**INDUSTRIAL TRAINING**

**OF**

**CORE JAVA**

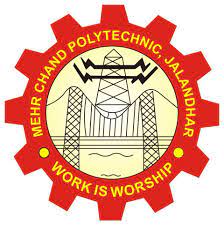
SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT

FOR

THE AWARD OF THE DIPLOMA (CSE)

**SUBMITTED BY:**

**Shoray (558/21)**

****

**MEHR CHAND POLYTECHNIC COLLEGE**

**JALANDHAR**

**ACKNOWLEDGEMENT**

Through this acknowledgement, I express my sincere gratitude to all those people who are associated with this project and are helping me with it to make it a worthwhile experience. First and foremost I would like to thank almighty for giving me courage to make this project. At the outset, I would like to propose a word of thanks for the people who gave me unending support and help in numerous ways.

Firstly, I express my thanks to **Mr. Prince madaan** (HOD CSE DEPTT) who gave me this opportunity to learn the subject in a practical approach who guided me and gave me valuable suggestions regarding the project report.

Secondly, I would like to thanks **Ms. Hitakshi** (Incharge of Final year CSE DEPTT) who gave me this opportunity to learn the subject in a practical approach who guided me and gave me valuable suggestions regarding the project report.

Third, I would like to thank **Mr. Pranav Singh** (Project guide) and FACULTY MEMBERS of O7 SERVICES who are giving me their immense support in completing the project. The atmosphere provided was full of gaining and motivation.

The teachers provide a lot of help in resolving my doubts and making this project successful in shorter time.

Lastly, I would also like to thank my parents and mates who helped me out a lot in finishing the project in limited time.

I am making this project to put forth the knowledge I got during my training. Thanks again to all who are helping me in completion of this project.

**Shoray**

|  |  |  |
| --- | --- | --- |
| **S.no** | **Name of topic** | **Page no** |
| **1** | **About the company profile** | **4 – 5** |
| **2** | **Introduction to java** | **6** |
| **3** | **History of JAVA** | **7 – 9** |
| **4** | **Terminology of JAVA** | **10 -11** |
| **5** | **Features of JAVA** | **12 – 24** |
| **6** | **Basics of JAVA** | **25 – 41** |
| **7** | **Advantages and disadvantages of JAVA** | **42 – 43** |
| **8** | **Netbeans IDE** | **44 – 45** |
| **9** | **Database** | **46 – 48** |
| **10** | **Conclusion** | **48** |
| **11** | **Bibliography** | **49** |

**INDEX**

**CHAPTER-1**

**About the company profile**

****

**Fig 1: O7 services**

**O7 Services is an ISO 9001:2015 Certified Company** and is blessed with Strong and Talented Team. They offer the most advanced IT solutions, supporting full business cycle: preliminary consulting, system development & deployment, quality assurance and **24×7** supports.

With a rich experience of over 9 successful years, they tend to build long lasting strategic partnerships with their clients to ensure affordable prices, timely delivery and measurable business results. Reliability, enthusiasm and creativeness are the keywords they live by and their professionalism gives it a definite shape in the form of our products & solutions. they have vast experience and expertise in Custom Software, Designing, Hosting services, Web Development, Registration of Domain Names with the various latest extensions, Internet and Social Media Marketing, Search Engine Optimization (SEO), Pay Per Click (PPC) Management etc.

Some of the products developed by O7 Services are- Tracking System, Invoice Software, School Management System, Hospital Management system, Parents- Teacher Communication App, Fee Management system, Task Management System, Online Food Ordering App, Security App, Admission system, Inventory Software, Car Servicing App etc.

Apart from this, O7 services also provide**6 Months/ 6 Weeks Industrial Training**to the students in all major IT Courses like:-

* **Android/IOS**
* **MCSE**
* **MCITP**
* **Angular JS**
* **Java**
* **Python**
* **PHP**
* **C/C++**
* **FIREBASE**
* **CorelDraw and many more**

**2.2 Mission-Vision-Goals**

**Vision:**

“*Vision is the Art of seeing what is invisible to others*”

Worldwide reputation is the dream of every company and **O7 Services** want to achieve it through their work.

