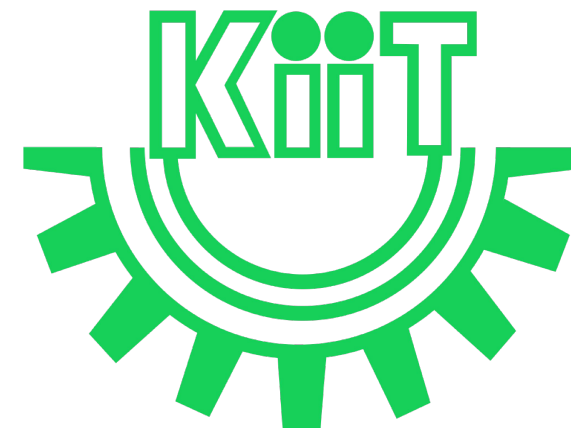


# CS20004: Object Oriented Programming using Java

## Lec-3



# In this Discussion . . .

- Java : Language Fundamentals
  - History
  - Motivation
- Object-oriented Programming concepts
  - Advantages
- Features of Java
- Java Architecture
  - JDK, JRE, and JVM
- Sample Java Program & Allied Explanation
- Java: Platform Independence, and Portability
- References



# Java Language Fundamentals: History

JAVA is a general purpose; object oriented programming language developed by James Gosling at Sun Microsystems, in 1991.

Year	Chain of Events
1989	Sun Microsystems, USA tried to develop a new language to manipulate consumer electronic devices [computers, laptops, smartphones, TV sets, tablets, game consoles, speakers and headphones, wearables, digital cameras, and VR (Virtual Reality) & AR (Augmented Reality) devices.]
1990	Three programmers from Sun Microsystems, namely, James Gosling, Patrick Naughton, and Ed. Frank tried to develop a language to manipulate consumer electronic devices
1991	Sun Microsystems developed a pure object oriented language called “Oak”
1992	Active Server Pages (ASP) was developed and its main aim is to introduce the language in internet [Active Server Pages (ASP) is Microsoft's first server-side scripting language and engine for dynamic web pages.]
1993	A team named Green Project Team, headed by James Gosling researched to launch this language on internet
1994	Sun Microsystems developed their own browser named as “Hot JAVA”
1995	Finally “Oak” was renamed as JAVA
1996	JDK 1.0 was launched
1999	Three Platforms of Java Introduced: 1. J2SE (Java 2 Standard Edition) - Used for standard applications 2. J2EE (Java 2 Enterprise Edition) - Used for Web applications 3. J2ME (Java 2 Micro Edition) - Used for mobile application, CDMA Operations, etc.

# Motivation for Java

- In-built garbage-collection facilities
- Portable facilities
- Multithreading concept
- Platform Independence
- Abundant API, Multiple Open Source Libraries

It is used by some **9 million developers** and **it runs on around 7 billion devices** worldwide according to Oracle blogs.

# Motivation for Java: Why is Java so popular? Platform independence

- The first and foremost reason is the **platform independence feature of Java**.
  - Platform independent means programmers who developed their software in one platform can execute it on any platform or operating system regardless of underlying configuration, because of **Java Virtual Machine**.
  - In other words, we can say that Java follows **WORA** rule which says Write Once Run Anywhere.

# Motivation for Java: Why is Java so popular? Multithreading

- Java also provides a multi-threading feature through which we can create multiple threads under any process which works independently and simultaneously.
  - **For Ex-** You may be having your evening snacks while reading this slide. Also, may be you're listening to a song. Thus, you are performing three different tasks in parallel.
    - Similarly, for processes to run in parallel on a computer, ***threads*** are used.
- **Multithreading** in Java gives the ability to execute code by different threads to perform tasks in parallel or as a separate task without waiting for other to complete.
- As an application, you can have computer games. You see objects in games like cars, motor bikes etc. They are just threads that run in the game application.

# Motivation for Java: Why is Java so popular?

- It uses JIT(Just In Time Compiler) that is capable of compiling only that method which is being called and not the whole program. This saves a lot of time and makes the overall development efficient.
- Due to its high performance, Java has a wide variety of applications.
- Java is one of the oldest programming languages used for the development of android applications. It is also used for scientific and web applications.

