The Open University

Undergraduate Computing: Level 2
M257 Putting Java to work

M257 Glossary

WEB 86859 3
1.1

absolute positioning A style of creating graphical user interfaces, in which the programmer specifies the absolute coordinates and sizes for visual components on the screen, rather than the programmer relying on a layout manager to place such components in a particular layout style.

abstract The Java keyword to mark a class or method as incomplete.

abstract class A class that cannot be instantiated and serves as a base for development of other classes by inheritance.

abstract method A method that has a header but no body. Such methods normally occur either in an abstract class or in an interface.

access modifier Any of the Java keywords that allow a programmer to specify an access level for a class, constructor or class member; namely `protected`, `public`, or `private`.

access to shared resources The problem of several processes having access to the same shared resource, which may lead to inconsistencies if no proper care is taken of how to handle mutual exclusion. That is, we need to ensure that only one process at a time can access a shared resource.

accessible member A member available for access using either its simple name within a subclass, or a qualified name using an object reference.

accessor method A method which, when invoked on an object, returns a value representing part of the object's state. Also known as a *getter method*.

actual argument A value passed to a method and assigned to its corresponding formal argument.

adapter A support class that provides empty implementations for all of the methods of a listener interface. By extending such an adapter class (rather than implementing an interface), only the relevant methods need to be overridden. This is because all the other interface methods will already have been implemented in the adapter class (albeit with empty code bodies). An example is the `MouseAdapter` class used for handling mouse events.

addressing Identifying a particular computer on the internet, using either a symbolic address or a numeric address.

aggregation See *composition*.

aglet An autonomous mobile agent that is able to move from one computer to another computer carrying along with itself state information and executable code. The name is taken from ‘agile applet’.

algorithm A series of steps for solving a problem, such as sorting data into a specific order.

anchor An instruction which specifies where text will be positioned relative to a point in some graphic.

animation The production of images such that they give the illusion of motion.

animation frame One of a sequence of images making up an animation.

API Application Programming Interface – the collection of class libraries or standard packages available for use in Java programs. An individual package may also be referred to as an API, for example, the networking API, `java.net`.

applet Java application attached to a web page, downloaded as bytecode. The applet is executed on the user’s computer by a Java-enabled web browser when it displays the document to the user.

application A Java program that runs independently (as opposed to an applet).

application protocol A protocol that defines communication at the level of complete applications, like file transfer (FTP) or web page access (HTTP); normally contrasted with system protocols.

argument A value passed to a method when the method is invoked. Same as *actual argument*.

array A data structure consisting of one or more locations capable of storing items of the same type, indexed by an integer.

array initializer A Java expression used to create an array object initialized to contain some values, e.g. `{1, 2, 3}` is an array initializer for an `int[]` type array. This kind of expression is not a literal and cannot be used in every circumstance that an array reference can be used.

`ArrayList` **capacity** The maximum number of object references that can be stored in an `ArrayList` object, without it being extended. Since `ArrayList` objects automatically extend as required, this concept is primarily used in terms of setting the initial capacity when the object is created.

`ArrayList` **size** The current number of object references stored in an `ArrayList` object.

`ArrayList` A collection class encapsulating an array data structure and implementing an ordered collection of objects (supporting the `List` interface). An `ArrayList` object automatically extends to accommodate more references as required and hence is often preferable to standard arrays.

assignment Assignment is the process of copying a value into a variable. When using object references, a reference is copied, otherwise a primitive value is copied. Indicated in Java by the symbol '`=`'.

autoboxing The conversion of a primitive such as an `int` to a representative object such as an instance of `Integer`, without the intervention of the programmer.

bag A data structure that keeps track of the number of times items of a certain kind have been added to it.

binary input/output Input or output where data is in the same binary format that is used to store data internally in computer memory. (See *text input/output*.)

block comment A comment marked in Java by a pair of /* and */ characters and usually extending over several lines.

blocked state A process is said to be blocked when it is waiting for something to happen before it can continue and move to the runnable state. A process may be blocked because it has executed the `wait` or `sleep` command, or because it is waiting for IO processing to complete.

blocking call A method call is a blocking call if a statement following it is not executed until the method call has been completed. Compare with *non-blocking call*.

body The part of the `if`, `while`, `for` and `switch` statements that is enclosed within curly brackets.

`boolean` The primitive logical data type in Java.

`break` The Java keyword indicating that the current statement should be terminated and control passed to the next statement. This typically means that control jumps to the statement following the body in which the break statement is situated.

buffer An area of temporary storage to allow a reader and writer to operate at different rates during a data transfer, or to allow 'chunking' of streamed data into larger sizes.

buffered input/output Input or output using temporary memory storage (a buffer) to make the data transfer more efficient.

button A visual component that triggers an event when clicked.

`byte` The smallest of the integral types in Java. A variable of this type occupies 8 bits and is capable of storing a number between –127 and +128.

bytecode The result of compiling Java source code. Bytecode is portable in that it is the same for all platforms. Bytecode is interpreted appropriately on each platform when the Java application or applet is run.

byte stream See *stream.*

`case` The Java keyword used to begin a section of code in a `switch` statement.

casting The process of converting one type to another by using a cast expression; an explicit conversion from one type to another, as compared to promotion, which is implicit.

catch The code that handles a particular type of exception is said to catch that type of exception, so that it will not normally be propagated further. In Java, this is the role of a `catch` clause in a `try-catch` statement.