**Mission**:

“*Your Dream is our Mission*”

O7 Services deliver Innovative Web and Mobile Solutions, ensure complete transparency in the procedure, provide regular updates on the project and give equal priority to all the customers regardless what is the size or type of their project.

**Goals:**

“*A Dream becomes a Goal when Action is taken towards its Achievement*”

Their Goals are SMART- Specific, Measurable, Attainable, Relevant and Timely goals. They tend to provide best Web Solutions, Desktop Applications & Mobile Applications, build the Brand of our Customers, build Trust and Long Lasting Relationships, help our customers with Internet Marketing and be the Best in the field of Industrial Training.

**CHAPTER 2**

**INTRODUCTION TO JAVA**

****

**Fig 2:** JAVA logo

Java is a class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is intended to let application developers ‘*write once, and run anywhere*’ (**WORA**), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java was first released in 1995 and is widely used for developing applications for desktop, web, and mobile devices. Java is known for its simplicity, robustness, and security features, making it a popular choice for enterprise-level applications.

**JAVA** was developed by James Gosling at **Sun Microsystems** Inc in the year **1995** and later acquired by Oracle Corporation. It is a simple programming language. Java makes writing, compiling, and debugging programming easy. It helps to create reusable code and modular programs. JAVA is a class-based, object-oriented programming language and is designed to have as few implementation dependencies as possible. A general-purpose programming language made for developers to *write once run anywhere* that is compiled Java code can run on all platforms that support Java. Java applications are compiled to byte code that can run on any Java Virtual Machine. The syntax of Java is similar to C/C++.

**JAVA** covers concepts useful to 6.005. Students will learn the fundamentals of Java. The focus is on developing high quality, working software that solves real problems.

**CHAPTER 3**

**HISTORY OF JAVA**

****

**Fig 3:** James Gosling

Java’s history is very interesting. It is a programming language created in 1991.James Gosling, Mike Sheridan, and Patrick Naughton, a team of Sun engineers known as the **Green team**initiated the Java language in 1991. **Sun Microsystems** released its first public implementation in 1996 as **Java 1.0**. It provides no-cost -run-times on popular platforms. Java1.0 compiler was re-written in Java by Arthur Van Hoff to strictly comply with its specifications. With the arrival of Java 2, new versions had multiple configurations built for different types of platforms.

In 1997, Sun Microsystems approached the ISO standards body and later formalized Java, but it soon withdrew from the process. At one time, Sun made most of its Java implementations available without charge, despite their proprietary software status. Sun generated revenue from Java through the selling of licenses for specialized products such as the Java Enterprise System.

On November 13, 2006, Sun released much of its Java virtual machine as free, open-source software. On May 8, 2007, Sun finished the process, making all of its JVM’s core code available under open-source distribution terms.

The principles for creating java were simple, robust, secured, high-performance, portable, multi-threaded, interpreted, dynamic, etc. In 1995 Java was developed by **James Gosling**, who is known as the Father of Java. Currently, Java is used in mobile devices, internet programming, games, e-business, etc.

**3.1 Why Java was named as "Oak"?**

Oak is a symbol of strength and chosen as a national tree of many countries like the U.S.A., France, Germany, Romania, etc.



**Fig 4:** oak tree

**3.2 Java programming language is named JAVA. Why?**

After the name OAK, the team decided to give it a new name to it and the suggested words were Silk, Jolt, revolutionary, DNA, dynamic, etc. These all names were easy to spell and fun to say, but they all wanted the name to reflect the essence of technology. In accordance with James Gosling, **Java** among the top names along with **Silk**, and since java was a unique name so most of them preferred it.

Java is the name of an **island** in Indonesia where the first coffee (named java coffee) was produced. And this name was chosen by James Gosling while having coffee near his office. Note that Java is just a name, not an acronym.

