



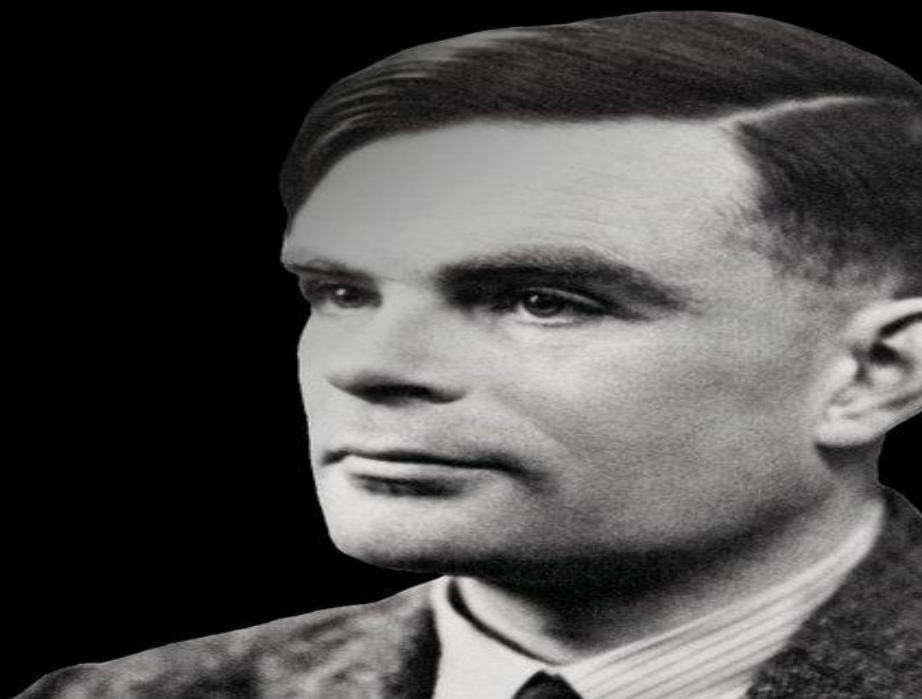
START
HACK

“

Programming is a skill best acquired
by **practice** and **example** rather
than from books.

”