CDC Connected Device Configuration. This defines a core set of class libraries along with a virtual machine for higher specification mobile devices such as PDAs and set-top boxes.

CGI Common Gateway Interface. A protocol that defines how a web server should take user data as input, usually supplied through an HTML form, process it and generate HTML as output.

`char` A type used to store individual characters, such as `'a'` or `'$'`. A string is made up of characters.

check box A visual component that allows the user to toggle a choice.

checked exception An exception of a type which the Java Language requires to be either declared as thrown or handled by the method in which it could arise.

circular buffer A data structure that acts like a buffer except that when it is full the oldest data is overwritten by subsequently stored data. This can be implemented by an array with front and end markers. A circular buffer is also referred to as a cyclic buffer or a bounded buffer.

class A basic structuring concept for object-oriented programming languages; a sort of template, specifying the structure of the data and the methods for objects of the class.

class definition The entire description of a class, from its header to its closing curly bracket. A class definition describes a category of objects.

class header The first line of a class definition (before the opening curly bracket), including any access modifier for the class and the class name.

class hierarchy A diagram or other representation of how a number of classes are related by inheritance.

class library A collection of reusable classes intended to extend the functionality of the basic Java language. May be provided as part of the standard Java offering (in which case, it is often used synonymously with the term ‘standard package’) or may be from another source.

class method A method defined once for the whole class, and not for each object (instance) of a class; same as *static method*.

class type The type of an object; same as *reference type*.

class variable A variable for which there is one copy for all objects of the class; same as *static variable*. To be contrasted with *instance variable*.

CLDC Connected Limited Device Configuration. This defines a core set of class libraries along with a virtual machine aimed at small resource devices and intermittent network connections such as mobile phones.

client A computer on which users run application programs or user interface code. Clients request services from servers.

code block A portion of code enclosed by a pair of curly brackets.

cohesion Belonging together. We aim to have classes that show cohesion, meaning that all the methods and data they define belong together in the sense that they are part of the same piece of work.

collection class A class whose objects act as containers – they store other object references and regulate access to objects in some way.

collision rectangle A rectangular region surrounding a sprite and used in determining when two sprites have 'collided'.

combo box A visual component that behaves like a drop-down list. Normally used to display a large number of options or in situations where the list of options changes dynamically. The user can select only one of the options offered.

compilation unit A source file or group of source files, which can be compiled independently of other parts of the system.

compiler In Java, a program that translates source code into a portable form known as bytecode. During the translation process, the compiler will indicate any syntax errors in the source code.

composition A way of creating more complex objects by using object types as data members. An object of the class is said to be 'composed of' objects from one or more classes. See also *has-a*.

concrete class A fully implemented class, for which objects can be created, in contrast to an *abstract class*.

concrete method A fully implemented method, having both a header and a body, in contrast to an *abstract method*.

concurrent system A system allowing various activities to be at some stages between their starting and finish points.

conditional processing Code to change the flow of a program's execution.

connectionless service Communication across the internet where the computers simply send and receive data without setting up any prior connection; data can therefore be sent quickly but some may be lost or corrupted. The internet uses the UDP protocol for this.

connection-oriented service Communication across the internet where the two computers establish a connection to each other by exchanging special packets before sending data – similar to establishing a voice telephone call before you start speaking; the internet uses the TCP protocol for this.

constant A data item whose value cannot be changed. In Java, this is done using the keyword `final`. A reference type constant always refers to the same object, but the state of that object may be changed.

construct To create an object.

construction See *object construction*.

constructor A piece of code that is invoked to create an object. The constructor code within a class specifies any actions to be carried out when an object of the class is created. In the absence of an explicit constructor, an object is created with default initial values for its data items.

container A graphical component that is able to hold other components.

control structure A structure such as the `if`, `for` and `switch` statements which determine the sequence of execution of code. Same as selection statement.

control variable A variable used to determine whether a loop should continue executing or terminate.

data type A characterization of the nature of an expression or variable, for example, the `boolean` data type has just two values, while a character can take on any value within the Unicode character set. Same as *type*.

datagram A data packet. When sent using UDP or a similar protocol, its arrival, arrival time, and content are not guaranteed.

DBMS Database Management System. It is responsible for interacting with users of database systems and organizing the reading and writing of the database to satisfy their requirements.

declaration The part of a Java source code that names a new data storage area, such as a variable or a constant. It associates an identifier with a memory location.

declare an exception To state, in a method header, the class of an exception object that could be generated by that method.

`default` The Java keyword used in the `switch` statement to begin a block of code that should be executed when the `switch` argument does not match any of the `case` selectors.

default access The access level that arises from no access modifier having been specified for a class member; sometimes known as package access.

default constructor The zero-argument constructor that the Java system provides as a convenience when no constructor for a class has been specified; it initializes an object to a default state.

default value The value assigned to an item such as a variable, if no explicit action is taken to give it a value. For example, instance variables of type `int` are initialized to a default value of zero, even if they are not set explicitly to any value in the class constructor.

defensive programming Anticipating possible errors that might arise during program execution, and including code to prevent them or to take appropriate action if errors do occur. To be contrasted with *Design by Contract*.

delimiter A character that separates tokens and is to be otherwise disregarded. It is used to parse strings in a string tokenizer.

deque Short for double-ended queue. A data structure that combines the features of a stack and queue; items may be inserted in one end, and removed from either end.

