CHAPTER-1
JAVA HISTORY

Introduction

- Java is a general-purpose, object-oriented programming language developed by Sun Microsystems of USA in 1991.
- James Goosling invented Java Programming Language(Originally called Oak) in 1991.
- Languages like C and C+ + had limitations in terms of both reliability and portability.
- However, they modelled their new language Java on C and C++ and thus made Java a really simple, reliable, portable, and Powerful language.

Features Of Java:

Compiled and Interpreted:

- First Java compiler translates source code into bytecode instructions.
- Bytecodes are not machine instructions
- Therefore, in the second stage, Java interpreter generates machine code that can be directly executed by the machine that is running the Java program.

Platform-Independent and Portable :

- It is platform-independent as Java follows the ‘write once, run anywhere’ approach.
- The most significant contribution of Java over other languages is its portability.
- Java programs can be easily moved from one computer system to another, anywhere and anytime.

Object-Oriented :

- Java is a true object-oriented language. Almost everything in Java is an object.
- All program code and data reside within objects and classes.

Robust and Secure:

- Java is a robust language. It provides many safeguards to ensure reliable code. It has strict compile time and run time checking for data types.
- Java also incorporates the concept of exception handling which captures series errors and eliminates any risk of crashing the system.

Distributed :

- Java is designed as a distributed language for creating applications on networks.
- It has the ability to share both data and programs.
- Java applications can open and access remote objects on Internet as easily as they can do in a local system

Simple, Small and Familiar:

- Java is a small and simple language. Java does not use pointers, pre processor header files, go to statement and many others.
- It also eliminates operator overloading and multiple inheritance.
- Java is modelled on C and C++ languages. So it is familiar to programmers.

Multithreaded and Interactive:

- Multithreaded means handling multiple task-simultaneously.
- Java supports multithreaded programs
- we need not wait for the application to finish one task before beginning another.. This feature greatly improves the interactive performance of graphical applications.

High Performance :

- Java performance is impressive for an interpreted language, mainly due to the use of intermediate bytecode.
- The incorporation of multithreading enhances the overall execution speed of Java programs.

Dynamic and Extensible:

Java is a dynamic language. Java is capable of dynamically linking in new class libraries, methods, and objects.

Difference between Java and C :

- Java does not include the C unique statement keywords goto, sizeof, and typedef.
- Java does not contain the data types struct, union and enum.
- Java does not support an explicit pointer type.
- A Java does not have a preprocessor and therefore we cannot use # define, # include, and # ifdef statements.
- Java requires that the functions with no arguments must be declared with empty parenthesis and not with the void keywords done in C.
- Java adds labelled break and continue statements.
- Java adds many features required for object-oriented programming

Difference between Java and C++:

- Java does not support operator overloading.
- Java does not have template classes as in C++.
- Java does not support multiple inheritance of classes. This is accomplished using a new feature called" interface".
- Java does not support global variables. Every variable and method is declared with in a class and forms part of that class.
- Java does not use pointers.
- Java has replaced the destructor function with a finalize( )function.
- There are no header files in Java.

Additional Feature:

- C+ + is a superset of C, Java is neither a superset nor a subset of C or C++.

HARDWARE AND SOFTWAREREQUIREMENTS

Java is currently supported on Windows 95, Windows NT, Sun Solaris, Macintosh, and UNIX machines.

The minimum hardware and software requirements for Windows 95 version of Java areas

- IBM: compatible 486 system
- Minimum of 8MB memory
- A Windows-compatible soundcard, if necessary
- A hard drive
- A CD-ROM drive
- A Microsoft-compatible mouse

JAVA SUPPORT SYSTEMS:

Support System	Description
Internet Connection	Local computer should be connected to the Internet.
Web Server	A program that accepts requests for information and sends the required documents.
Web Browser	A program that provides access to WWW and runs Java applets.

Support System	Description
HTML	A language for creating hypertext for the Web.
APPLET Tag	For placing Java applets in HTML document.
Java Code	Java code is used for defining Java applets.
Bytecode	Compiled Java code that is referred to in the APPLET tag and transferred to the user computer.

JAVA ENVIRONMENT

Java environment includes a large number of development tools and hundreds of classes and methods development tools are part of the system known as Java Development Kit (JDK) and the classes and methods are part of the Java Standard Library (JSL), also known as the Application Programming Interface (API).

Java Development Kit

The Java Development Kit comes with a collection of tools that are used for developing and running Java programs. They include:

appletviewer: Enables us to run Java applets (without actually using a Java-compatible browser).

java: Java interpreter, which runs applets and applications by reading and interpreting bytecode files.

javac: The Java compiler, which translates Java source code to bytecode files that the interpreter can understand.

javadoc: Creates HTML-format documentation from Java source code files.

javah: Produces header files for use with native methods.

javap: Java disassembler, which enables us to covert bytecode files into a program description.

jdb: Java debugger, which helps us to find errors in our programs.