### 3.3 Java Version History

### Many java versions have been released till now. The current stable release of Java is Java SE 10.

1. JDK Alpha and Beta (1995)
2. JDK 1.0 (**oak)** (Jan 1996)
3. JDK 1.1 (Feb 1997)
4. J2SE 1.2**(playground)** (Dec 1998)
5. J2SE 1.3 **(Kestrel)**(May 2000)
6. J2SE 1.4 **(Merlin)**(6th Feb 2002)
7. J2SE 5.0 **(Tiger)** (11th Dec 2006)
8. Java SE 6 **(Mustang)** (December 2006)
9. Java SE 7**(Dolphin)** (28th July 2011)
10. Java SE 8 (18th Mar 2014)
11. Java SE 9 (21st Sep 2017)
12. 30th Sep 2004)
13. Java SE 10 (20th Mar 2018
14. Java SE 11 (September 2018)
15. Java SE 12 (March 2019)
16. Java SE 13 (September 2019)
17. Java SE 14 (Mar 2020)
18. Java SE 15 (September 2020)
19. Java SE 16 (Mar 2021)
20. Java SE 17 (September 2021)
21. Java SE 18 (March 2022)
22. Java SE 19 (September 2022)
23. Java SE 20(September, 20th 2022)

Since Java SE1 8 release, the Oracle corporation follows a pattern in which every even version is release in March Month and an odd version released in Sept.

**CHAPTER 4**

**TERMNOLOGY OF JAVA**

**4.1 Java Virtual Machine (JVM)**

This is generally referred to as JVM. There are three execution phases of a program. They are written, compile and run the program.

* Writing a program is done by a java programmer like you and me.
* The compilation is done by the **JAVAC** compiler which is a primary Java compiler included in the Java development kit (JDK). It takes the Java program as input and generates byte code as output.
* In the Runningphase of a program,**JVM** executes the byte code generated by the compiler.

Now, we understood that the function of Java Virtual Machine is to execute the byte code produced by the compiler. Every Operating System has a different JVM but the output they produce after the execution of byte code is the same across all the operating systems. This is why Java is known as a**platform-independent language.**

**4.2 Byte code in**the **Development Process**

As discussed, the Javac compiler of JDK compiles the java source code into byte code so that it can be executed by JVM. It is saved as **.class** file by the compiler. To view the byte code, a disassembler like javap can be used.

**4.3 Java Development Kit (JDK)**

While we were using the term JDK when we learn about byte code and JVM. So, as the name suggests, it is a complete Java development kit that includes everything including compiler, Java Runtime Environment (JRE), java debuggers, java docs, etc. For the program to execute in java, we need to install JDK on our computer in order to create, compile and run the java program.

**4.4 Java Runtime Environment (JRE)**

JDK includes JRE. JRE installation on our computers allows the java program to run however, we cannot compile it. JRE includes a browser, JVM, applet support, and plugins. For running the java program, a computer needs JRE.

**4.5 Garbage Collector**

In Java, programmers can’t delete the objects. To delete or recollect that memory JVM has a program called Garbage Collector. Garbage Collectors can recollect the objects that are not referenced. So Java makes the life of a programmer easy by handling memory management. However, programmers should be careful about their code whether they are using objects that have been used for a long time, because Garbage cannot recover the memory of objects being referenced.

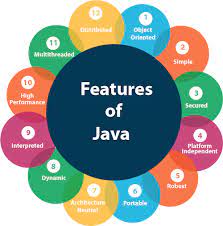
**4.6 ClassPath**

. The classpath is the file path where the java runtime and Java compiler look for **.class** files to load. By default, JDK provides many libraries. If you want to include external libraries they should be added to the classpath.

**CHAPTER 5**

**FEATURES OF JAVA**

Java is a high-level and purely **object oriented programming language**. It is platform independent, robust, secure, and multithreaded programming language which makes it popular among other OOP languages. It is widely used for software, web, and mobile application development, along with this it is also used in big data analytics and server-side technology. Before moving towards features of Java, let us see how Java originated.