Design by Contract Specifying methods with a clear definition (a 'contract') of what is expected before and after the method executes. If the expected conditions hold before the method executes, the method guarantees to produce the specified conditions after it has completed. To be contrasted with *defensive programming*.

distributed system A collection of computers at different locations, connected by communications links. The functions and data of the system are distributed across these computers.

DNS See *Domain Name Service*.

Domain Name Service (DNS) A service on the internet that keeps track of the numeric address (IP address) associated with each symbolic address.

domain name The part of a symbolic address associated with a particular internet site or organization, such as `google.com` or `open.ac.uk`. Domain names have two or more parts, separated by dots.

dotted quad notation The notation used for a numeric address, which uniquely identifies a device on the internet, also known as IP Version 4 (IPv4) addressing; the address consists of four numbers in the range 0 to 255, separated by dots, such as `193.22.33.201`.

`double` The Java double precision floating-point type, occupying 64 bits.

drop-down list A list of items which is displayed when the user chooses to reveal the list and from which the user can select any of the items.

dynamic (of a process) Code under execution.

embedded system a piece of hardware and software dedicated to a particular task, often as part of a larger system,

empty string The string with no characters in it, denoted by the literal `""`.

`equals` A method that all classes inherit from the fundamental `Object` class; it is used for testing whether or not two objects of that class are equivalent in some appropriate sense for that class of objects. In most cases, each class should override this standard `equals` method with a method appropriate to objects of that class, which may involve comparing each of the instance variables of the two objects. Some methods of collection classes require that the objects they store have an appropriate `equals` method.

escape sequence A sequence of characters not to be interpreted literally. The first character in any escape sequence is the backslash character.

event A signal to the programming environment that something has happened, for example, a button has been clicked.

event listener See *listener*.

event source A component that generates an event.

event-driven programming A style of programming whereby code is executed when an event occurs rather than in a fixed procedural fashion.

exception A Java object that is created when the system detects certain kinds of error during program execution. The object contains information about the cause of the error.

exception handler A section of code which carries out exception handling – indicated by a `catch` clause in a `try-catch` statement.

exception handling Taking action when an exception has occurred, in order to allow a program to recover in a systematic fashion or to exit gracefully.

exception propagation Passing an exception object back to the program context in which the code causing an exception was invoked, rather than handling the exception locally. Propagation may occur along a chain of contexts until an appropriate exception handler is found or the program exits with the error condition that was raised.

execute The process of carrying out the statements specified in a program. In the case of Java, statements are compiled into bytecode and executed by a Java Virtual Machine (interpreter).

explicit initialization Assignment of a value to a variable outside of a constructor.

expression A combination of variables or literals that can be said to have a value, often in conjunction with one or more operators.

`extends` The Java keyword indicating that one class inherits from another.

factor out (common behaviour) The process of identifying similar behaviour, as indicated by the methods required, across a number of classes and defining a class hierarchy to avoid duplication.

`false` One of the two `boolean` values, the other being `true`.

`final` The Java keyword for a constant.

finished state A process is said to be in the finished state when it has completed executing. In a Java thread, this is when the `run` method has completed.

`float` The Java floating-point type occupying 32 bits.

floating-point type A numeric type to represent numbers that may include fractional parts, for example, 3.8. The term 'floating point' refers to the way such numbers are stored, that is, the significant digits are stored separately from the order of magnitude. In Java there are two such types, `float` and `double`.

flow control structure See *control structure*.

flushing Clearing the buffer used in buffered input/output, ensuring that any data that was being temporarily stored is immediately sent to its destination.

formal argument A name in a method signature used as a place holder for an actual argument to be received and assigned to the formal argument when the method is invoked. Same as *formal parameter*.

formal type parameter A formal parameter whose actual argument is a type as opposed to a primitive or reference value. Formal type parameters are used in Java generic collections to allow instantiation of the collection for a particular type.

`for` **statement** A looping statement which indicates a starting value of a control variable, a condition to evaluate to determine when the loop should end, and an adjustment to the control variable, together with statements to be performed while the loop condition holds.

frame A user-interface window holding graphical components.

fully qualified name The complete name of an item. For example, the package name together with the simple name of a class or other item that can be imported from a package as in `java.net.URL` from the standard package `java.net`. See also *qualified name*.

game loop A loop whose purpose is to control execution of a game.

garbage collection The process of destroying objects that are no longer referenced and allowing the memory they occupied to be reused. In certain programming languages, such as Java, this process is automated.

getter method A method returning some attribute of an object. Also known as *accessor method*.

Google Almost certainly, at the time of writing, the most popular search engine.

graphical user interface (GUI) The interface to a program that the user sees (or hears) on the screen, usually represented in terms of windows and other graphical components.

graphics A means of producing visual images on screen. The special `Graphics` class provides the graphical context for drawing. The actual drawing is achieved through the `paint` method, for which programmer needs to provide an implementation.

handshake An exchange of a number of special packets of data in order to establish a connection across the internet between two computers.

has-a Expresses that a class contains as instance data a reference type; in other words, that composition is being used. This is distinguished from an inheritance relationship between classes, where the term *is-a* is sometimes used.

hash table A data structure that can store items in key–value pairs. The hash table is constructed so that the value corresponding to a given key can quickly be retrieved. The standard implementation in Java 2 is the `HashMap` class, which replaces the `Hashtable` class from earlier versions of Java.

`HashMap` An implementation of a hash table, forming part of the Java Collections Framework – replaces the `Hashtable` class from earlier versions of Java.