## **According to the stats published by Oracle:**

45 Billion active Java virtual machines globally  
Java is the #1 development language for microservices  
Java is used in more organizations than any other development language  
25 Billion active cloud Java virtual machines

# Motivation for Java: Why is Java so popular?

- A java program is compiled & interpreted. The source code is compiled into an intermediate **bytecode** file. *This intermediate file is interpreted by a Java interpreter*
  - Bytecode makes the java program portable across different h/w and o.s. platforms
  - Increased security due to the control of the JVM over the execution of the byte code file



# Motivation for Java: Why is Java so popular?

- Java provides implicit support to memory management. The automatic technique of freeing unused memory is called **garbage collection**
- Java has a strong exception-handling mechanism
- Java supports TCP/IP and UDP protocol families and it can be used for programming in a networked environment

# Popular Sites using Java

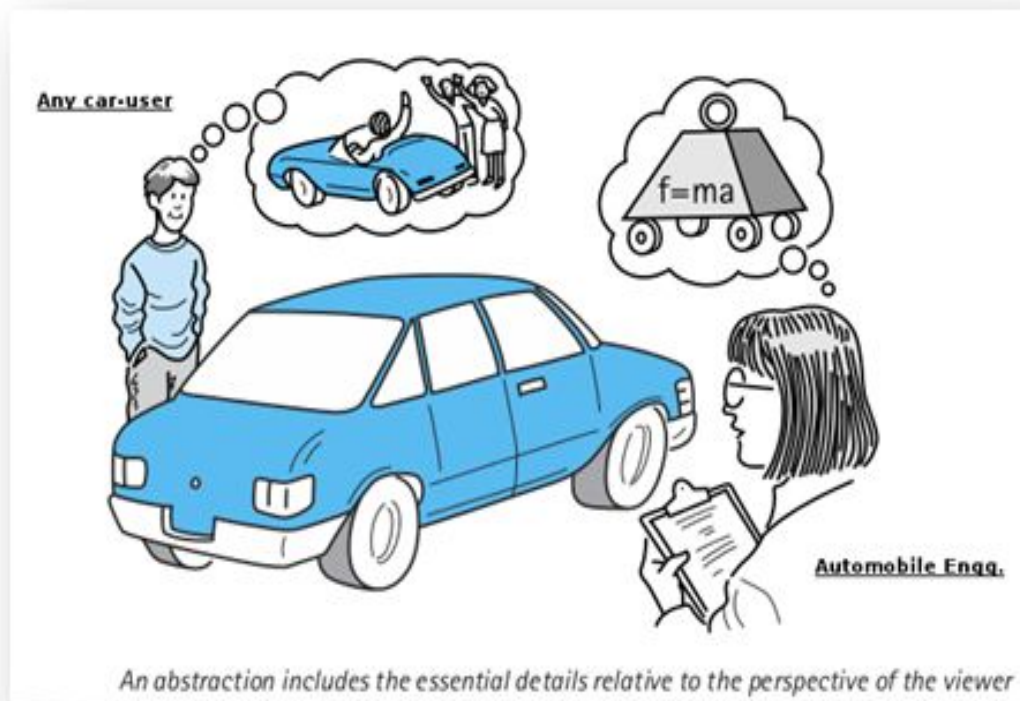
- ❖ Yahoo.com
- ❖ LinkedIn.com
- ❖ Microsoft.com
- ❖ Amazonaws.com
- ❖ 163.com
- ❖ Digicert.com
- ❖ Adobe.com
- ❖ Booking.com
- ❖ Oracle.com
- ❖ Wiley.com

