**Basic Of Java**

Q)What is Java?

**Java** is a [high-level](https://en.wikipedia.org/wiki/High-level_programming_language), [class-based](https://en.wikipedia.org/wiki/Class-based_programming), [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) [programming language](https://en.wikipedia.org/wiki/Programming_language) that is designed to have as few implementation [dependencies](https://en.wikipedia.org/wiki/Dependency_(computer_science)) as possible. It is a [general-purpose](https://en.wikipedia.org/wiki/General-purpose_language) programming language intended to let [programmers](https://en.wikipedia.org/wiki/Programmer) *write once, run anywhere* ([WORA](https://en.wikipedia.org/wiki/Write_once,_run_anywhere)),[[16]](https://en.wikipedia.org/wiki/Java_(programming_language)#cite_note-16) meaning that [compiled](https://en.wikipedia.org/wiki/Compiler) Java code can run on all platforms that support Java without the need to recompile.[[17]](https://en.wikipedia.org/wiki/Java_(programming_language)#cite_note-design_goals-17) Java applications are typically compiled to [bytecode](https://en.wikipedia.org/wiki/Java_bytecode" \o "Java bytecode) that can run on any [Java virtual machine](https://en.wikipedia.org/wiki/Java_virtual_machine) (JVM) regardless of the underlying [computer architecture](https://en.wikipedia.org/wiki/Computer_architecture). The [syntax](https://en.wikipedia.org/wiki/Syntax_(programming_languages)) of Java is similar to [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [C++](https://en.wikipedia.org/wiki/C%2B%2B), but has fewer [low-level](https://en.wikipedia.org/wiki/Low-level_programming_language) facilities than either of them. The Java runtime provides dynamic capabilities (such as [reflection](https://en.wikipedia.org/wiki/Reflective_programming) and runtime code modification) that are typically not available in traditional compiled languages.

Java gained popularity shortly after its release, and has been a very popular programming language since then.[[18]](https://en.wikipedia.org/wiki/Java_(programming_language)#cite_note-18) Java was the third most popular programming language in 2022 according to [GitHub](https://en.wikipedia.org/wiki/GitHub" \o "GitHub)[[19]](https://en.wikipedia.org/wiki/Java_(programming_language)#cite_note-19) and it is ranked fourth on [TIOBE index](https://en.wikipedia.org/wiki/TIOBE_index) as of January 2024.[[20]](https://en.wikipedia.org/wiki/Java_(programming_language)#cite_note-20) Although still widely popular, there has been a gradual decline in use of Java in recent years with [other languages using JVM](https://en.wikipedia.org/wiki/List_of_JVM_languages) gaining popularity.[[21]](https://en.wikipedia.org/wiki/Java_(programming_language)#cite_note-:0-21)

Java was originally developed by [James Gosling](https://en.wikipedia.org/wiki/James_Gosling) at [Sun Microsystems](https://en.wikipedia.org/wiki/Sun_Microsystems). It was released in May 1995 as a core component of Sun's [Java platform](https://en.wikipedia.org/wiki/Java_(software_platform)). The original and [reference implementation](https://en.wikipedia.org/wiki/Reference_implementation) Java [compilers](https://en.wikipedia.org/wiki/Compiler), virtual machines, and [class libraries](https://en.wikipedia.org/wiki/Library_(computing)) were originally released by Sun under [proprietary licenses](https://en.wikipedia.org/wiki/Proprietary_license). As of May 2007, in compliance with the specifications of the [Java Community Process](https://en.wikipedia.org/wiki/Java_Community_Process), Sun had [relicensed](https://en.wikipedia.org/wiki/Software_relicensing) most of its Java technologies under the [GPL-2.0-only](https://en.wikipedia.org/wiki/GNU_General_Public_License) license. [Oracle](https://en.wikipedia.org/wiki/Oracle_Corporation) offers its own [HotSpot](https://en.wikipedia.org/wiki/HotSpot_(virtual_machine)" \o "HotSpot (virtual machine)) Java Virtual Machine, however the official [reference implementation](https://en.wikipedia.org/wiki/Reference_implementation) is the [OpenJDK](https://en.wikipedia.org/wiki/OpenJDK" \o "OpenJDK) JVM which is free open-source software and used by most developers and is the default JVM for almost all Linux distributions.