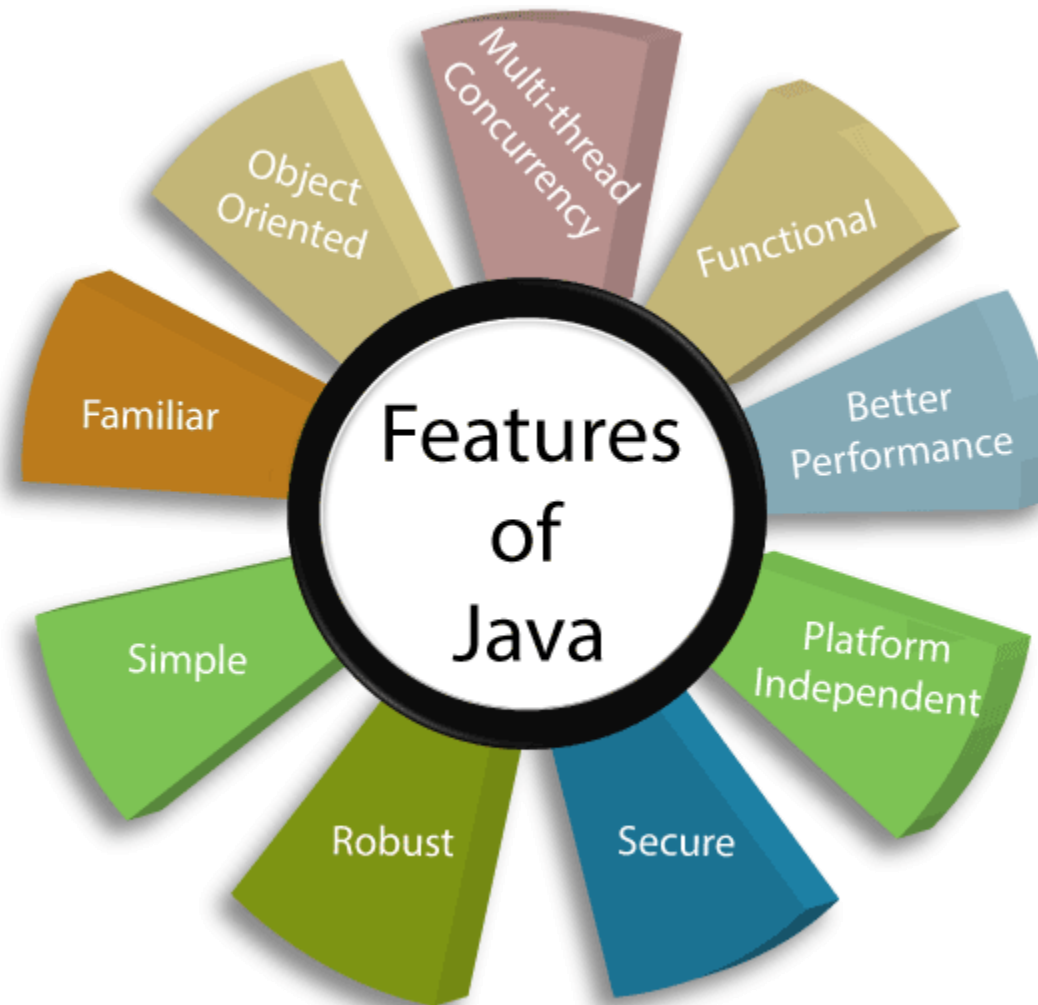
ALAN TURING
Computer Scientist



What is Java

- Java is Programming Language and Platform.
- More than 3 billion devices run on Java.
- Two types In Java – 1] Core Java 2] Advance Java





Features of Java

Simple

According to Sun, Java language is simple because:

- Syntax is based on C++ (so easier for programmers to learn it after C++).
- Removed many confusing and/or rarely-used features e.g., explicit pointers, operator overloading etc.
- No need to remove unreferenced objects because there is Automatic Garbage Collection in java.

Presentation by Alok Gupta

Platform Independent/Architecture Neutral

- ◉ The biggest strength of Java is its platform independent nature. Being platform independent means a program compiled on one machine can be executed on any machine of the world without any change. Java achieves platform independency by using the concept of BYTE code. The Java compiler never converts the source code to machine code like C/C++ compiler. Rather it converts the source code into an intermediate code called as the byte code and this byte code is further translated to machine dependent form by another layer of software called as JVM(Java Virtual Machine). Thus any platform (operating system) for which a JVM is present can execute byte code irrespective of the fact where the byte code has been generated. Due to this a java program becomes portable as well as platform independent.

Presentation by Alok Gupta

Multithreaded

- ◉ Multithreading means executing multiple portions (functions) of the same program in parallel. Just like in Microsoft word, multiple functionalities take place at same time, like typing, spell check, grammar check, auto save, auto complete, office assistant etc... Since all this is taking place inside MS Word, we can say that it is an example of multithreading. Java is one of the pioneer language to support this concept. Thus development of multithreaded application is one of the important strength of Java.

Presentation by Alok Gupta

Object Oriented

- ◉ Java strongly supports the concept of OOPs due to which it is called a Pure object oriented language. Moreover, Java strictly follows every principle of OOPs. For example we cannot develop an executable program in Java without making use of class. This indicates that Java very strictly applies the principal of Encapsulation.