**Fig 5:** Features of JAVA

**5.1 Simple**

Java is a simple programming language and easy to understand because it does not contain complexities that exist in prior programming languages. In fact, simplicity was the design aim of Javasoft people, because it has to work on electronic devices where less memory/resources are available. Java contains the same syntax as C, and C++, so the programmers who are switching to Java will not face any problems in terms of syntax. Secondly, the concept of pointers has been completely removed from Java which leads to confusion for a programmer and pointers are also vulnerable to security.

**5.2 Object-oriented**

Java is an Object Oriented Programming Language, which means in Java everything is written in terms of classes and objects. Now, what is an Object? The object is nothing but a real-world entity that can represent any person, place, or thing and can be distinguished from others. Every object near us has some state and behavior associated with it.

For example, my mobile phone is a real-world entity and has states like color, model, brand, camera quality, etc, and these properties are represented by variables. Also mobile is associated with actions like, calling, messaging, photography, etc and these actions are represented by methods in Java.

Now, we saw what an object is and also learned about the state and behavior associated with the object.

What is Class? A collection of objects that exhibits the same state and behavior will come under the same group called class. For example, Samsung, Apple, Vivo, Oppo, Xiaomi, etc are different brands making various models of smart phones, but they all come under the same group known as Mobile Phones.

The main concepts of any Object Oriented Programming language are given below:

1. Class and Object
2. Encapsulation
3. Abstraction
4. Inheritance
5. Polymorphism

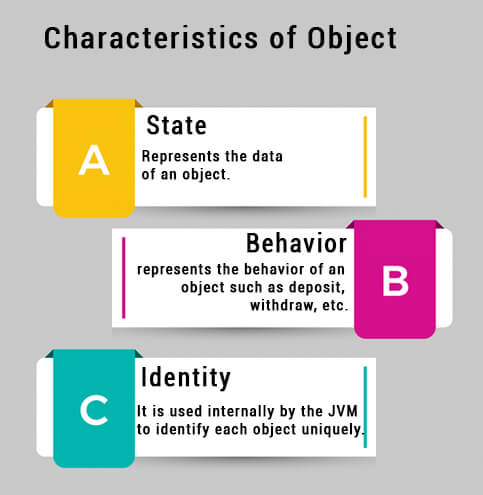
**5.2.1 Class and Object**

An entity that has state and behavior is known as an object e.g., chair, bike, marker, pen, table, car, etc. It can be physical or logical (tangible and intangible). The example of an intangible object is the banking system.

An object has three characteristics:

* **State:** represents the data (value) of an object.
* **Behavior:** represents the behavior (functionality) of an object such as deposit, withdraw, etc.
* **Identity:** An object identity is typically implemented via a unique ID. The value of the ID is not visible to the external user. However, it is used internally by the JVM to identify each object uniquely.

**An object is an instance of a class.** A class is a template or blueprint from which objects are created. So, an object is the instance (result) of a class.



**Fig 6:** Characteristics of Object

**Syntax to create an object:**

Class <class name>

{

Public static void main(String[] args)

{

<class name> <instance name>=new <class name>(parameters);

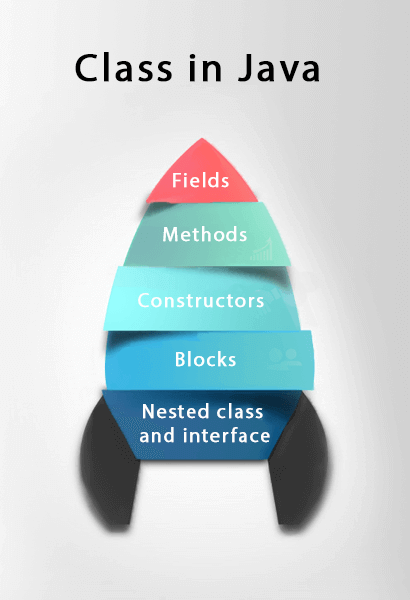
}

}

A **class** is a group of objects which have common properties. It is a template or blueprint from which objects are created. It is a logical entity. It can't be physical.