Application Programming Interface :

The Java Standard Library (or API) includes hundreds of classes and methods grouped into several functional packages .

Most commonly used packages are:

Language Support Package: A collection of classes and methods required for implementing basic features of Java.

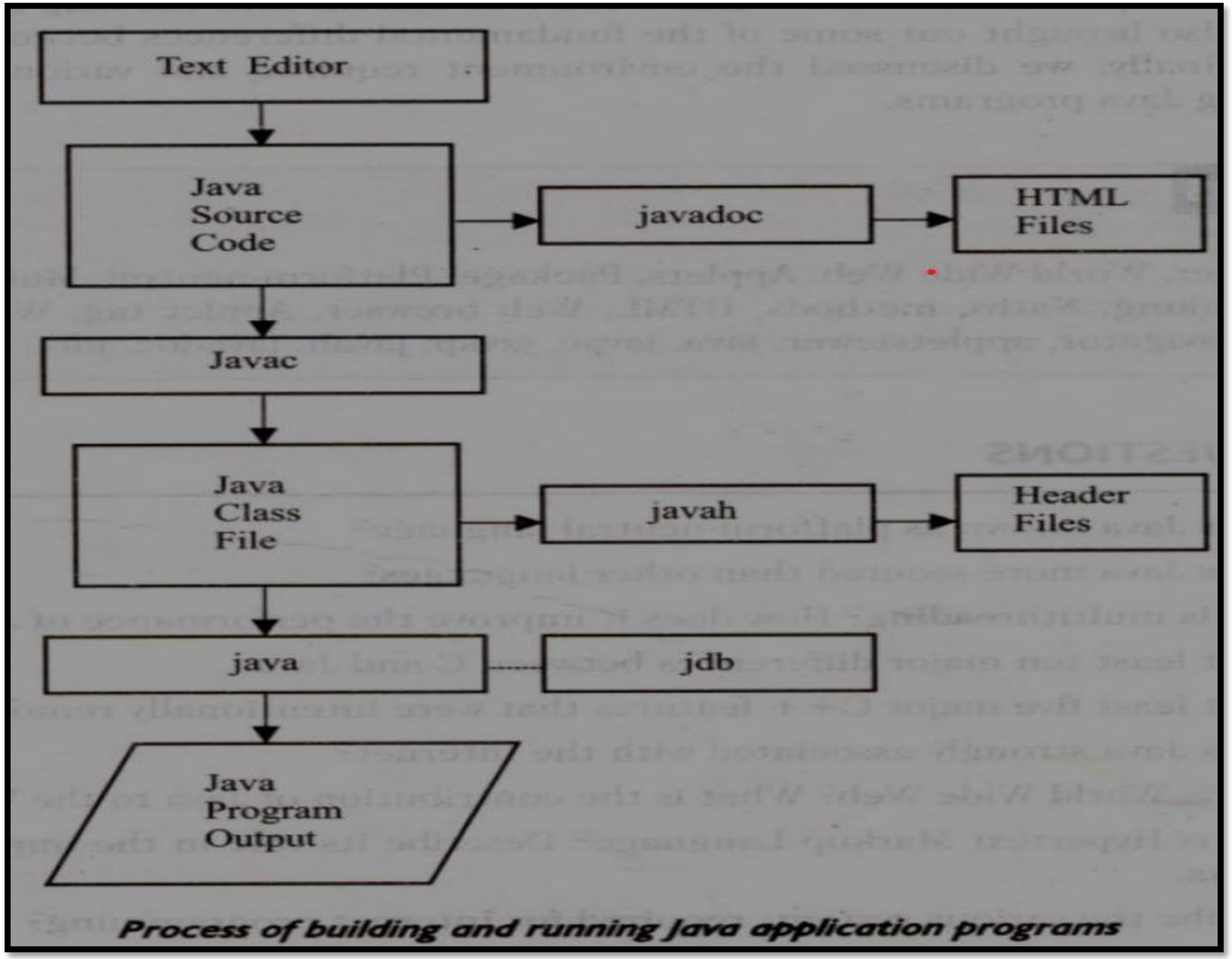
Utilities Package: A collection of classes to provide utility functions such as date and time functions.

Input/Output Package: A collection of classes required for input/output manipulation.

Networking Package: A collection of classes for communicating with other computers via Internet

Awt Package: The Abstract Window Tool Kit package contains classes that implements platform-independent graphical user interface.

Applet Package: This includes a set of classes that allows us to create Java applets



Process of building and running Java application programs