

# The benefits of java

[Technology](#), [Computer](#)



## 5. 1. 2. 4 JAVA INCLUDES A LIBRARY OF CLASSES AND INTERFACES:

The Java platform includes an extensive class library so that programmers can use already existing classes, as it is, create subclasses to modify existing classes, or implement interfaces to augment the capabilities of classes.

Both classes and interfaces contain data members (fields) and functions (methods), but there are major differences. In a class, fields may be either variable or constant, and methods are fully implemented.

To use an interface, a programmer defines a class, declares that it implements the Interface, and then implements all the methods in that interface as part of the class. These methods are implemented in a way that is appropriate for the class in which the methods are being used. Interfaces let one add functionality to a class and give a great deal of flexibility in doing it.

A package is a collection of related Java classes and interfaces. The following list, however not complete, gives example of some Java packages and what they cover.

Java. lang: The basic classes. This package is so basic that it automatically is included in any Java program. It includes classes that intercepts with numeric, strings, objects, runtime, security, and threads.

Java. io: Package that includes classes managing reading data in input streams and writing data into output streams.

Java. util: Miscellaneous utility classes, including generic data structures, bit sets, time, date, the string manipulation, random number generation, system

properties, notification and enumeration of data structures.

Java. net: Classes for network support.

Java. awt: Classes that manage user interface components such as windows, dialog boxes, buttons, checkboxes, lists, menus, scrollbars, and text fields, the “ AWT” stands for Abstract Window Toolkit.

Java. awt. image: Classes for managing image data, including color models, dropping color flittering, setting pixel values, and grabbing snapshots.

Java. applet: The Applet class, which provides the ability to write applets, this package also includes several interfaces that connect an applet to its documents and to its document.

Java. sql: The JDBC API, classes and interfaces that access databases and send SQL Statements.

The first three packages listed, java. lang, java. io and java. util form the basis, they are basic classes and interfaces for general-purpose programming.

Java development kit version1. 1 added some new packages, with JDBC being one of them. Other new packages include such thing as Remote Method Invocation, Security and Java Beans, the new API for creating reusable components.

In Java, packages serve as the foundation for building other packages, as discussed in the following section.

5. 1. 2. 4. 1 JAVA IS EXTENSIBLE:

A big plus for Java is the fact it can be extended. It was purposely written to be lean with the emphasis on doing what it does very well, instead of trying to do everything from the beginning, it was return so that extending it is very simple. The JDBC API, the java. sql package, is one example upon which extensions are being built.

In addition to extensions there are also main tools being developed to make existing capabilities easier to use. For example, there is already a tool that greatly Simplifies creating and laying out Graphical User Interfaces such as menus, Dialog boxes and buttons.

#### 5. 1. 2. 4. 2 SECURITY:

It is important that a programmer not be able to write subversive code for Applications or applets. This is especially true with the Internet being used more and more extensively for services such as electronic commerce and electronic delivery of software and multimedia content.

The Java platform builds in security in four ways.

The way memory is Allocated and laid out: In Java an object's location in memory is not determined until The runtime, as opposed to C and C++, where the compiler makes memory layout Decisions. As the result, a programmer cannot look at a class definition and figure out how it might be laid out in memory. Also since, Java has no pointers, a programmer cannot forge pointers to memory.

The way incoming code is checked: The Java virtual machine doesn't trust any incoming code and subjects it to what is called thoughte code

verification. The thought code Verifier, part of the virtual machine, checks that the format of incoming code is correct

incoming code doesn't forge pointers, it doesn't violate access restrictions, it accesses objects what they are.

The way classes are loaded: The Java thought code loader, another part of the virtual machine, whether classes loaded during program execution are local or from across a network. Imported classes cannot be substituted for built in classes, and built in classes cannot accidentally reference classes brought in over a network.

The way access is restricted for untested code: The Java security manager allows user to restrict untested Java applets so that they cannot access the local network, files and other resources.

#### 5. 1. 2. 5 JAVA PERFORMS WELL:

Java performance is better than one might expect. Java has many advantages, such as having built in security and being interpreted as well as compiled, do have a cost attached to them. As a result, Java has done quite respectably in performance tests. Its performance numbers for interpreting thought codes are usually more than adequate to run interactive graphical end user applications.

For situations that require unusually high performance, thought codes can be translated on the fly, generating the final machine code for the particular CPU on which the application is running at run time. High level interpreted scripting language generally offer great portability and fast prototyping but

poor performance. Low level compiled language like C and C++ offer great performance but require large amounts of time for writing and debugging code because of problems with areas such as memory management, pointers and multiple inheritance. Java offers good performance with the advantages of high level languages but without the disadvantages of C and C++.

In the world of design trade-off,

#### 5. 1. 2. 6 JAVA IS ROBUST:

The multi platformed environment of the WEB places extraordinary demands on a program, because it must execute reliably in a variety of systems. Thus the ability to create robust programs was given a high priority in the design of Java. To gain reliability, Java restricts you in a few key areas to force you to find your mistakes early in program developments. At the same time, Java frees you from having to worry about many of the most common cause of programming errors. Because Java is strictly typed language, it checks your code at compile time. However, it also checks your code at run time. In fact, many hard to track down bugs that often turn up in hard to reproduce runtime situations are simply impossible to create in Java. Knowing that what you have written will behave in a predictable way under diverse conditions is a key feature of Java to understand how Java robust.

For example in C/C++ the programmer must manually allocate and free all dynamic memory. This sometimes leads to problems. For example some

programmers some times forget the free memory that has been previously allocated.

#### 5. 1. 2. 7 JAVA SCALES WELL:

Java platform is designed to scale well, from portable consumer electronic devices to powerful desktop and server machines. As a result, Java accommodates the need for low storage and for low bandwidth transmission over the Internet. In addition the Java operating system offers a standalone Java platform that eliminates host operating system overhead while still supporting the full Java platform. API makes Java ideal for low cost network computers whose sole purpose is to access the Internet.

#### 5. 1. 2. 8 JAVA IS MULTITHREADED:

Multithreading is simply the ability of a program to do more than one thing at a time. For example an application could be faxing a document at the same time it is printing another document. Or a program could process new inventory figures while it sustains a feed for current prices.

#### 5. 1. 2. 9 JAVA IS IMPORTANT TO THE INTERNET:

The Internet helped catapult Java to the forefront of programming and Java in turn has a profound effect on the Internet. The reason is simple. Java expands the universe of objects that can move about freely in cyberspace. In a network, there are two broad categories of objects transmitted between the server, your personal computer, passive info and dynamic, active programs. For example, when you read your e-mail, you are viewing passive

data. Even when you download a program, the program's code is still only passive data until you execute it. However, there is a second type of object that can be transmitted to your computer, a dynamic, self executing program. Such a program would be an active agent on the client computer, yet it would be initiated through the server. As desirable as dynamic, networked programs are, they also current serious problems in the areas of security and portability. Prior to Java cyberspace was effectively closed to half the entities that now live there. Java addresses these concerns and doing so, has opened the door to an exiting a new form of program.