

Introduction To Java

Lesson 1

Two Parts of Computers

The two parts of a computer are **hardware** and **software**. The physical part is hardware and the program part is software.

Various Kinds of Software

There are **two kinds of software**. They are **System Software** and **Application Software**.

System Software

The software that is used to manage the computer system is **System Software**. It consists:

- 1) **Operating system**. E.g.: WINDOWS, System7.5, LINUX, UNIX, MS DOS etc.
It helps the user to operate both the hardware and software.
- 2) **Language processors**. E.g.: **Compiler**, **assembler** and **interpreter**.
These are used to convert the programs which are written in a language of one level to another level.
(Various levels of language are discussed in coming section)

Application Software

The software to satisfy user's needs is **Application Software**: It can be again divided into two:

- 1) **General purpose software and Customized software**
Programs for the common people and organisations etc.
- 2) **Programming software / Computer language**
It is used to develop Application software and System software. E.g.: **Java**, **C**, **C++** etc.

Various Levels of Computer Languages

- 1) **Low Level Language (LLL)**
Machine language is the Low Level Language, i.e., **0** and **1**. This is also known as **binary language** because it has only two digits. The **0** and **1** that represents absence and presence of electrical energy.
E.g.: **01000001** to represent the character **A**.
- 2) **Middle Level Language (MLL)**
Assembly language is the Middle Level Language. It is the second generation language and it consists variables and codes like ADD, SUBTRACT, PRINT etc instead of the binary digits.
- 3) **High Level Language (HLL)**
Java, **C**, **C++** are high level languages. It contains operators also apart from variables and codes.

Language Processors

The Language processors are **interpreter**, **assembler** and **compiler**. These are used to convert the programs which are written in a language of one level to another level.

- 1) **Interpreter**
The interpreter converts the program written in **high level language** into **low level language**, i.e., **the machine language, instruction by instruction (statement by statement)**.
- 2) **Assembler**
The assembler converts the program written in **assembly language** into **low level language**, i.e., **the machine language, instruction by instruction**.
- 3) **Compiler**
The compiler converts the program written in **high level language** into **low level language**, i.e., **to the machine language, the program as a whole**.

Lesson 2

Introduction To Java

Two Types of Java Programs

Two types of Java programs are **Internet Applets** and **Stand Alone Applications**.

Internet applets are the programs embedded in web pages and used for internet applications.

Stand Alone Applications are the programs used in independent computers without web pages.

Source Code

The program written in High Level Languages such as **Java**, C, C++ etc. is known as **source code**.

Machine Code

The language of computer is machine language i.e., **binary (0 and 1)**. The instructions in binary form are **machine code**. It is the Low Level Language. The binary code of **A** is **01000001** and **B** is **01000010**.

Object Code / Machine Executable Code / Native Executable Code

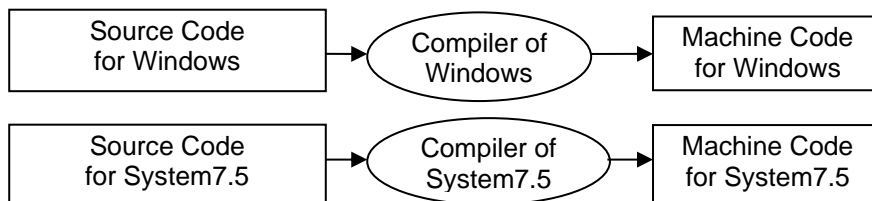
The source code converted into machine code to a **particular platform** is object code (machine executable code / native executable code). For different platforms, different object codes are produced.

Note: The **platform** generally refers to **combination of system software and hardware** like **Windows** on **Pentium IV** or **System7.5** on **PowerMacs**.

Java Compilation

Ordinary Compilation Process

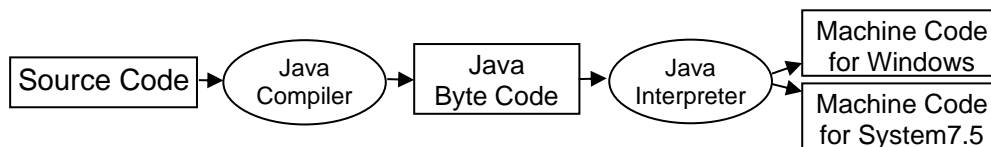
The **compiler for Windows** converts the source program directly to machine code for Windows only. The **compiler for System7.5** compiles to machine code only for **System7.5**. The compiler for a particular system software can convert only for that system software.



Java Compilation Process

Java Compiler compiles the source code to **Java Byte Code**, then this byte code is interpreted by **Java Interpreter** into **Machine Code** for a specific platform i.e., if the operating system is *Windows* then to the *machine code for Windows*, if the operating system is *System7.5* then to the *machine code for System7.5*.