A class in Java can contain:

* **Fields**
* **Methods**
* **Constructors**
* **Blocks**
* **Nested class and interface**



**Fig 7:** Class in JAVA

**Syntax to create a class:**

class <class name>

{

Field;

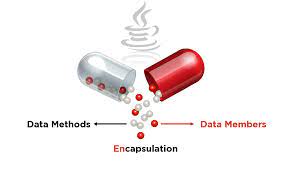
Method;

}

**5.2.2 Encapsulation**

Encapsulation in Java is a powerful mechanism for storing the data members and data methods of a class together. It is done in the form of a secure field accessible by only the members of the same class.

Encapsulation in Java is the process by which data (variables) and the code that acts upon them (methods) are integrated as a single unit. By encapsulating a class's variables, other classes cannot access them, and only the methods of the class can access them.



**Fig 8:** Encapsulation

**What is Encapsulation in Java?**

Encapsulation in Java refers to integrating data (variables) and code (methods) into a single unit. In encapsulation, a class's variables are hidden from other classes and can only be accessed by the methods of the class in which they are found.

**Syntax of encapsulation**:

<Access Modifier> class <Class Name>

{

 Private <Data Members>;

 Private <Data Methods>;

}

**5.2.3 Abstraction**

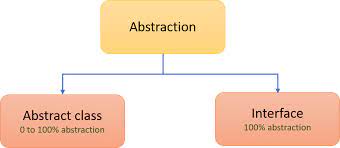
**Data Abstraction** is the property by virtue of which only the essential details are displayed to the user. The trivial or the non-essential units are not displayed to the user.

In Java, abstraction is achieved by interfaces and abstract classes. We can achieve 100% abstraction using interfaces.

Data Abstraction may also be defined as the process of identifying only the required characteristics of an object ignoring the irrelevant details. The properties and behaviors of an object differentiate it from other objects of similar type and also help in classifying/grouping the objects.

**Java Abstract classes and Java Abstract methods**

1. An abstract class is a class that is declared with an abstract keyword.
2. An abstract method is a method that is declared without implementation.
3. An abstract class may or may not have all abstract methods. Some of them can be concrete methods
4. A method-defined abstract must always be redefined in the subclass, thus making overriding compulsory or making the subclass itself abstract.
5. Any class that contains one or more abstract methods must also be declared with an abstract keyword.
6. There can be no object of an abstract class. That is, an abstract class can not be directly instantiated with the *new operator*.
7. An abstract class can have parameterized constructors and the default constructor is always present in an abstract class.



**Fig 9:** Abstraction

**Example of abstraction:**

abstract class Animal

{

public abstract void animalSound();

public void sleep()

{

System.out.println("Zzz");

}

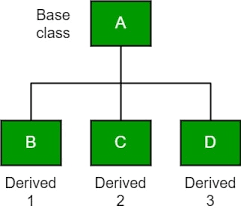
}

**5.2.4 Inheritance**

Inheritance in Java is a mechanism in which one object acquires all the properties and behaviors of a parent object. It is an important part of OOPs (Object Oriented programming system).

The idea behind inheritance in Java is that you can create new classes that are built upon existing classes. When you inherit from an existing class, you can reuse methods and fields of the parent class. Moreover, you can add new methods and fields in your current class also.

Inheritance represents the IS-A relationship which is also known as a *parent-child* relationship.



