**Beginnings of the Java TMProgramming Language Project**

* wide variety of network devices and embedded systems ke liye start huva tha
* Design and architecture decisions drew from a variety of languages such as Eiffel, SmallTalk, Objective C, and Cedar/Mesa.

**The Java Platform**

A *platform* is the hardware or software environment in which a program runs.

Java is software-only platform that runs on top of other hardware-based platforms.

The Java platform has two components:

* The *Java Virtual Machine*
* The *Java Application Programming Interface* (API): → Java provides built-in libraries for networking, databases, GUIs, etc.

**Different java platforms:**

* + **Java SE** = Core Java (foundation for all Java apps). basics
  + **Java EE** = Enterprise extensions (web, servers, distributed computing).
  + **Java ME** (Micro Edition) → For mobile/embedded devices (subset of Java SE).
  + **JavaFX** → For rich desktop/UI applications (replaced Swing).

**Features of java**

|  |  |
| --- | --- |
| * Simple * Object oriented * Distributed * Multithreaded * Dynamic(runtime memory allocation) | * Architecture neutral * Portable * High performance * Robust * Secure * Interpreted(line by line) |

**Architecture neutral** -> can work on various platforms

The Java virtual machine is based primarily on the POSIX interface specification ( Portable Operating System Interface).

**High perormance** :-The *automatic garbage collector* runs as a low-priority background thread, ensuring a high probability that memory is available when required,

**Distributed**: work on multiple computers or networks

**Robust**: type checking at compile and runtime

**Portable**:Java guarantees that int data types are always 32-bit, regardless of the underlying hardware.

**Naming, Compiling and Running Java Files Containing More Than One Class Definitions(scenrios in concept of the day)**

* Public class jedar define kiya hai uss naam se save karo
* One class not conatin more than one public class
* Class agar default hoga toh koi bhi naam dedo chalega

Varibales:

1. Instance variables:

* Non static variables
* This values are different to different objects.

1. Class variables:

* Static variables
* It is same for every object
* One copy exist

**Integer Literals:**

* Decimal: Base 10, whose digits consists of the numbers 0 through 9; this is the number system you use every day
* Hexadecimal: Base 16, whose digits consist of the numbers 0 through 9 and the letters A through F
* Binary: Base 2, whose digits consists of the numbers 0 and 1 (you can create binary literals in Java SE 7 and later)

Eg:

// The number 26, in decimal

int decVal = 26;

// The number 26, in hexadecimal

int hexVal = 0x1a;

// The number 26, in binary

int binVal = 0b11010;

**Floating-Point Literals:**

double d1 = 123.4;

// same value as d1, but in scientific notation /exponentail notation

double d2 = 1.234e2;

float f1 = 123.4f;

**String and char literals**

Literals of types char and String may contain any Unicode (UTF-16) characters: direct emojios and other lang ko hum likh sakte hai

Agar editior nhi likhne deta hai direct emojis wagere toh hum unicode escapes likh sakte hai .

| Escape Sequence | Meaning | Example Text | Output |
| --- | --- | --- | --- |
| \b | Backspace – deletes previous char | "Hello\bWorld" | HellWorld (W replaces o) |
| \t | Horizontal Tab – adds tab space | "Java\tProgramming" | Java Programming |
| \n | New Line – moves to next line | "Line1\nLine2" | Line1Line2 |
| \f | Form Feed – page break (rare) | "Form\fFeed" | FormFeed (symbol varies) |
| \r | Carriage Return – goes to line start | "Carriage\rReturn" | May show Returnage or Return |
| \" | Double Quote | "She said: \"Hello\"" | She said: "Hello" |
| \' | Single Quote | "It\'s Java!" | It's Java! |
| \\ | Backslash | "C:\\Program Files\\Java" | C:\Program Files\Java |

Eclipse ide outputs : outputs depends on ide to ide

| Escape Sequence | Expected | Why Your Output Differs |
| --- | --- | --- |
| \b | Erase char | Your IDE shows literal backspace (?) |
| \r | Overwrite line start | Your IDE prints the second word in a new line |
| \f | Page break | Most consoles show a symbol instead (?) |

 null literal

In java 1.7 🡪 we use underscore between digits

Increment and decrement(in eclipse mycodes basics)

* We cant write increment decrement operator for boolean
* We can write increment and decrement for charcahter it will calculate its unicode
* We cant use nested increment and decrement operator.

Packages and import

Packages 🡪 for packages to compile javac -d filename.java

To run 🡪 java packagename.classname.

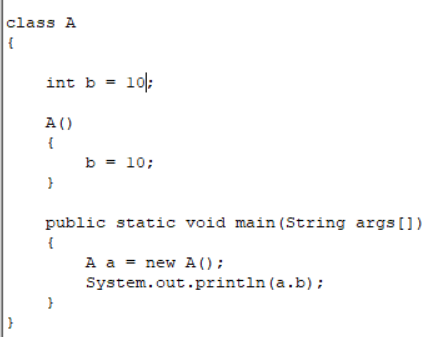
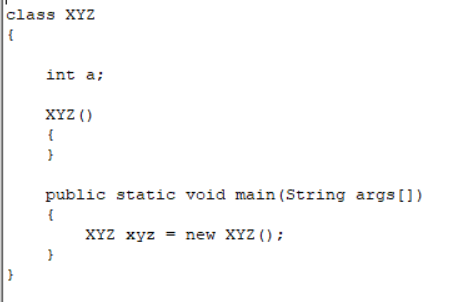
import

We will only get to import those thing in another class only if the class is public and fields public (fields mai without inheritance) else protected fields from subclass 🡪 in case of inheritance

Default Constructor:

The default constructor only intialize that value which you declare as global variable at declaration

NOTE: the default constrcutor doesn’t assign default value to the global variable it is JVM Itself give default values based on the datatype.first JVM will give default value for any object

Intialize in global var 🡪 by jvm

Intialize in global var 🡪 by default constructor

If JVM intialize default values than why constuctor is needed?

Because Java’s new keyword **requires a constructor** to be called—even if it does nothing!

Passing Information to a Method or a Constructor

Parameters:  list of variables in a method declaration.

*Arguments* : the actual values that are passed in when the method is invoked

Parameters Types:

**Note:** If you want to pass a method into a method, then use a [lambda expression](https://docs.oracle.com/javase/tutorial/java/javaOO/lambdaexpressions.html) or a [method reference](https://docs.oracle.com/javase/tutorial/java/javaOO/methodreferences.html).

Arbitrary Number of Arguments(int …a) (in array)

Shortcut for array

It is overcoming the method overloading problems

Primitives datatypes