As of September 2023, [Java 21](https://en.wikipedia.org/wiki/Java_version_history) is the latest version, which is also a [long-term support](https://en.wikipedia.org/wiki/Long-term_support) (LTS) version. Java 8, 11, and 17 are previous LTS versions still officially supported.

Q)Who invented java in which year?

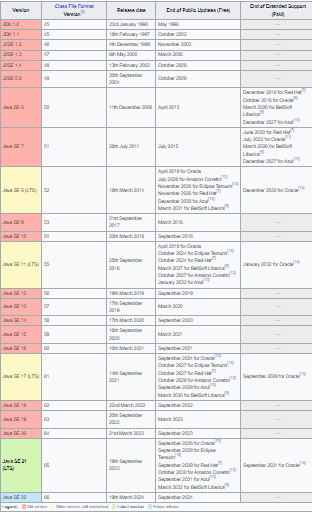
1) [**James Gosling**](https://www.javatpoint.com/james-gosling-father-of-java)**, Mike Sheridan**, and **Patrick Naughton** initiated the Java language project in June 1991. The small team of sun engineers called **Green Team**.

2) Initially it was designed for small, [embedded systems](https://www.javatpoint.com/embedded-system-tutorial) in electronic appliances like set-top boxes.

3) Firstly, it was called **"Greentalk"** by James Gosling, and the file extension was .gt.

4) After that, it was called **Oak** and was developed as a part of the Green project.

Q)What are the version/flavour of java?



Q)What do you mean by stand alone application?

A standalone application, also known as a desktop application is a software program designed in such a way that to run this software program, users don’t need an internet connection or any server access. Web-based applications need an internet connection, servers, and any additional resources to run but standalone applications do not require any additional resources such as an internet connection, server, etc.

Q)What do you mean by Web application?

A web application is software that runs in your web browser. Businesses have to exchange information and deliver services remotely. They use web applications to connect with customers conveniently and securely. The most common website features like shopping carts, product search and filtering, instant messaging, and social media newsfeeds are web applications in their design. They allow you to access complex functionality without installing or configuring software.

Q)What is the old name of java?

The language was initially called Oak after an oak tree that stood outside Gosling's office. Later the project went by the name Green and was finally renamed Java, from Java coffee, a type of coffee from Indonesia.

Q)What are the features of Java?

The primary objective of [Java programming](https://www.javatpoint.com/java-tutorial) language creation was to make it portable, simple and secure programming language. Apart from this, there are also some excellent features which play an important role in the popularity of this language. The features of Java are also known as Java buzzwords.

A list of the most important features of the Java language is given below.



1. [Simple](https://www.javatpoint.com/features-of-java#Simple)
2. [Object-Oriented](https://www.javatpoint.com/features-of-java#Object-Oriented)
3. [Portable](https://www.javatpoint.com/features-of-java#Portable)
4. [Platform independent](https://www.javatpoint.com/features-of-java#Platform-independent)
5. [Secured](https://www.javatpoint.com/features-of-java#Secured)
6. [Robust](https://www.javatpoint.com/features-of-java#Robust)
7. [Architecture neutral](https://www.javatpoint.com/features-of-java#Architecture-neutral)
8. [Interpreted](https://www.javatpoint.com/features-of-java#Interpreted)
9. [High Performance](https://www.javatpoint.com/features-of-java#High-Performance)
10. [Multithreaded](https://www.javatpoint.com/features-of-java#Multithreaded)
11. [Distributed](https://www.javatpoint.com/features-of-java#Distributed)
12. [Dynamic](https://www.javatpoint.com/features-of-java#Dynamic)

Q)What is platform?

A platform is a group of technologies that are used as a base upon which other applications, processes or technologies are developed.

In personal computing, a platform is the basic hardware (computer) and software (operating system) on which software applications can be run. This environment constitutes the basic foundation upon which any application or software is supported and/or developed.

Computers use specific central processing units (CPUs) that are designed to run specific machine language code. In order for the computer to run software applications, the applications must be in that CPU’s binary-coded machine language.

Thus, historically, application programs written for one platform would not work on a different platform.