# Object-oriented programming: A concepts brush-up

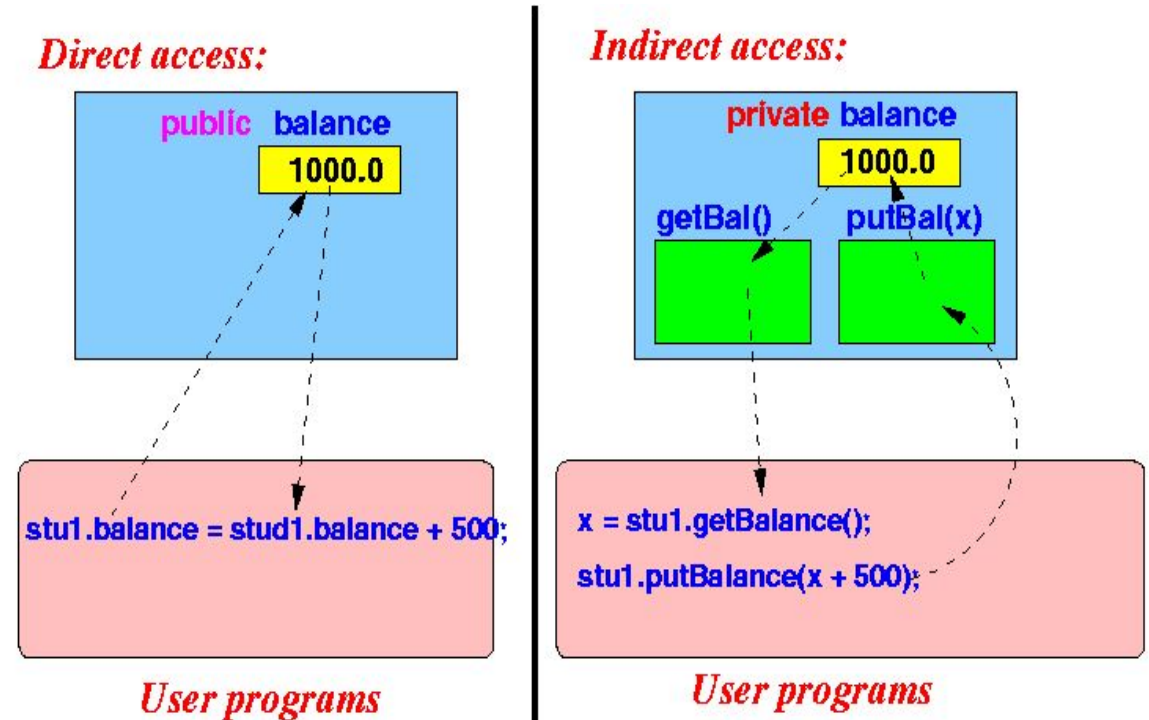
- The data is treated as the most important element and it can't flow freely around the system
- **Encapsulation**
  - Class is used as a unit to group related attributes and operations together.
  - The outside world can interact with the data stored in the variables that represent the attributes of the class only through the operations of that class.
- **Abstraction & Implementation Hiding**
  - Class is one abstract unit. In order to perform a task that involves an object of that class, a message must be sent to the object asking it to execute the respective operation
  - Implementation hiding means the manner in which data is stored and manipulated using operations of the class, is hidden from the outside

# Object-oriented programming: A concepts brush-up

- Abstraction & Implementation Hiding



Abstraction Example

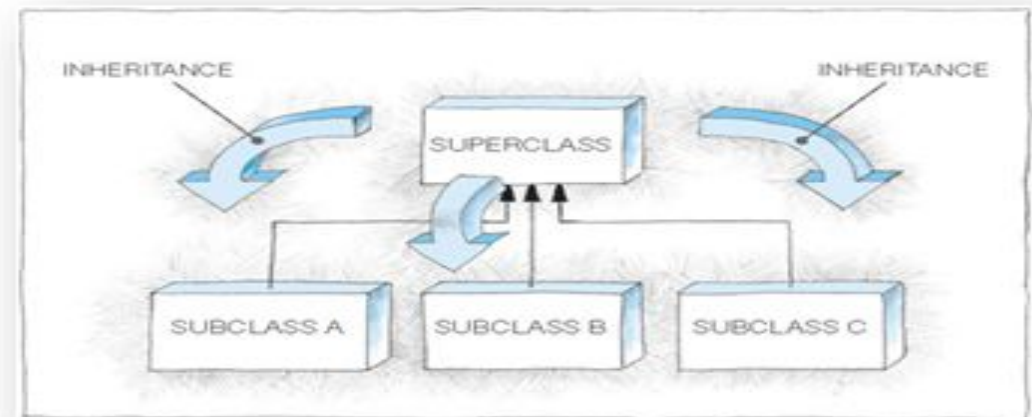


Information Hiding Example

# Object-oriented programming: A concepts brush-up

- **Inheritance, Dynamic Binding and Polymorphism**

- Inheritance is a property by which one class inherits the features of another class
- Dynamic binding or runtime binding or late binding is a technique in which the piece of code to be executed is determined only at runtime
- Polymorphism means the ability to take more than one form