`HashSet` An implementation of a mathematical set, based on a hash table, forming part of the Java Collections Framework.

helper method A private method of a class that performs some internal work, but does not directly provide a service required by clients of the class.

heterogeneous collection An object of a collection class containing references to objects of a variety of different classes.

hiding Occurs when an inherited variable is redefined by a subclass, so that the superclass variable is not accessible using its simple name.

high level user interface A set of user interface classes designed to provide components that are portable from one device to another as the detailed implementation in terms of, for example, font size, scrolling and display are taken care of by the particular device.

homogeneous collection An object of a collection class containing references to objects that are all of the same class.

host A computer forming part of a distributed system such as the internet.

HTML HyperText Markup Language – a language for constructing web pages that is understood by web browsers. The statements in HTML determine how the page is displayed.

HTTP HyperText Transfer Protocol, defining the rules for communication between a client computer (usually using a browser) and a web server.

identifier The name for an item in a program, such as the name of a variable or a method.

`if` **statement** A selection statement allowing flow of program control to be changed.

immutable An entity is said to be immutable if, once created, it cannot be changed. Strings are immutable in Java.

implement Provide code to form the body of one or more methods, e.g. those defined in an interface or an abstract superclass. A concrete class that implements a given interface must implement all the methods defined by that interface.

`implements` Java keyword used in a class header to indicate that the class will implement the interface specified.

`import` This indicates to the compiler where to find any classes that may be required, but are not defined locally. The `import` declaration may name a specific class or other item, or may specify import-on-demand.

import-on-demand This indicates to the compiler a package to be searched for classes not defined locally. For example, `import java.net.*` makes available any accessible types in the package `java.net`.

index A means of referring to a location in an array.

information hiding The principle that parts of a program (such as an object) should make available as little as possible of the details of how the data they hold is structured or formatted; access and modification of the internal data of an object should normally be only through invocation of its methods.

inheritance A relationship between two classes: a superclass and a subclass. The subclass normally represents a more specialized version of the superclass. It has the same methods and internal data structure as its superclass, but may also add additional methods or variables, and may redefine (override) the implementation of the methods it inherits from the superclass.

inherited An inherited member can be used in a subclass as if it had been defined by the subclass.

initial state A process is said to be in the initial state when it has just been created. In Java, a thread is in its initial state after creation with `new`.

initialization The process of placing a value in a memory location for the first time, so that a variable has a predictable starting value.

in-line comment A comment that occupies the latter part of a line of Java code and normally describes that code. It is preceded by a `//` symbol.

inner class A class defined within the definition of another class.

instance variable A variable defined within a class as part of the state of an object. Each object (instance) of the class has its own copy of the instance variables for the class and can set their values independently. This contrasts with a class variable, which is shared by all the instances of a class.

`int` The most commonly used integral type in Java, occupying 32 bits of memory and the type of integer literals in Java.

integral type Any type whose values are positive or negative whole numbers. In Java the integral types are `byte`, `short`, `int` and `long`.

`interface` In Java, a named construct similar to a class, which defines a reference type and provides a specification of behaviour of objects of the class in the form of abstract methods. A concrete class that is declared as implementing the interface must provide implementations for the abstract methods. A Java `interface` is not allowed to have instance variables, but may contain constant data (implicitly declared as `final`).

internet An abbreviation of 'interconnected networks'. The internet is the worldwide network of networks that is the basis of the World Wide Web. It uses client–server technology and provides a means of exchanging information, including email. Some references capitalize 'the Internet' to make a distinction from 'an internet'.

Internet Domain Name Service See *Domain Name Service*.

Internet Protocol (IP) The basic protocol for the internet, which is concerned with enabling individual data packets to get from their source to a destination, without any error checking or reordering of packets.

interpreter In Java, the software that translates and executes the bytecode which is the result of compiling Java source code. It does this by translating the bytecode into the native code of the computer where the Java program is to run, and causing the native code to be executed immediately.

invoke a method Cause a method to be executed. Same as call a method.

invoking a constructor Causing a constructor to be executed.

IP address The numeric address of a device on the internet; in IP version 4 this requires 32 bits; IP version 6 addresses are 128 bits long.

is-a Indicates an inheritance relationship between two classes. If class `X` is a subclass of class `Y`, we say `X` is-a `Y`, which means that `X` is a more specialized version of `Y`. This is contrasted with the 'has-a' relationship which describes composition.

iterate To repeat a sequence of instructions, so that one can process a number of items in a collection.

iteration The process of repeating a sequence of instructions or considering in turn the items in a collection.

iterator An object which can be used efficiently to access the items of a collection in turn, without needing to know the details of how the collection object works. The order in which the items are visited depends on the underlying properties of the collection object, for example, whether or not the objects are stored in some specific order.

J2EE Java 2 Enterprise Edition. Java edition specifically aimed at building large scale systems.

J2ME Java 2 Micro Edition. Java edition designed for use on smaller systems with limited resources, such as mobile devices.

J2SE Java 2 Standard Edition. The 'core' Java technology platform.

JAD Java Application Descriptor. This contains information required by a mobile device to decide whether it is possible to download files containing a MIDlet.

JAR JAR files provide a means of packaging groups of files, similar to zip files, and can be used for tasks like data compression, archiving, decompression, and archive unpacking. It can be used to hold a MIDlet's suite and its resources.

