Java Introduction

Java:

* Java is high-level, class-based, object-oriented programming language.
* It is a general-purpose programming language intended to let programmers write once, run anywhere(WORA).
* Java was originally developed by James Gosling at Sun Microsystems.
* It was released in May 1995.

Principles of Java:

* Object-oriented : organizes software design around data or objects.
* Robust : its ability to heandle error and unexpected situation efficiently, ensuring stable and reliable application execution.
* Secure : ensure that only legitimate Java code is executed.
* Portable : ability to run a program on different machines.
* High performance : faster than other language.
* Interpreted : translate high level program statements into assembly level programming language.
* Multithreaded:allowing developers to write programs that can perform many tasks simultaneously
* Distributed:used in distributed environments
* Dynamic and Extensible:it can dynamically extend themselves by linking in new class libraries, methods, and objects, enabling dynamic code behavior

JDK:

* It is Java Development Kit.
* It is a software development environment which is used to develop Java applications.
* It physically exists.
* It contains JRE + development tools.
* It contains a private Java Virtual Machine(JVM) and few other resources such as interpreter, a compiler, an archiver, a documentation generator etc

JRE :

* It is Java runtime environment.
* It is a set of software tools which are used for developing Java applications.
* It is used to provide the runtime environment.
* It physically exists.
* It contains a set of libraries + other files that JVM uses at runtime.

JVM :

* It is Java Virtual Machine.
* It is a specification that provide a runtime environment in which Java bytecode can be executed.
* JVM is plateform dependent.
* A specification where working of java virtual machine is specified. But implementation provider id independent to choose the algorithm. Its implementation has provided by Oracle and other companies.
* An implementation is known as JRE.
* An instance of JVM is created whenever command written on the command prompt to run the java class.

JVM’s main tasks:

* Loads code
* Verifies code
* Executes code
* Provide runtime environment

Core Java Components :

Java Standard Edition(SE) :

* Provides core functionality for general purpose use, including classes for networking, security, database access, graphical user interface(GUI) development, and XML parsing.

Java Enterprise Edition(EE) :

* An extension of Java SE, it provides an API and runtime environment for developing and running large-scale, multi-tiered, scalable and secure network applications. It includes additional libraries for enterprise features such as distributed computing and web service.

Java Micro Edition(ME) :

* A subset of Java SE, designed for developing applications for small, resource-contrained devices like mobile phones and embedded systems.

Java Version History :

JDK Alpha and Beta (1995) :

* The java Alpha and Beta was first release but they have highly unstable APIs and ABIs.
* The supplied Java web browser was named WebRunner.

JDK 1.0 (January 1996) :

* It was the first stable released version of Java. Key features included:
* Basic language constructs (classes, interfaces, etc)
* AWT(Abstract Window Toolkit) for GUI development
* Basic standard libraries

JDK 1.1 (February 1997) :

* Inner classes
* JavaBeans
* RMI(Remote Method Invocation)
* JDBC(Java Database Connectivity)
* Reflection API
* JIT (just in time) compiler

J2SE 1.2 (December 1998) :

* Swing : a new set of GUI components
* Collections Framework
* Java Plugin
* Java IDL(CORBA)
* JIT compiler enhancements
* Security enhancements

J2SE 1.3 (May 2000) :

* HotSpot JVM
* JavaSound API
* JPDA (Java Platform Debugger Architecture)
* RMI-IIOP(RMI over IIOP)

J2SE 1.4(February 2002) :

* Assertions
* Regular Expressions(java.util.regex)
* NIO(New I/O)
* Logging API(java.util.logging)
* XML Parsing(JAXP)
* Java Web Start

J2SE 5.0 (September 2004) :

* Generics
* Metadata (Annotations)
* Autoboxing/ Unboxing
* Enumerated types
* Varargs
* Enhanced for loop
* Concurrency untilities(java.util.concurrent)

Java SE 6 (December 2006) :

* Scripting language support(JSR 223)
* Compiler API(JSR 199)
* Pluggable annotations processing (JSR 269)
* Web Services(JAX-WS)
* Improvements to Swing

Java SE 7 (July 2011) :

* Diamond oprator
* Try with resources
* String in switch
* Binary literals, underscores in numeric literals
* Fork/Join Framework
* NIO.2(New File I/O)
* JVM support for dynamic languages(InvokeDynamic)

Java SE 8(March 2014) :

* Lambda Expressions
* Stream API
* Optional Class
* Default and static methods in interfaces
* Nashorn Javascript Engine
* New Date and Time API(Java.time)
* Concurrent Accumulators

Java SE 9(September 2017) :

* Module System (Project Jigsaw)
* Jshell (REPL)
* Improved Javdoc
* Stream API improvements
* HTTP/2 Client(incubator module)
* Multi-release JARs

Java SE 10 (March 2018) :

* Local-variable type inference(var)
* Application Class-Data Sharing
* Thread-local handshakes
* Garbage-collector interface

Java SE 11 (September 2018) :

* Long-term support(LTS) release
* New HTTP client
* Nest-based access control
* Dynamic class-file constants
* Epsilon garbage collecto

Java SE 12 (March 2019) :

* Switch expressions(preview)
* JVM constants API
* Shenandoah garbage collector

Java SE 13 (September 2019) :

* Text blocks(preview)
* Switch expressions(second preview)
* Reimplement the lagacy socket API

Java SE 14 (March 2020) :

* Pattern matching for instanceof(preview)
* Packaging tool(incubator)
* Records(preview)
* Switch expressions(Standard)

Java SE 15 (September 2020) :

* Edwards-Curve Digital Signature algorithm(EdDSA)
* Sealed classes(preview)
* Hidden classes
* Text blocks

Java SE 16 (March 2021) :

* Vector API(incubator)
* Records
* ZGC: Concurrent Thread-Stack processing

Java SE 17 (September 2021) :

* Long-term support(LTS) release
* Sealed Classes
* Enhanced Pseudo-Random Number Generators
* New macOS Rendering Pipeline

Java SE 18 (March 2022) :

* UTF-8 by Default
* Code Snippets in Java API Documentation
* Internet-Address Resolution SPI

Java SE 19 (September 2022) :

* Record Patterns(Preview)
* Code Snippets in Java API Documentation
* Virtual Threads(Preview)

Java SE 20 (March 2023) :

* Record Patterns(Second Preview)
* Pattern Matching for switch(Fourth Preview)
* Foreign Function & Memory API (Second Preview)

Java SE 21 (September 2023) :

* Long term support(LTS) release
* Pattern matching for switch
* Foreign function & Memory API
* Unnamed Patterns and Variables(Preview)

Data flow diagram :

Find the greatest number among the 2 numbers :

Start

int a,b;

N

Y

a>b?

b

a

end

Check whether the input number is odd or even:

Start

Int n;

N

Y

n%2==0

odd

even

end

Print the input number 5 times:

Start

int n;

int i=0;

i++

N

Y

i < 5 ?

end

Print n

Print sum of 1 to 10 numbers:

Start

int sum=0;

int i=1;

i++

N

Y

i <= 10 ?

Print sum

sum+=i

end

Print the first 5 fibonacci numbers:

Start

int a=0,b=1,i=1;

int next;

i++

N

Y

i <= 5 ?

end

print a

next=a;

a=b;

b=next;