Q)What do you mean by platform dependent?

Platform dependent typically refers to applications that run under only one operating system in one series of computers (one operating environment); for example, Windows running on x86 hardware or Solaris running on SPARC hardware. Sometimes, it means the same as "hardware dependent" or "machine dependent" and refers to applications that run in only one hardware series with the operating system not being relevant.

In contrast, "platform independent" means that the application can run in different operating environments. Applications written in Java are a prime example. See [platform](https://www.pcmag.com/encyclopedia/term/platform) and [environment](https://www.pcmag.com/encyclopedia/term/environment).

Q)What do you mean by platform independent?

Platform independent refers to software that runs on a variety of operating systems or hardware platforms. It is the opposite of platform dependent, which refers to software that is only to run on one operating system or hardware platform. An example of platform-independent software is Microsoft Windows, which also runs on Mac systems. Because platform-independent software works across different computing environments, businesses don't have to maintain multiple versions of a type of program.

Q)How Java achive Platform independency?

Java is platform-independent because it is compiled to a bytecode that can be run on any device that has a Java Virtual Machine (JVM). This means that you can write a Java program on one platform (such as Windows) and then run it on a different platform (such as macOS or Linux) without making any changes to the code. The JVM acts as an interpreter for the Java bytecode, translating it into instructions that the host machine can understand and execute. This means that the same Java program can run on any device that has a JVM, making it a truly “write once, run anywhere” language.

Q)Where will we use Java?

Java is a widely-used programming language for coding web applications. It has been a popular choice among developers for over two decades, with millions of Java applications in use today. Java is a multi-platform, object-oriented, and network-centric language that can be used as a platform in itself. It is a fast, secure, reliable programming language for coding everything from mobile apps and enterprise software to big data applications and server-side technologies.

What is Java programming language used for?

**What is Java programming language used for?**

Because Java is a free-to-use and a versatile language, it builds localized and distributed software. Some common uses of Java include:

**1.     Game Development**

Many popular mobile, computer, and video games are built in Java. Even modern games that integrate advanced technology like machine learning or virtual reality are built with Java technology.

**2.     Cloud computing**

Java is often referred to as WORA – Write Once and Run Anywhere, making it perfect for decentralized cloud-based applications. Cloud providers choose Java language to run programs on a wide range of underlying platforms.

**3.     Big Data**

Java is used for data processing engines that can work with complex data sets and massive amounts of real-time data.

**4.     Artificial Intelligence**

Java is a powerhouse of machine learning libraries. Its stability and speed make it perfect for artificial intelligence application development like natural language processing and deep learning.

**5.     Internet of Things**

Java has been used to program sensors and hardware in edge devices that can connect independently to the internet.

Q)What is the work of java compiler?

A Java compiler is a program that takes the text file work of a developer and [compiles](https://www.techtarget.com/whatis/definition/compiler) it into a platform-independent [Java](https://www.theserverside.com/definition/Java) file. Java compilers include the Java Programming Language Compiler (javac), the [GNU](https://www.techtarget.com/searchdatacenter/definition/GNU-General-Public-License-GNU-GPL-or-simply-GPL) Compiler for Java (GCJ), the [Eclipse](https://www.techtarget.com/searchapparchitecture/definition/Eclipse-Eclipse-Foundation) Compiler for Java (ECJ), and [Jikes](https://www.theserverside.com/definition/Jikes).

Programmers typically write language statements in a given programming language one line at a time using a code editor or an integrated development environment ([IDE](https://www.techtarget.com/searchsoftwarequality/definition/integrated-development-environment)). The resulting file contains what are called the source statements. The programmer then runs a compiler for the appropriate language, specifying the name of the file that contains the source statements.

Java and Java programmers are no exception. The Java compiler accepts high-level Java [source code](https://www.techtarget.com/searchapparchitecture/definition/source-code) and converts it into [bytecode](https://www.techtarget.com/whatis/definition/bytecode) that can be understood by a Java Virtual Machine ([JVM](https://www.theserverside.com/definition/Java-virtual-machine-JVM)) in order to produce the desired results.

At run time, the compiler parses (analyzes) all the language statements syntactically and then, in one or more successive stages or "passes," builds the output [code](https://www.techtarget.com/whatis/definition/code), making sure that statements referring to other statements are referred to correctly in the final code.