Java A programming language released by Sun Microsystems in 1995, initially intended as a programming language for consumer electronics. It became popular due to its suitability for internet programming.

Java Card A technology, defined by Sun Microsystems, that enables smart cards and similar devices with very limited resources to run small Java applications, using a very restricted subset of the Java APIs (even more restricted than those defined for J2ME).

Java class hierarchy The class hierarchy expresses the inheritance relationship between classes (superclasses and subclasses); the root of the Java hierarchy is class `Object`, from which all classes ultimately inherit, including user-defined classes.

Java Collections Framework A consistent and carefully integrated group of classes and interfaces, which provide specifications and implementations of a number of standard collection types, including lists, sets and hash tables. It was introduced as part of Java 2 and largely supersedes earlier collection classes such as `Vector` and `Hashtable`.

JavaDoc An automatic documentation tool for Java, which converts specially formatted comments to documentation in the standard form, used in the Java API documentation.

Java edition A means by which Java software can be made available to a very wide range of devices from very small scale systems with few resources, such as mobile phones, to large networks of powerful servers with very extensive hardware resources.

Java-enabled browser A web browser that makes use of an installed Java Runtime Environment (JRE) to run Java applets.

Java version One of the periodic releases of a Java edition. Versions are released to add new features or to correct faults that become evident in earlier versions and are identified by version numbers such as Java 2 SE version 5.0.

Java Virtual Machine (JVM) Part of the Java Runtime Environment software that behaves as if it were a hardware platform in that it has its own 'native' code, bytecode. See also *interpreter*.

JDBC Java Database Connectivity. A set of classes, structured into packages, which provides an API for accessing databases from Java programs. It has the advantage that it works with a variety of databases.

JDBC driver A software component that provides the interface between the Java program and the DBMS. The JDBC driver for a specific DBMS, such as Oracle or Access, communicates with the DBMS in the specific language required by that DBMS.

JIT compiler See *Just-in-time compiler*.

JRE Java Runtime Environment. The collective name for the Java Virtual Machine, core classes and other files that support execution of Java bytecode.

JSP Java Server Pages. A JSP is a page of HTML and text, which may also contain sections of Java code (so-called scriptlets), indicated by special tags. Particularly useful if the response page contains a lot of static HTML code.

Just-in-time compiler Operates by compiling parts of the Java program through to native code only just before it is first required for execution. This means that Java can achieve high performance if required, even in embedded systems.

key–value pair The way in which items are stored in a map, of which a hash table is an example. The value is some data associated with the key. Each key must be unique and is used to quickly find the corresponding data. For example, the key could be a person's name and the value their telephone number. Several people could share the same telephone number but the names must be unique for the hash table to work properly.

keyword A word that has a special meaning in a Java program and, therefore, cannot be used for any other purpose.

label A display area for a piece of text often used to label other visual components.

layout manager This controls the way in which the components of a user interface are laid out in a container. Layout managers can have different layout styles, for example, `GridLayout`, `FlowLayout` or `BorderLayout`.

legacy collection class One of a number of collection classes that have been superseded in more recent versions of Java. These include `Vector`, `Hashtable`, `Properties`, `Dictionary`, `Stack` and `BitSet`. They are widely used in older Java software, so it is useful to know about them.

library class A class available for reuse in a library.

line comment A comment extending from a `//` symbol to the end of the same line, often preceding the line of code that it describes.

linked list A data structure in which items of data are arranged in a sequence, linked together by each item containing a reference to the next item. Access to the contents of a linked list is therefore necessarily sequential rather than random (as for an array).

`LinkedList` A class implementing the linked list data structure, forming part of the Java Collections Framework.

list (1) A data structure in which items of data are accessed sequentially rather than randomly (as for an array). The `List` interface in the Java Collections Framework specifies the required behaviour for lists in Java. This may be implemented in a variety of ways, with a linked list being one common approach.

list (2) A visual component that holds a number of items from which the user can select one or more items. The Java class `JList` can be used to create such a list, and the list can be given a scroll bar using the `JScrollPane` class.

listener An object which is registered with a visual component and is capable of receiving and then handling an event.

literal A representation of an actual value of a type, a constant expression, for example `3` or `'c'`.

local host The computer on which a networked program is running accessed using the loopback address.

local variable A variable declared within a code block, often inside a method's body.

localization The process of developing versions of software suited to different countries.

lock The means by which the Java system determines which process has exclusive access to the synchronized code of an object. Since each object has only one lock, only one of its synchronized methods can execute at any one point.

logical expression Any expression involving a logical operator.

long An integral type occupying 64 bits of memory.

loopback address A special IP address (`127.0.0.1`) which always accesses the local host; useful for testing two communicating programs on the same computer.

lost update problem The situation where, during the execution of an operation that consists of several suboperations, the intermediate results (that is, results of sub-operations) become accessible to other operations, with the potential danger that a previous update becomes lost.

low level user interface A user interface class giving the programmer low level control over how what is drawn on the screen is displayed and how the user is able to interact with the screen.

map A data structure that consists of a number of key–value pairs. The `Map` interface in the Java Collections Framework specifies the required behaviour for maps in Java. This may be implemented in a variety of ways, with a hash table being one possible approach.

markup language A set of instructions, normally in the form of 'tags' which can be distinguished from normal text and provide semantic information about the text, for example they may define how it should be formatted or displayed. HTML and XML are examples of markup languages.

member A variable or method of a class.

menu A visual component that provides a means of offering choice to the user; located immediately below the title bar of a frame.

method An encapsulation of statements within a class that defines an operation which can be carried out by an object of that class.

method header The first line of a method, including its access modifier (if any), return type, method name and arguments.

method invocation The process of causing a method to be executed. Same as calling a method.