**Fig 10:** Inheritance

**Terms used in Inheritance**

* **Class:** A class is a group of objects which have common properties. It is a template or blueprint from which objects are created.
* **Sub Class/Child Class:** Subclass is a class which inherits the other class. It is also called a derived class, extended class, or child class.
* **Super Class/Parent Class:** Superclass is the class from where a subclass inherits the features. It is also called a base class or a parent class.
* **Reusability:** As the name specifies, reusability is a mechanism which facilitates you to reuse the fields and methods of the existing class when you create a new class. You can use the same fields and methods already defined in the previous class.

**Syntax of inheritance:**

class Subclass-name extends Superclass-name

{

   methods and fields

}

The ‘*extends*’ keyword indicates that you are making a new class that derives from an existing class. The meaning of "extends" is to increase the functionality.

**5.2.5 Polymorphism**

**Polymorphism in Java** is a concept by which we can perform a *single action in different ways*. Polymorphism is derived from 2 Greek words: poly and morphs. The word "poly" means many and "morphs" means forms. So polymorphism means many forms.

There are two types of polymorphism in Java: compile-time polymorphism and runtime polymorphism. We can perform polymorphism in java by method overloading and method overriding.

If you overload a static method in Java, it is the example of compile time polymorphism. Here, we will focus on runtime polymorphism in java.

**Runtime polymorphism** or **Dynamic Method Dispatch** is a process in which a call to an overridden method is resolved at runtime rather than compile-time.

In this process, an overridden method is called through the reference variable of a superclass. The determination of the method to be called is based on the object being referred to by the reference variable.

**Upcasting**

If the reference variable of Parent class refers to the object of Child class, it is known as upcasting.

For example:

**class** A{}

**class** B **extends** A{}

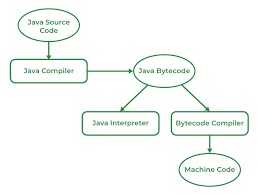
A a=**new** B();



**Fig 11:** Upcasting

**5.3 Platform independent**

The design objective of javasoft people is to develop a language that must work on any platform. Here platform means a type of operating system and hardware technology. Java allows programmers to write their program on any machine with any configuration and to execute it on any other machine having different configurations.

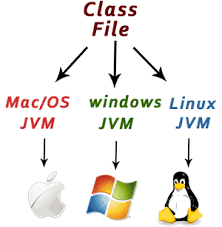


**Fig 12:** independent JAVA

In Java, Java source code is compiled to bytecode and this bytecode is not bound to any platform. In fact, this bytecode is only understandable by the Java Virtual Machine which is installed in our system. What I meant to say is that every operating system has its own version of JVM, which is capable of reading and converting bytecode to an equivalent machine’s native language. This reduces the overhead of programmers writing system-specific code. Now programmers write programs only once, compile them, generate the bytecode and then export it anywhere.

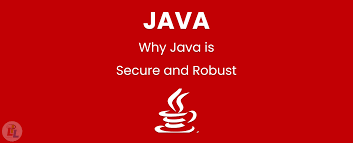
**5.4 Portable**

The WORA (Write Once Run Anywhere) concept and platform-independent feature make Java portable. Now using the Java programming language, developers can yield the same result on any machine, by writing code only once. The reason behind this is JVM and bytecode. Suppose you wrote any code in Java, then that code is first converted to equivalent bytecode which is only readable by JVM. We have different versions of JVM for different platforms. Windows machines have their own version of JVM, Linux has its own and macOS has its own version of JVM. So if you distribute your bytecode to any machine, the JVM of that machine would translate the bytecode into the respective machine code. In this way portability lets the programmers focus on development and productivity rather than writing different code for different platforms.



**Fig 13:** Portability

**5.5 Robust**



**Fig 14:** Robust in JAVA

The Java Programming language is robust, which means it is capable of handling unexpected termination of a program. There are 2 reasons behind this, first, it has a most important and helpful feature called Exception Handling. If an exception occurs in java code then no harm will happen whereas, in other low-level languages, the program will crash.