Diagrammatic Representation of Java Compilation Process



Java Byte Code

Java byte code is a machine instruction i.e., an **intermediate code** which is converted from source code by **Java compiler**. The **Java byte code** is machine independent.

Java Virtual Machine (JVM)

Java interpreter is **Java Virtual Machine (JVM)**. It used to convert Java byte code to native machine executable code (object code).

Java API (Java Application Programming Interface)

Java API is a library of compiled code (small built-in programs) that can be used in our program in order to lessen our program effort.

Java Platform:

The **Java Compiler**, **JVM** combined with **Java APIs** makes **Java platform**.

JDK (Java Development Kit):

It contains **Java compiler**, **Java Virtual Machine (JVM)**, **Java API** (libraries) etc.

Lesson 3

Importance of Java

Java is Platform Independent

The program written in Java can be run on any platform that means it can be executed by any operating system and any processors; because the source code is compiled by Java Compiler to byte code which is suited to any platform and then interpreted by JVM to native executable code. So Java is platform independent.

Java is a Programming Language as well as a Platform

Java can be used to develop various application programs like MS Word, PowerPoint, Excel etc. So **Java is a programming language**. Java includes some programs like compiler, interpreter - generally which are parts of a system software. So **Java is a platform also**.

Features of Java

The following are the **features of Java**

1. **Simple:** By simple code we can create big programs.
2. **Robust:** Java can be used to develop even complex softwares.
3. **Secured:** Java offers many security features to make the programs safe and secure.
4. **Object Oriented Language:** Similar kinds problems are considered as a class and each individual is considered object of this class and their problem is solved this class definition.
5. **Platform Independent:** Java can be run on any platform.
6. **Write Once Run Anywhere (WORA):** Java programs need to be written only once. It can be run on different platforms.
7. **Built-in Graphics:** Java offers many built-in graphics features.
8. **Open Product:** Freely available to all.

Short History of Java

1. When was Java introduced, who developed it and what was its initial name?
In **1991**, **Java was developed** by **James Gosling**. Initially its name was **Oak**.
2. When was it formally announced as Java?
In **1995**, **Oak** was formally **announced** as **Java** by **Sun Micro Systems**.
3. What was its initial use and when is it incorporated to web browsers?
Initially **Oak** is used in consumer electronic devices such as **microwave oven, washing machine** etc.
In **1995** its name changed to **Java** and it is incorporated to the web browsers **Netscape Navigator**.

Lesson 4

BlueJ an IDE

IDE

An **IDE** (Integrated **D**evelopment **E**nvironment) is used to develop programs. It consists:

- 1) **Editor** – to write the programs
- 2) **Debugger** – to find errors. Bugs means errors in a program. Debug means correction of errors.
- 3) **Viewer** – to see the parts of program graphically
- 4) **Terminal window** – to input values to the program and get output from it.

Some IDEs to Develop Java Programs

DOS Editor with *DOS Prompt*, **NotePad** with *DOS Prompt* and **BlueJ** in Windows etc.

BlueJ

BlueJ is an IDE specially designed for learning Java at an introductory level.

Steps of Writing a Program in BlueJ

New Project Creation

- 1) Open **BlueJ**.
- 2) Click on **Project** menu and **New Project** option.
- 3) Type Folder name (e.g.: **Lesson1**) and click on **Create** button (or press **Enter** key).
- 4) Click on **New class...** button and type any alphabet in the **Class name:** text box (e.g.: **a**).
- 5) Right click on the icon that is appeared and left click on **Open Editor** and maximize the window.
- 6) Press **Ctrl+A** to select all contents and **type your program**.

New Class Creation in the Existing Project

Do the steps 4 to 6.

Compiling and Running the Program

- 1) Click on **Compile** button or press **Ctrl+k**.
If there is no error then there will be a message **Class compiled – no syntax error**.
If there is error the error will be shown. Correct the error and compile again.
- 2) Click on the **Project window** on the **Task bar** and right click on the **class icon** and left click on **void main()** option in the list shown.
- 3) The output is displayed in the **Terminal Window**. Click on **Options** menu and select **Clear screen at method call** option in order to avoid this output with the next one.

Close the Class and Project

Use the **close button** or **Ctrl+w**.

Open Recent Project

Click on **Project** window, point to **Open Recent** option, select the required project from the list.

Remove Unwanted Class

Right click on the class icon and click on the **Delete**.

Remove Unwanted Project

Use the Windows Explorer and delete from there.

Questions

1. What is an IDE?
2. What is meant by debugging?