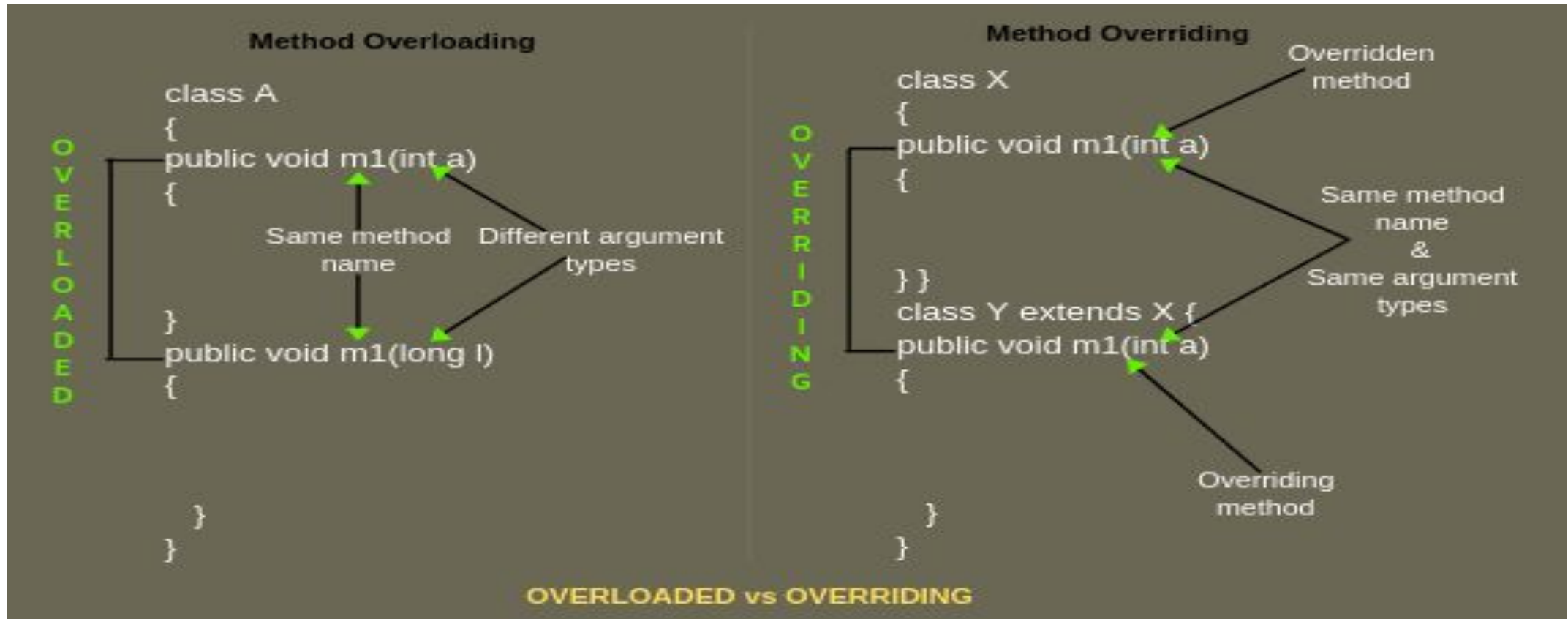
# Object-oriented programming: A concepts brush-up

- **Overriding and Overloading**

- Overriding of a method in a class is the redefinition of a method in the sub-classes of that class
- Method overloading is a term used when several methods within the same class have the same name but different signatures (signature means the number and type of parameters in the method)
- In overriding, the same method name and same signature can be used in different sub-classes of the class and defined differently in each sub-class; whereas in overloading, the same name of the method with different signatures is used multiple times in the same class

# Object-oriented programming: A concepts brush-up

- Overriding and Overloading



# Advantages of Object Oriented Programming

- It supports reusability of code
- Frameworks can be created
- It provides a clear modular structure for programs
- Easier for updating of codes in method



# Features of Java

- **Simple**

- Difficult concepts of C & C++ (covering pointers) have been omitted in JAVA. Java is designed to be easy for the professional programmer to learn and use.

- **Object Oriented**

- Java is an object oriented programming language. In order to write a program in Java, we need the aid of class or an object.
- A class is a user defined data type, i.e., a class is a model or a blueprint for creating the objects
- An object is an instance of a class.
- Java provides a clean, usable, pragmatic approach to objects, not restricted by the need for compatibility with other languages

# Features of Java

- **Robust**

- Restricts the programmer to find the mistakes early, performs compile-time (strong typing) and run-time (exception-handling) checks, manages memory automatically.
- Java programs don't crash easily due to its inbuilt exception handling and memory management features.