MIDlets Applications that run on so-called Mobile Information Devices, such as PDAs and mobile phones. MIDlets can be constructed using J2ME.

MIDP Mobile Information Device Profile. A profile that targets particular categories of device and particular applications. Each profile has APIs additional to those in the configurations.

multiple inheritance The situation where a class inherits from more than one superclass – possible in some languages, but not in Java because a class can only extend one other class. Interfaces allow a weak form of multiple inheritance by allowing specification to be 'inherited' from more than one superclass.

mutator method A method which, when invoked on an object, potentially alters the state of the object. Setter methods are examples of mutators.

name clash The situation where two or more identical names are in scope at the same time, so that the compiler cannot determine which is required.

name service A service that provides information about a resource, given the symbolic name of the resource; for example, the Internet DNS.

native code Instructions that can be executed by a particular hardware platform, typically represented as binary numbers.

native compiler Translates Java source code directly into the native code required by the processor, thus cutting out the bytecode interpretation process.

`new` The Java keyword used to invoke a constructor in most contexts.

non-blocking call When a call to a method does not interrupt the execution of the code.

numeric address A set of numbers that uniquely identifies a device on the internet; in IP version 4, the address can be written in dotted quad notation – four numbers in the range 0 to 255, separated by full stops, such as `193.22.33.201`.

object A self-contained component of a program that has a state – defined by the values of its instance variables – and a number of methods, some of which may be used to access or modify the state of the object. Objects belong to a class; all objects of a particular class have the same methods and the same internal structure for their data, although each object typically has different values for that data.

object construction Creating an object using a constructor.

object wrapper A class whose objects are used to hold primitive data for use in situations where objects are required, such as storage in collection classes. For example, the class `Integer` is used to wrap data of type `int`.

off-screen buffer An area of memory which holds graphics data that is to be displayed on an output device.

operator A symbol denoting some operation to be performed on one or more arguments, for example `+`.

operator precedence The rules determining the order in which operators are evaluated in an expression.

overloading Where a class has more than one method of the same name, this method is said to be overloaded. However, each of these identically named methods must have a different signature – that is, it must be possible to distinguish each method from the others by the number and/or the type of its arguments.

overriding The redefinition of an inherited method in a subclass. The subclass must declare the method again, with the same signature and compatible return type as the method in the superclass, and then redefine the implementation appropriately for the subclass.

package A named group of related classes and interfaces. Large programs typically consist of a number of packages – they help with management of the software development and maintenance processes.

package visibility The access level whereby items are only accessible to classes within the same package. This is the default if a specific access level such as `private` or `public` is not specified.

packet A fixed-size unit, consisting of some data together with a packet header, to be sent across the internet.

packet header The part of a packet that contains details about the data being transported, such as its length and the addresses of its source and destination.

packet switching The process used for transporting internet data, whereby the data is broken up into a series of fixed-size packets and each packet is sent and received separately. Packets for a particular message may travel by different routes across the network, making the data transfer process more resilient to faults or damage to the network.

panel A container which can contain other containers and also GUI visual components, such as buttons.

pattern A string representing a set of strings, possibly an empty set or a set with only one member.

pattern matching The process by which it is determined whether a string matches a pattern or not.

peer-to-peer (P2P) A distributed system architecture where each computer has similar capabilities and responsibilities – each can act as a server as well as a client.

pixel Short for 'picture element'. A pixel represents a single dot in a displayed image. When drawing objects on the screen with the `paint` method, measurements are in terms of pixels.

polling The process of cyclically checking for some event happening.

`Polygon` A Java class used to draw line segments forming closed two-dimensional shapes.

polymorphism An approach where a number of objects of different, but related, classes may respond differently to invocation of a given method. In Java, a variable may reference an object of its declared class or an object of any of its subclasses. Invocation of a method common to the declared class and to its subclasses may have different effects depending on the actual class of the object referenced at run time. It can help make software more extensible and reusable. Derived from Greek meaning 'having many shapes'.

port A numbered conduit into a host on the internet, which allows a client to request a particular kind of service and allowing a server to listen for particular kinds of requests on a specific port; for example, port 80 is normally used to access a web server.

portability The extent to which software can be reused on another platform without changes; a measure of the effort required to transfer the system from one hardware platform and/or software environment to another.

postfix operator An operator that appears to the right of its argument(s).

prefix operator An operator that appears to the left of its argument(s).

primitive (data) type One of a number of types of simple data, such as integers, characters and floating-point numbers; to be contrasted with reference types, which refer to where in memory an object is located.

`private` The Java keyword designating a class member as only for use within the class and not accessible to subclasses or via an object reference. When it is used in a definition for the class itself, it designates this class as an inner class.

programming-in-the-large Programming techniques for specifying the ways in which parts of programs may interact with one another. To be contrasted with *programming-in-the-small*.

programming-in-the-small Programming of the internal parts of programs or methods. To be contrasted with *programming-in-the-large*.

propagate an exception See *exception propagation*.

promotion The automatic conversion of one type to another.