Generally, Java compilers are run and pointed to a programmer's code in a text file to produce a [class](https://www.techtarget.com/whatis/definition/class) file for use by the JVM on different [platforms](https://www.techtarget.com/searchitoperations/definition/platform). Jikes, for example, is an [open source](https://www.techtarget.com/whatis/definition/open-source) compiler that works in this way, and so does the primary compiler included in the Java Development Kit ([JDK](https://www.theserverside.com/definition/Java-Development-Kit-JDK)) called *javac.*This compiler, which is written in Java, reads class and interface definitions written in Java and like Jikes, converts them into bytecode class files.

To run the Java compiler, the programmer must run the Javac.exe command from the command prompt. The compiler, like Java is [platform-independent](https://www.theserverside.com/answer/Why-is-Java-platform-independent), meaning it can compile code and then run it on any operating system ([OS](https://www.techtarget.com/whatis/definition/operating-system-OS)). However, it is language-specific, so it cannot be used to compile and convert source code written in other languages, like [Python](https://www.techtarget.com/whatis/definition/Python), [C++](https://www.techtarget.com/searchdatamanagement/definition/C), etc.

## What is the purpose of a Java compiler?

The main purpose of a Java compiler (or a compiler in any programming language) is to translate the high-level Java source code into a [machine code](https://www.techtarget.com/whatis/definition/machine-code-machine-language) file consisting of machine-readable 0s and 1s, and then execute the file.

Compilation is essential because the machine cannot understand a human-readable language like Java. By translating human-readable code into machine-readable language, the Java compiler ensures that the code runs and produces the expected output.

A Java compiler also functions as Java's error detection mechanism. Once it is activated, it checks for syntax errors and generates a list of all detected errors. It does not generate [object code](https://www.techtarget.com/whatis/definition/object-code) unless the programmer rectifies the errors. The compiler can also add additional code to the program if required.

Q)What is the work of java virtual machine?

A **Java virtual machine** (**JVM**) is a [virtual machine](https://en.wikipedia.org/wiki/Virtual_machine) that enables a computer to run [Java](https://en.wikipedia.org/wiki/Java_(software_platform)) programs as well as programs written in [other languages](https://en.wikipedia.org/wiki/List_of_JVM_languages) that are also compiled to [Java bytecode](https://en.wikipedia.org/wiki/Java_bytecode). The JVM is detailed by a [specification](https://en.wikipedia.org/wiki/Specification_(technical_standard)) that formally describes what is required in a JVM implementation. Having a specification ensures interoperability of Java programs across different implementations so that program authors using the [Java Development Kit](https://en.wikipedia.org/wiki/Java_Development_Kit) (JDK) need not worry about idiosyncrasies of the underlying hardware platform.

The JVM [reference implementation](https://en.wikipedia.org/wiki/Reference_implementation) is developed by the [OpenJDK](https://en.wikipedia.org/wiki/OpenJDK" \o "OpenJDK) project as [open source](https://en.wikipedia.org/wiki/Open-source_software) code and includes a [JIT compiler](https://en.wikipedia.org/wiki/JIT_compiler) called [HotSpot](https://en.wikipedia.org/wiki/HotSpot_(virtual_machine)" \o "HotSpot (virtual machine)). The commercially supported Java releases available from [Oracle](https://en.wikipedia.org/wiki/Oracle_Corporation) are based on the OpenJDK runtime. Eclipse [OpenJ9](https://en.wikipedia.org/wiki/OpenJ9) is another open source JVM for OpenJDK.