- **Multithreaded**

- Java supports multi-threaded programming for writing program that perform concurrent computations

- **Architecture Neutral or System Independence**

- JVM provides a platform independent environment for the execution of byte code. This byte code can be run on any machine with any processor and any operating system.

# Features of Java

- **Interpreted and high-performance**

- Java programs are compiled into an intermediate representation- byte code:
  - can be later interpreted by any JVM
  - can be also translated into the native machine code for efficiency
  - The interpreter within the JVM is slow. Thus, Java provides a Just-In-Time (JIT) Compiler to improve performance.

- **Distributed**

- Using java, we can write programs, which capture information & distribute it to the client in the network. This is possible because java supports and handles protocols like TCP/IP and UDP.

- **Dynamic**

- substantial amounts of run-time type information to verify and resolve access to objects at run-time

- **Secure**

- programs are confined to the Java execution environment and can't access other parts of the computer

# JVM

- JVM (Java Virtual Machine) is an abstract machine.
- It is called a virtual machine because it doesn't physically exist.
- It is a specification that provides a runtime environment in which Java bytecode can be executed.
- It can also run those programs which are written in other languages and compiled to Java bytecode.
- The JVM performs the following main tasks:
  - Loads code
  - Verifies code
  - Executes code
  - Provides runtime environment

# JRE

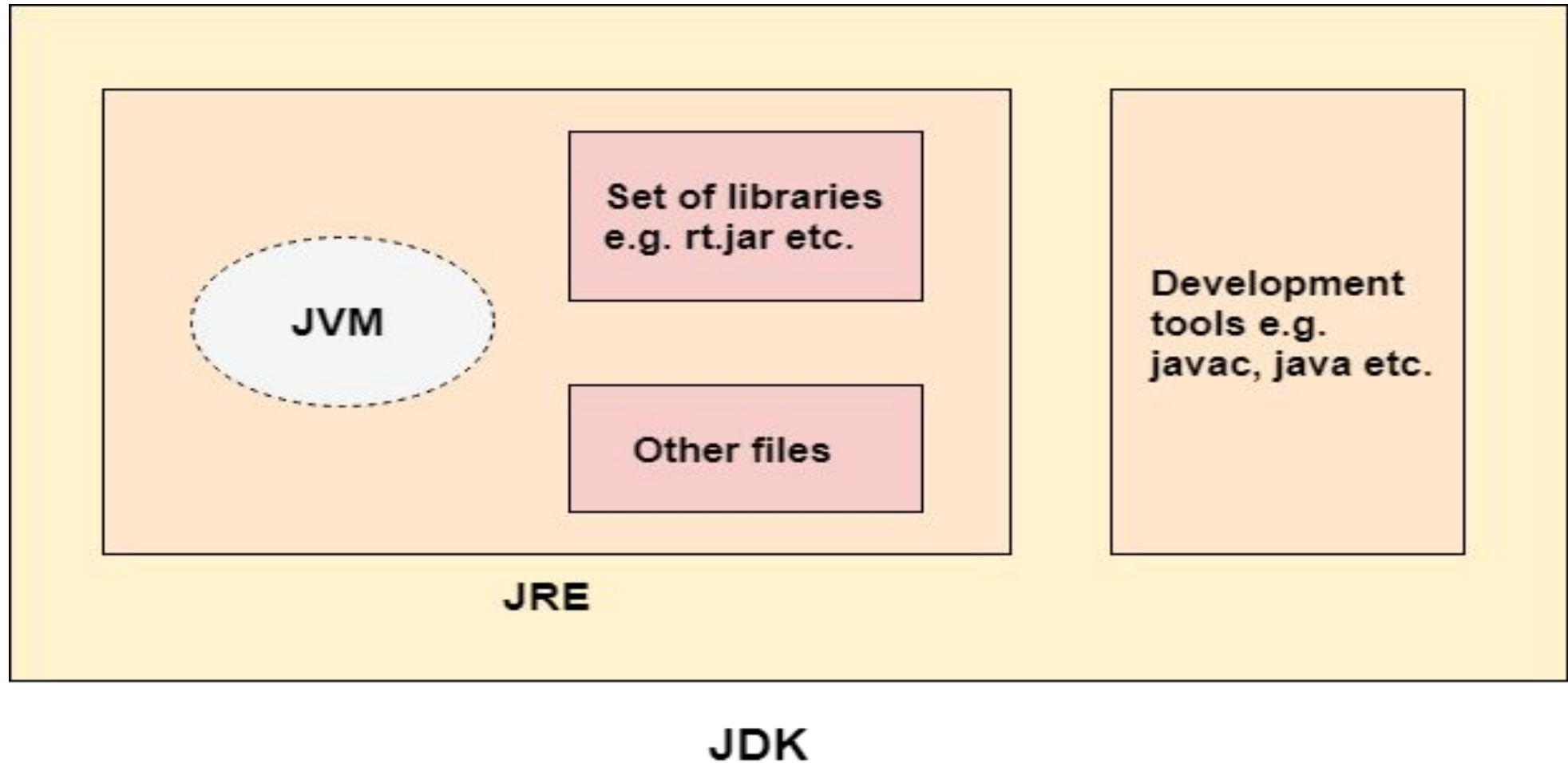
- JRE is an acronym for Java Runtime Environment.
- It is used to provide the runtime environment.
- It is the implementation of JVM. It physically exists.
- JRE consists of JVM, Java binaries, and other classes to execute any program successfully.
- JRE doesn't contain any development tools such as Java compiler, debugger, JShell, etc.
- It contains a set of libraries and other files that JVM uses at runtime.