`protected` The Java keyword designating a class member as only for the use of subclasses or via an object reference by classes in the same package.

protocol A set of rules that defines the details of how computers or other devices can communicate, so that both sides can understand the communication; for example, HTTP or TCP.

protocol levels Higher level protocols (such as HTTP) assume that the lower level protocols (such as TCP and IP) handle the lower level details of the communication process; designing protocols as a number of levels simplifies the software needed to manage each level.

`public` The Java keyword designating a class member as available for access by any subclasses or via any object reference.

qualified name A name preceded by an object identifier and a dot; for example, `myVariable.name` is a qualified access to the member called `name`. See also *fully qualified name*.

radio button A visual component that allows the user to select exactly one choice from many.

real-time Java A version of Java with specialized classes for real-time systems.

real-time system A software or hardware system that needs to respond within specified time limits in order to function properly.

`Rectangle` A Java class used to handle rectangles in a graphics context.

reference type Any type defined by a class definition; a type that stores a reference value, which indicates where in memory an object is located; to be contrasted with a primitive type, which is not dependent on a class definition, is often platform dependent and stores a data value.

reference variable Any variable of a reference type.

regular expression A pattern of characters which is used to detect substrings.

relational operators An operator describing relationships between its arguments, for example `<`, `<=`, `==`, `>`, `>=`, `!=`. Each operator can be used with a variety of types to return a `boolean` value, such as, `2 < 3` is `true`.

request The part of a protocol exchange that asks for information; for example, a HTTP request for the contents of a web page.

response The part of a protocol exchange that returns information or a status message corresponding to a request.

RGB colours They are red, green and blue colours. By specifying values for these three primary colours, a huge range of colours can be specified and used for the appearance of text and other graphical user interface components. The Java class `Color` uses the RGB colours approach.

RTSJ Real-Time Specification for Java. This is a standard specification of what is required from a real-time version of Java.

runnable state A process is said to be in the runnable state when it has the opportunity to be selected by the scheduler to be the process that will actually run.

running state A process is said to be in the running state when it has been selected by the scheduler to run, and is thus actually executing at that point.

sandbox The metaphor of a children's play area (containing sand) is used to describe Java's security model. The sandbox imposes strict controls on what certain kinds of Java programs (such as applets) can do.

scheduler A part of the Java system that decides which thread will run next.

scope The part of a program in which a variable identifier is meaningful.

scriptlet In a JSP document, a section of Java code enclosed in the special tags `<%` and `%>`.

scroll bar A device to allow the user to visually select a value from a range of values. Scroll bars can be vertical or horizontal.

search engine An internet-based program that allows its user to search for information on the net.

selection statement A statement used to singly change the flow of control in a program, for example, an `if` or `switch` statement.

self-launching thread A thread that is started as a side-effect of another action, that is, without an explicit request from a programmer to start the thread.

separation of concerns This ensures that the work of each method is self-contained and well-defined; likewise at the level of a class, we aim to ensure that a class is only concerned with one job.

server A computer that manages data, printers or network traffic and provides services to clients.

server context An environment for hosting and managing visiting aglets on a server and providing protection for the host server from malicious code.

servlet A module of Java code that is stored and run on a web server, which is used in web-based applications. Java servlets can be used to implement a wide variety of functions, such as processing data, accessing databases and constructing complex web pages dynamically for return to the client.

servlet container Software environment for a servlet to run in.

setter method A method used to set an attribute of an object. A setter method is an example of a mutator method.

`short` The integral type occupying 16 bits of memory.

signature The name of a method together with the number and types of its arguments; this defines what is needed in order to invoke the method.

signed applet An applet that has a digital signature to prove that it came from a particular author. A user can choose to give a signed applet more privileges than ordinary applets.

simple name The name of an item without the name of the higher level structure that contains it. It needs to be distinguished from a fully qualified name. For example, a data field identifier without the object it belongs to, or the name of a class without its package.

socket The software mechanism that allows programs to transfer data across the internet using TCP/IP; a `Socket` object in Java is associated with a port number and the address of the other host.

source code The language in which a programmer writes a program. For example, when we say a program is written in Java then, strictly speaking, we mean that the source code of the program is written in Java. The Java compiler translates the source code into bytecode.

spider A program which traverses the web looking for information. Spiders are often associated with search engines.

SQL Structured Query Language. A programming language used to access and manipulate data in relational databases.

SQL query The primary mechanism for retrieving information from a database. A query consists of questions presented to the database in a predefined format.

standard Java package A library of predefined classes and other items provided as part of the standard Java offering. For example, the `java.net` package contains classes and related items for use in writing networking and communications software in Java.

state The current values of the instance variables defined by an object, which together form a snapshot of the properties of the object at a point in time.

statement A piece of code in Java (or other programming languages) that forms a unit of execution. This is usually terminated by a semicolon. It does not necessarily translate into one bytecode instruction.

static (of code) Program code can be seen as essentially static, to distinguish it from the dynamic process (or thread) that is executing this code. In Java, the keyword `static` is used.

static method A method defined once for the whole class, and not for each object of a class; same as *class method*.

static variable A variable for which there is one copy for all objects of the class; same as a *class variable*. To be contrasted with *instance variable*.

stream A sequence of bytes representing a flow of data from a source (for example, a disk file or a keyboard) to a destination (for example, memory or the screen).

string A sequence of characters.