## JVM specification[[edit](https://en.wikipedia.org/w/index.php?title=Java_virtual_machine&action=edit&section=1" \o "Edit section: JVM specification)]

The Java virtual machine is an abstract (virtual) computer defined by a specification. It is a part of the Java runtime environment. The [garbage collection](https://en.wikipedia.org/wiki/Garbage_collection_(computer_science)) algorithm used and any internal optimization of the Java virtual machine instructions (their translation into [machine code](https://en.wikipedia.org/wiki/Machine_code)) are not specified. The main reason for this omission is to not unnecessarily constrain implementers. Any Java application can be run only inside some concrete implementation of the abstract specification of the Java virtual machine.[[2]](https://en.wikipedia.org/wiki/Java_virtual_machine#cite_note-2)

Starting with [Java Platform, Standard Edition](https://en.wikipedia.org/wiki/Java_Platform,_Standard_Edition) (J2SE) 5.0, changes to the JVM specification have been developed under the [Java Community Process](https://en.wikipedia.org/wiki/Java_Community_Process) as JSR 924.[[3]](https://en.wikipedia.org/wiki/Java_virtual_machine#cite_note-3) As of 2006, changes to the specification to support changes proposed to the [class file format](https://en.wikipedia.org/wiki/Class_(file_format)) (JSR 202)[[4]](https://en.wikipedia.org/wiki/Java_virtual_machine" \l "cite_note-4) are being done as a maintenance release of JSR 924. The specification for the JVM was published as the *blue book*,[[5]](https://en.wikipedia.org/wiki/Java_virtual_machine" \l "cite_note-5) whose preface states:

[We](https://en.wikipedia.org/wiki/Sun_Microsystems) intend that this specification should sufficiently document the Java Virtual Machine to make possible compatible clean-room implementations. Oracle provides tests that verify the proper operation of implementations of the Java Virtual Machine.

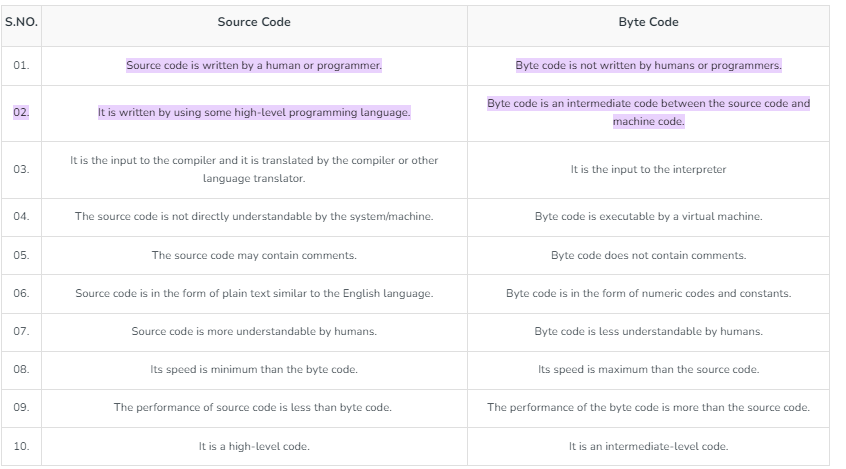
One of Oracle's JVMs is named HotSpot; the other, inherited from [BEA Systems](https://en.wikipedia.org/wiki/BEA_Systems), is [JRockit](https://en.wikipedia.org/wiki/JRockit" \o "JRockit). Oracle owns the Java trademark and may allow its use to certify implementation suites as fully compatible with Oracle's specification.

Q)Is JVM platform independent?

* In the case of Java, **it is the magic of Bytecode that makes it platform-independent**.
* This adds to an important feature in the JAVA language termed **portability**. Every system has its own JVM which gets installed automatically when the JDK software is installed. For every operating system separate JVM is available which is capable to read the .class file or byte code.
* An important point to be noted is that while **JAVA is a platform-independent language, the JVM is platform-dependent.** Different JVM is designed for different OS and byte code is able to run on different OS.

***Note:****As JVM is not platform-independent because of which Java is not considered completely platform-independent.*

Q)What is source code and byte code?



Q)What is JDK?

We can define the Java Development Kit as a software development environment responsible for creating a run-time environment for the Java source code to run.

JDK in Java is an abbreviation for Java Development Kit. It is a bundle of software development tools and supporting libraries combined with the Java Runtime Environment (JRE) and Java Virtual Machine (JVM).

Programmers familiar with Java might have the following common question.