Another reason why Java is strong lies in its memory management features. Unlike other low-level languages, Java provides a runtime Garbage collector offered by JVM, which collects all the unused variables. The garbage collector is a special program under JVM that runs from time to time and detects any unused variables and objects and removes them from the memory to free up space. But in the case of other prior languages, there is no such program to handle memory management, programmers are solely responsible for allocating and deallocating memory spaces, otherwise, the program may crash due to insufficient memory.

**5.6 Secure**

In today’s era, security is a major concern of every application. As of now, every device is connected to each other using the internet and this opens up the possibility of hacking. And our application built using java also needs some sort of security. So Java also provides security features to the programmers. Security problems like virus threats, tampering, eavesdropping, and impersonation can be handled or minimized using Java. Encryption and Decryption feature to secure your data from *eavesdropping*and *tampering*over the internet. An *Impersonation*is an act of pretending to be another person on the internet.

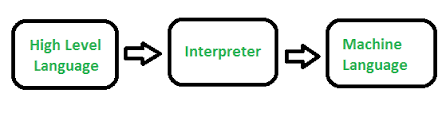


**Fig 15:** Security

The impersonation problem is a digital signature, a file that contains personal identification information in an unreadable format. Digital Signature can be generated using Java. *Virus*is a program that is capable of harming our system and this is generally spread with .exe files, image files, and video files but cannot be spread using a text file the good thing is java bytecode is also a text file (yes .class file also a text file with non-human-readable format). Even if somebody tries to add virus code in a bytecode file, then also we are safe, because our JVM is smart enough to distinguish viruses from normal programs. If a virus is found in a bytecode file, JVM will throw an exception and abort execution.

**5.7 Interpreted**

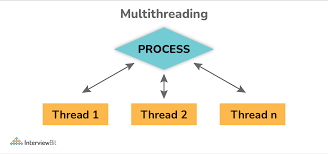
In programming languages, you have learned that they use either the compiler or an interpreter, but Java programming language uses both a compiler and an interpreter. Java programs are compiled to generate bytecode files then JVM interprets the bytecode file during execution. Along with this JVM also uses a JIT compiler (it increases the speed of execution).



**Fig 16:** working of interpreter

**5.8 Multi-Threaded**

Thread is a lightweight and independent subprocess of a running program (i.e, process) that shares resources. And when multiple threads run simultaneously is called multithreading. In many applications, you have seen multiple tasks running simultaneously, for example, Google Docs where while typing text, the spell check and autocorrect tasks are running.



**Fig 17:** Multithreading

The server also uses multithreading to provide its services to multiple client requests. In Java, you can create threads in two ways, either by implementing the Runnable interface or by extending the Thread class.

**CHAPTER 6**

**BASICS OF JAVA**

Irrespective of the programming language you choose to learn, the basic concepts of programming are similar across languages. Some of these concepts include:

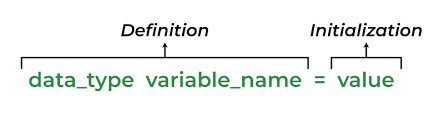
* Variable Declaration
* Basic Syntax
* Data Type and Structures
* Flow Control Structures (Conditionals and loops)
* Functional Programming
* Object-Oriented Programming
* Debugging

**6.1 Variable declaration**

**Variables** are containers for storing data values, a memory location for a data type. Variables are created using a declaration or keyword that varies across languages.

Variable names are usually alphanumeric, that is, they contain a-z and 0-9. They can also include special characters like underscore or the dollar sign.

Variables can hold values of any data type supported by the programming language. This value may change during program execution.



**Fig 18:** Syntax for variable declaration

**Types of variables**

In Java, there are three types of variables:

1. Local Variables
2. Instance Variables
3. Static Variables

**1) Local Variables**

Local Variables are a variable that are declared inside the body of a method.

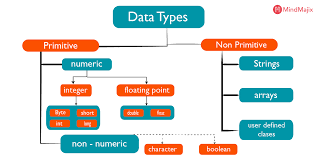
**2) Instance Variables**

Instance variables are defined without the STATIC keyword .They are defined Outside a method declaration. They are Object specific and are known as instance variables.