# JDK

- JDK is an acronym for Java Development Kit.
- The Java Development Kit (JDK) is a software development environment which is used to develop Java applications and applets.
- It physically exists. It contains JRE and other development tools.
- The JDK contains a private Java Virtual Machine (JVM) and a few other resources such as an:
  - Interpreter/loader (java)
  - a compiler (javac)
  - an archiver (jar)
  - a documentation generator (Javadoc), etc.

to complete the development of a Java Application.

# JDK, JRE, and JVM



# Tools in the JDK

- JDK contains the necessary facilities for the compilation of Java programs into intermediate **byte-code** and their interpretation. It consists of various tools that can be used by the programmer to develop java programs:

<b>javac</b>	It takes as input a java source code file and produces a class file that contains the byte code
<b>java</b>	It takes as input, a class file containing the byte code and runs the program by interpreting it
<b>jdb</b>	it is a debugger that assists the developer in detecting errors
<b>javah</b>	it takes a java source code file as input and produces header files for use with native methods
<b>javap</b>	it takes the byte code file as input and produces a file that gives a description of the source code file from which the byte code file was created after compilation. It is a java disassembler
<b>appletviewer</b>	it is a tool that permits the user to execute java applets without using any java-compatible browser



# Working of the Java programming language



- We need to have a java source code otherwise we won't be able to run the program you need to save it with the **Program.java** extension.
- Secondly, we need to use a compiler so that it **compiles** the source code which in turn gives out the **java bytecode** and that needs to have a **Program.class** extension.
- The **Java bytecode** is a redesigned version of the java source codes, and this bytecode can be run anywhere irrespective of the machine on which it has been built.
- Later on, we put the **java bytecode** through the **Java Virtual Machine** which is an interpreter that reads all the statements thoroughly step by step from the java bytecode which will further convert it to the **machine-level language** so that the machine can execute the code. We get the output only after the conversion is through.

# References

1. <https://www.scaler.com/topics/div-and-span-in-html/>
2. <https://www.interviewbit.com/blog/features-of-java/>
3. <https://www.oracle.com/se/a/ocom/docs/java-strength-in-numbers.pdf>
4. <https://w3techs.com/technologies/details/pl-java#:~:text=These%20diagrams%20show%20the%20usage,side%20programming%20language%20we%20know.>
5. <https://www.javatpoint.com/advantages-and-disadvantages-of-java>
6. <https://www.geeksforgeeks.org/top-10-reasons-to-learn-java/>
7. <https://www.quora.com/What-are-some-real-life-examples-of-multi-threading-as-we-study-in-Java>
8. <https://www.javatpoint.com/difference-between-jdk-jre-and-jvm>
9. <https://www.oracle.com/java/technologies/architecture-neutral-portable-robust.html#:~:text=The%20Java%20environment%20itself%20is,which%20is%20essentially%20POSIX%2Dcompliant.>
10. <https://reachmnadeem.wordpress.com/2014/05/15/very-basic-object-oriented-concepts/>
11. <https://www.scientecheasy.com/2020/08/difference-overloading-overriding-java.html/>