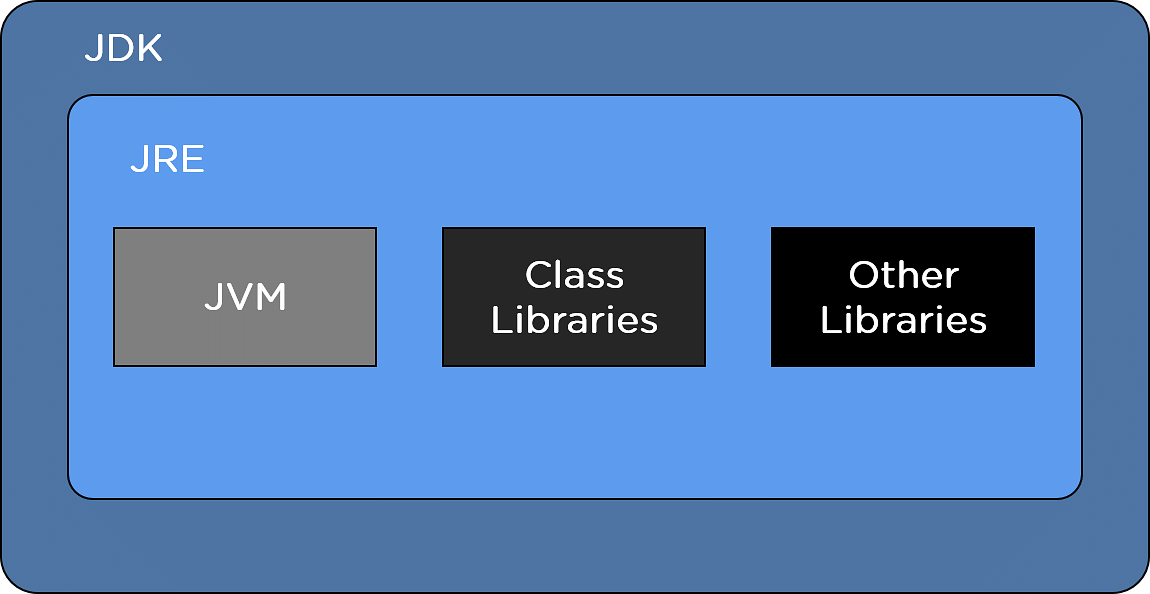
To run Java Programs, just the Java Run-time Environment would be enough, but why the complete JDK?

The answer is comparatively simple. Unlike typical Java programs, the real-time Java applications require complex software toolkits and libraries available only in JDK.

In the next segment, we will learn the architecture of JDK in Java.

## The Architecture of JDK in Java

The architecture of JDK in Java includes the following modules as described in the image below.



Q)What is JRE?

The Java Runtime Environment (JRE) is software that Java programs require to run correctly. [Java](https://aws.amazon.com/what-is/java/) is a computer language that powers many current web and mobile applications. The JRE is the underlying technology that communicates between the Java program and the operating system. It acts as a translator and facilitator, providing all the resources so that once you write Java software, it runs on any operating system without further modifications.

Q)What is JVM?

JVM stands for Java Virtual Machine. It provides a runtime environment for driving Java applications or code. JVM is an abstract machine that converts the Java bytecode into a machine language. It is also capable of running the programs written by programmers in other languages (compiled to the Java bytecode). The JVM is also known as a virtual machine as it does not exist physically.

JVM is essentially a part of the JRE (Java Run Environment). You cannot separately download and install it. You first need to install the JRE to install the JVM. It is available for many software and hardware platforms. In various distinct programming languages, the compiler functions to produce machine code for specific systems. However, only the Java compiler produces code for a virtual machine- also known as JVM.

All three, JDK, JRE, and JVM, are dependent. It is because each Operating System’s (OS) condition is different from one another. But Java is independent of the platform. The JVM has three notions: *implementation*, *instance,*and *specification*.

JVM primarily performs the following tasks:

* Provides runtime environment
* Verifies code
* Loads code
* Executes code

Q)What is JIT?

The Just-In-Time (JIT) compiler is a component of the runtime environment that improves the performance of Java™ applications by compiling bytecodes to native machine code at run time.

Java programs consists of classes, which contain platform-neutral bytecodes that can be interpreted by a JVM on many different computer architectures. At run time, the JVM loads the class files, determines the semantics of each individual bytecode, and performs the appropriate computation. The additional processor and memory usage during interpretation means that a Java application performs more slowly than a native application. The JIT compiler helps improve the performance of Java programs by compiling bytecodes into native machine code at run time.