Automatic Memory Management

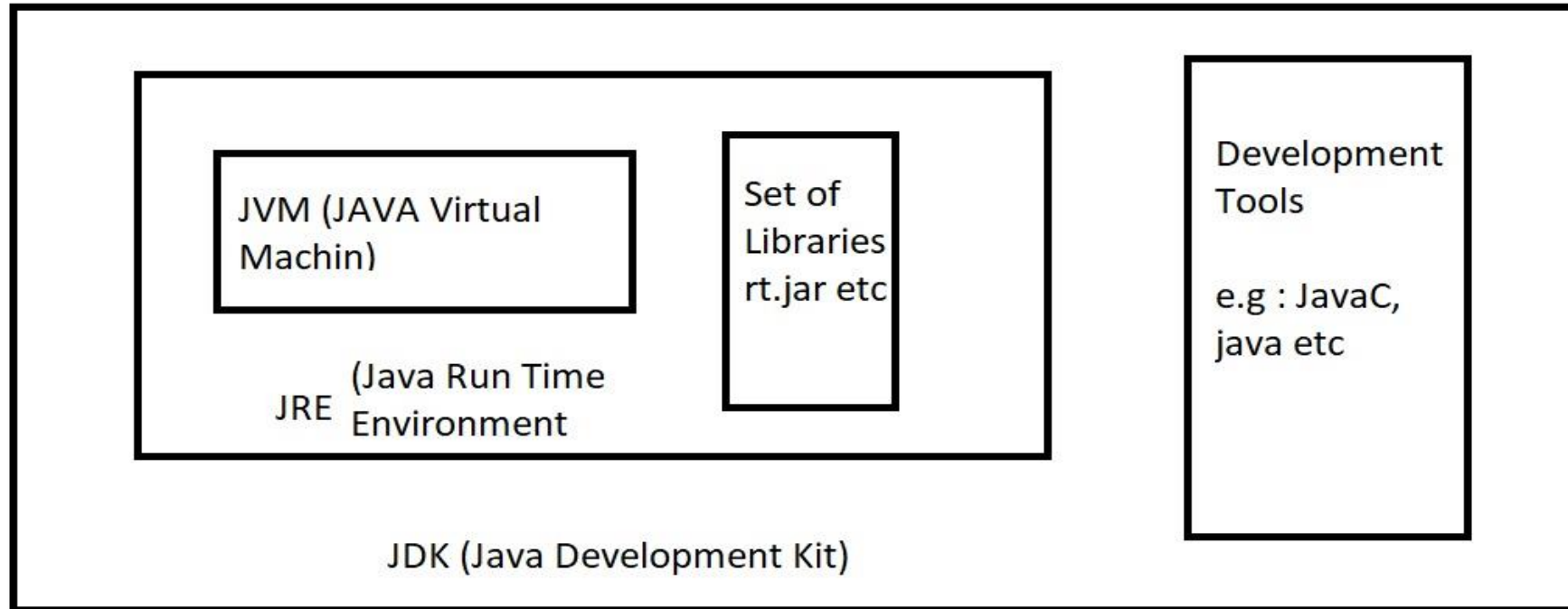
- ◉ In Java, just like C or C++ programmer can request dynamic memory block at runtime using the keyword **new**. But the deallocation of these memory blocks is automatically handled by JVM i.e. a Java programmer does not have to worry about deallocation of dynamic memory. So, runtime errors like memory leaks which were very common in C/C++ do not happen at all in Java.

Presentation by Alok Gupta

Robust (Fault Tolerance)

- ◉ In Java, any runtime error encountered by the JVM is never passed on to the underlying system rather as soon as any such error occurs, the JVM immediately terminates the program stopping it from causing any harm to the underlying system. Thus we can say that Java is a robust language which does not allow any such operation by the program which can cause any kind of harm to OS.

Basics Of Java



25. What are the differences between JVM, JRE and JDK in Java?

Criteria	JDK	JRE	JVM
Abbreviation	Java Development Kit	Java Runtime Environment	Java Virtual Machine
Definition	JDK is a complete software development kit for developing Java applications. It comprises JRE, JavaDoc, compiler, debuggers, etc.	JRE is a software package providing Java class libraries, JVM and all the required components to run the Java applications.	JVM is a platform-dependent, abstract machine comprising of 3 specifications - document describing the JVM implementation requirements, computer program meeting the JVM requirements and instance object for executing the Java byte code and provide the runtime environment for execution.
Main Purpose	JDK is mainly used for code development and execution.	JRE is mainly used for environment creation to execute the code.	JVM provides specifications for all the implementations to JRE.
Tools provided	JDK provides tools like compiler, debuggers, etc for code development	JRE provides libraries and classes required by JVM to run the program.	JVM does not include any tools, but instead, it provides the specification for implementation.
Summary	JDK = (JRE) + Development tools	JRE = (JVM) + Libraries to execute the application	JVM = Runtime environment to execute Java byte code.