`String` The Java reference type used to work with sequences of characters.

string tokenizer A facility for extracting tokens from a text string; provided in Java by objects of the `StringTokenizer` class.

strongly typed language A language that performs checks to ensure that types are not incorrectly converted.

subclass A class that inherits from another class (its superclass).

subpackage A package that is a member of another package, providing additional classes, interfaces or subpackages to those supplied by another package. The dependency is indicated by subpackages having the first part of their name in common with the package on which they depend. This is useful for organizing sets of related classes.

`super` The Java keyword used to refer to a superclass. Either it begins a qualified name, as in `super.calculate()`, to refer to an accessible (but hidden) member in a superclass, or it appears as a superclass constructor invocation, as in `super()`, where it must be the first line of a constructor.

superclass A class from which one or more other classes inherit (these are its subclasses).

Swing The Java library of graphical interface elements found in the `javax.swing` class library.

`switch` **statement** A selection statement allowing choice of one of several paths under the control of an argument.

symbolic address a hierarchical, text-based name that uniquely identifies a particular device on the internet, such as `www.open.ac.uk`.

synchronised block A Java structure which ensure orderly concurrent access to shared data.

synchronized Enforcing sequential (as opposed to concurrent) access to certain code. In Java, the keyword `synchronized` is used.

system architecture In this context, architecture is a high level description for the ways of structuring the network of computers in a distributed system, for example, the client server architecture or the peer-to-peer architecture.

system protocol A protocol that relates to the basic operation of the internet system, getting data from one place to another, rather than to any particular use or application of the internet; for example, TCP or IP.

TCP/IP Transmission Control Protocol/Internet Protocol. The two protocols that form the basis of network communication on the internet.

text area A visual component that acts as an input/output area where the user can enter characters or read displayed characters over several lines.

text field A visual component that acts as an input/output area where the user can enter characters or read displayed characters on a single line.

text input/output Input or output where data is converted to or from text format – Unicode, in Java. (See *binary input/output*.)

thin client In web applications, much of the processing can be carried out on the server, and thus the client software can be kept simple and responsive – a so-called thin client.

`this` Can be thought of as meaning 'this object'. The Java keyword meaning a reference to the object executing the code in which the word `this` appears. Can also be used to invoke a constructor, as in `this()`, from within another constructor.

thread Code under execution; a task or activity that can proceed simultaneously with other threads.

`Thread` Java class defining a thread of execution in a program. A programmer may subclass `Thread` to define a thread or implement the `Runnable` interface.

thread state A thread can be in different states, depending on whether or not it is currently executing (running state), or waiting to be given processing time (runnable state), or even waiting for a condition to become true so that it becomes ready to execute (blocked state). There are well-defined ways in which a thread can move from one state to the next.

throw To generate an exception object.

time out The expiry of a preset time interval causing a thread to stop running because it has used its allocated time to execute.

token An element in a string, such as a word or a number. Tokens are separated by delimiters.

Transmission Control Protocol (TCP) A protocol that is used to provide a connection-oriented service for transmission of data across the internet, with facilities to re-send any lost or damaged packets and to reassemble packets into the correct order, if necessary.

tree A data structure where items are related to each other by links that can be represented diagrammatically by a schematic tree (normally upside down with the root at the top). Trees offer an efficient way of storing and retrieving data in sorted order. There are many possible ways to implement trees. The `TreeMap` and `TreeSet` classes in the Java Collections Framework use trees to implement a sorted map and a sorted set, respectively.

`true` One of the two literal values of the boolean type, the other being `false`.

`try-catch` A statement in two blocks or 'clauses' – the first encloses code that may cause an exception, while the second specifies errors that should be handled and code to execute in that case. Optionally, there is a third clause, 'finally', containing code that will always be executed.

type A set of values from which a variable or expression may take its value together with the operations on those values.

unboxing The process by which a primitive value is extracted from an object wrapper. For example, extracting an `int` from an `Integer` object.

unchecked exception An exception of any type apart from checked exceptions. There is no language requirement to declare or to handle unchecked exceptions.

Unicode A character set containing characters from many languages, an abbreviation of Universal Code.

Uniform Resource Locator (URL) The unique location of a resource on the web, such as a web page, together with the protocol for accessing the resource; for example, `http://www.open.ac.uk/Staff/I_Newton.htm`.

User Datagram Protocol (UDP) A protocol that is used to provide a connectionless service for transmission of data across the internet. Data is sent quickly, but some data may be lost or corrupted in the transmission.

variable A named location in computer memory which may contain a primitive value (such as an integer) or a reference value linked to an object. The value associated with a variable can be changed, in contrast to that of a constant which cannot be changed.

version See *Java version.*

visual component A graphical component that provides the means by which users are able to interact with applications, for example, a button, menu or frame. In Java, the visual components are found in the Swing library.

visual programming A style of programming that allows user interfaces to be created by adopting a 'drag and drop' approach using a palette of components visual device. Many programming IDEs provide facilities for this style of programming, and will automatically produce the code necessary to produce the interface that has been created visually.

web server A computer which makes web pages available in response to requests from clients running browsers.

`while` **statement** a looping statement in which some statements are executed as long as some condition is `true`.

white space Characters in text, such as space, tab, new line or carriage return, which affect the positioning of visible characters.

wild card In general, a symbol in a command or statement that can match a number of different items. An asterisk is used as part of an import-on-demand statement to indicate that any of the accessible classes or interfaces in a specified package are to be available for import. For example, `import java.net.*` means make available any accessible items in the package `java.net`.

World Wide Web The resources on the internet accessible via HTTP. Loosely, it refers to the internet and resources accessible via any protocol.