The JIT compiler is enabled by default. When a method has been compiled, the JVM calls the compiled code of that method directly instead of interpreting it. Theoretically, if compilation did not require processor time and memory usage, compiling every method could allow the speed of the Java program to match that of a native application.

JIT compilation does require processor time and memory usage. When the JVM first starts up, thousands of methods are called. Compiling all of these methods can significantly affect startup time, even if the program eventually achieves very good peak performance.

In practice, methods are not compiled the first time they are called. For each method, the JVM maintains an invocation count, which starts at a predefined compilation threshold value and is decremented every time the method is called. When the invocation count reaches zero, a just-in-time compilation for the method is triggered. Therefore, often-used methods are compiled soon after the JVM has started, and less-used methods are compiled much later, or not at all. The JIT compilation threshold helps the JVM start quickly and still have improved performance. The threshold value was selected to obtain an optimal balance between startup times and long-term performance.

The JIT compiler can compile a method at different optimization levels: **cold**, **warm**, **hot**, **veryHot**, or **scorching** (see **optlevel** in [-Xjit](https://www.ibm.com/docs/en/SSYKE2_8.0.0/openj9/xjit/index.html)). Higher optimization levels are expected to provide better performance, but they also have a higher compilation cost in terms of CPU and memory. The initial or default optimization level for a method is **warm**, but sometimes the JIT heuristics downgrade the optimization level to **cold** to improve startup time.

A method can be re-compiled to a higher optimization level through different mechanisms. One of these mechanisms is sampling: the JIT compiler maintains a dedicated sampling thread which wakes up periodically and determines which Java methods appear more often at the top of the stack. Such methods are deemed to be more important for performance, and they are candidates for being re-optimized at the higher levels of **hot**, **veryHot**, or **scorching**.

You can disable the JIT compiler, in which case the entire Java program will be interpreted. Disabling the JIT compiler is not recommended except to diagnose or work around JIT compilation problems.

* [**How the JIT compiler optimizes code**](https://www.ibm.com/docs/en/SSYKE2_8.0.0/com.ibm.java.vm.80.doc/docs/jit_optimize.html)  
  When a method is chosen for compilation, the JVM feeds its bytecodes to the Just-In-Time compiler (JIT). The JIT needs to understand the semantics and syntax of the bytecodes before it can compile the method correctly.
* [**Frequently asked questions about the JIT compiler**](https://www.ibm.com/docs/en/SSYKE2_8.0.0/com.ibm.java.vm.80.doc/docs/jit_faq.html)  
  Answers on subjects such as disabling the JIT compiler, use of alternative JIT compilers, control of JIT compilation and dynamic control of the JIT compiler.
* [**Enabling and disabling the JIT compiler**](https://www.ibm.com/docs/en/SSYKE2_8.0.0/com.ibm.java.vm.80.doc/docs/jit.html)  
  The Just-In-Time (JIT) compiler dynamically generates machine code for frequently used bytecode sequences in Java applications during their execution. The JIT compiler delivers new optimizations as a result of compiler research, improves optimizations implemented in previous versions of the JIT, and provides better hardware exploitation.
* [**Diagnosing a JIT or AOT problem**](https://www.ibm.com/docs/en/SSYKE2_8.0.0/com.ibm.java.vm.80.doc/docs/jit_pd_diagnose.html)  
  Occasionally, valid bytecodes might compile into invalid native code, causing the Java program to fail. By determining whether the JIT or AOT compiler is faulty and, if so, *where* it is faulty, you can provide valuable help to the Java service team.
* [**Performance of short-running applications**](https://www.ibm.com/docs/en/SSYKE2_8.0.0/com.ibm.java.vm.80.doc/docs/jit_pd_short_run.html)  
  The JIT compiler is tuned for long-running applications typically used on a server. You can use the **-Xquickstart** command-line option to improve the performance of short-running applications.

------------------------------------------------------------------------

Q)What is Compile time error?

