated to be shared by all instances of the class by including the keyword `static` in its declaration

Static method A method that is designated to be invoked by preceding the method name by the method's class name followed by a dot; they are intended to be methods that do not operate on instances of the class

String A finite sequence of characters

Subclass A class that inherits from (extends) another class, also known as a derived or child class

Super class A class that is inherited from, also known as a parent or base class

Swapping algorithm An algorithm that swaps the values contained in two variables

Synchronized buffer A buffer whose access is managed in a way that imposes protocols of proper access to the data on the threads that share the buffer

Syntax The rules for forming properly constructed program instructions; the grammar of a programming language

Text file A file whose information is intended to be characters and is therefore interpreted using the ASCII or Unicode tables; ordinarily the file extension `.txt` is appended to the file's name

Thread An independent execution path through a program

Token A component part of a string that is terminated by a specified delimiter, for example, a space

Tokenizing a string Extracting all of the tokens from a string

Top-level container The basic building block component of a graphical interface, which contains the other GUI components that make up the interface

toString method A method named `toString` whose task is to return the string representation of an object; ordinarily, the string contains the annotated values of all of an object's data members

Totaling or summation algorithm An algorithm that computes the sum of a set of numeric values by repeatedly adding each value to the subtotal of the values in the set that preceded it

Type placeholder Any valid identifier that is not the name of a class used within the application of which it is a part; a placeholder is used as a type of a generic parameter and can be used as a returned type

Unboxing A context-sensitive feature of Java in which an instance of a wrapper class object is replaced with the primitive value it contains

Unicode An expanded tabulation of characters and control characters and the bit patterns used to represent them

Universal modeling language (UML) diagram A graphical representation of a class that specifies the class's name, data members, and the signatures of its methods; it is used to design a class

Variable A named memory cell that can store a specific type of data item

Void method A method that does not return a value

Worker method A method that is invoked by another method to perform a specific task (work) for it; for example, fetching the value of one of an object's data members or drawing the object

Wrapper class An API class that contains non-static primitive data members of a particular type