These errors occur when we violate the rules present in a syntax. The compile-time error indicates something that we need to fix before compiling the code. A compiler can easily detect these errors. It is the reason why we call them compile-time errors. Here are the most frequent errors (compile-time):

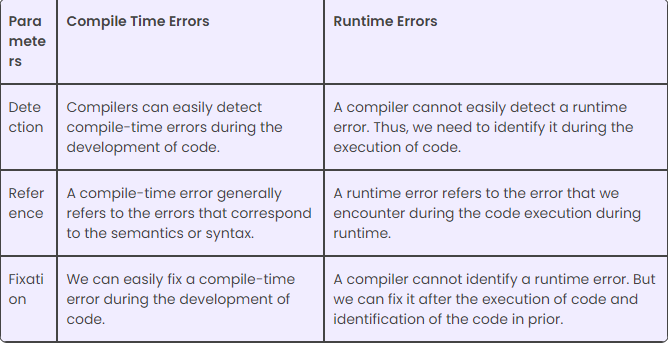
* Terminator- missing semicolon.
* Missing parenthesis.
* Printing the overall value of a variable with no declaration

Q)What is runtime error?

These errors occur during the run-time program execution after a successful compilation. Division error is one of the most common errors (runtime). It occurs due to the division by zero. It is very difficult for a compiler to find out a runtime error because it cannot point out the exact line at which this particular error occurs.

## Difference Between Compile Time Errors and Runtime Errors

Here is a list of the differences present between Compile Time Errors and Runtime Errors.



Q)Who will give CE?

Compile Time Errors are those errors which prevent the code from running because of an incorrect syntax such as a missing semicolon at the end of a statement or a missing bracket, class not found, etc. These errors are detected by the java compiler and an error message is displayed on the screen while compiling. Compile Time Errors are sometimes also referred to as **Syntax errors**. These kind of errors are easy to spot and rectify because the java compiler finds them for you. The compiler will tell you which piece of code in the program got in trouble and its best guess as to what you did wrong. Usually, the compiler indicates the exact line where the error is, or sometimes the line just before it, however, if the problem is with incorrectly nested braces, the actual error may be at the beginning of the block. In effect, syntax errors represent grammatical errors in the use of the programming language.

Q)Who will give RE?

A runtime error in Java is an application error that occurs during the execution of a program. A runtime error occurs when a program is syntactically correct but contains an issue that is only detected during program execution. These issues cannot be caught at compile-time by the Java compiler and are only detected by the Java Virtual Machine (JVM) when the application is running.

Runtime errors are a category of exception that contains several more specific error types. Some of the most common types of runtime errors are:

* IO errors
* Division by zero errors
* Out of range errors
* Undefined object errors

Q)If without .class file we executed a .java file then which error will come?

Q)Can we create empty Java file?

Q)Can we create empty class?

Q)What is the difference between predifined class and user defined class?

Q)What is identifier?

Q)Java supports how many keyword?

Q)How many Comments java supports?

Q)In java how many keywords are available?

Q)Types of literal?

Q)What is jaba tokens?

Q)How many tokens are are there?

1.Identifiers

2.Keywords

3.Literals

4.Operator

5.Separator

Q)What are the programming elements of Java?

-------------------------------------------------------------------------

Q)What is datatype?

1.Type of the memory

2.Size of the memory

3.Type of value and range of the value

4.Diff operator allowed to apply.

5.Result type comes out from an expression

Q)Types of data type?

1.Primitive(8)

2.Referenced(5) (Array,class,Enum,Interface,annotation)

Q)Data type flow chart?

Q)What is a variable?

Q)Types of Variable?

Q)What is Static variable?

Q)What is non-static variable?

Q)What is local variable?

Q)What is type-casting?

Q)What is implicit type casting and explicit type casting?

Q)What is incompatible type?

--------------------------------------------------------------------------

---------------------------------------------------------------------------

Q)What is main method?

Q)What is the use of main method?

Q)What is the syntax of main method?

Q)Can we change the parts the main method?

Q)How many ways we can write main method?

Q)Why main method is public?

Q)Why main method is static?

Q)Why main method has void?

Q)Why main method has parameter?

Q)Why main method parameter is String array type?

Q)Can we create , compile and execute a class without main method?

Q)Can we Overload main method, if yes then which main method will called by JVM?

Q)Main method not available is which time error?