**3) Static Variables**

Static variables are initialized only once, at the start of the program execution. These variables should be initialized first, before the initialization of any instance variables**.**

**6.2 Data types**

****

**Fig 19:** data types in java

**Data Types in Java** are defined as specifiers that allocate different sizes and types of values that can be stored in the variable or an identifier. Java has a rich set of data types. Data types in Java can be divided into two parts :

1. **Primitive Data Types** :- which include integer, character, boolean, and float
2. **Non-primitive Data Types** :- which include classes, arrays and interfaces.

**Primitive Data Types**

Primitive Data Types are predefined and available within the Java language. Primitive values do not share state with other primitive values.

There are 8 primitive types: byte, short, int, long, char, float, double, and Boolean

|  |  |  |
| --- | --- | --- |
| Data Type | Default Value | Default size |
| Byte | 0 | 1 byte |
| Short | 0 | 2 bytes |
| Int | 0 | 4 bytes |
| Long | 0L | 8 bytes |
| Float | 0.0f | 4 bytes |
| Double | 0.0d | 8 bytes |
| boolean | False | 1 bit |
| Char | ‘\u0000’ | 2 bytes |

**Non-Primitive Data Types**

Non-primitive data types are called **reference types** because they refer to objects.

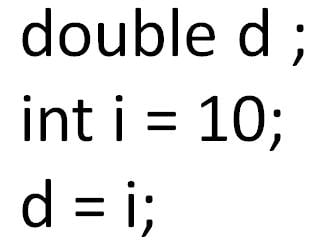
The main difference between **primitive** and **non-primitive** data types are:

* Primitive types are predefined (already defined) in Java. Non-primitive types are created by the programmer and is not defined by Java (except for String).
* Non-primitive types can be used to call methods to perform certain operations, while primitive types cannot.
* A primitive type has always a value, while non-primitive types can be null.
* A primitive type starts with a lowercase letter, while non-primitive types starts with an uppercase letter.

**6.3 Typecasting**

A variable of one type can receive the value of another type. Here there are 2 cases –

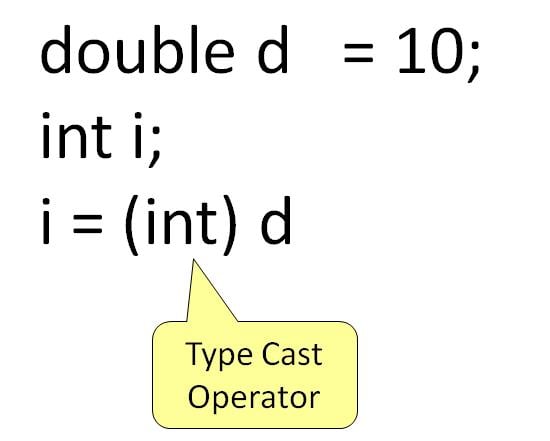
**Case 1)**Variable of smaller capacity is be assigned to another variable of bigger capacity.

[](https://www.guru99.com/images/uploads/2012/07/java-type-conversion.jpg)

**Fig 20:** typecasting in large capacity data type

This process is Automatic, and non-explicit is known as ***Conversion***

**Case 2)**Variable of larger capacity is be assigned to another variable of smaller capacity

[[](https://www.guru99.com/images/uploads/2012/07/java-type-cast-operator.jpg)](https://www.guru99.com/images/uploads/2012/07/java-type-cast-operator.jpg)

**Fig 21:** typecasting in small capacity data type

In such cases, you have to explicitly specify the **type cast operator. This process is known as *Type Casting.***

In case, you do not specify a type cast operator; the compiler gives an error. Since this rule is enforced by the compiler, it makes the programmer aware that the conversion he is about to do may cause some loss in data and prevents **accidental losses.**

**6.4 Looping statement**

Looping in programming languages is a feature which facilitates the execution of a set of instructions/functions repeatedly while some condition evaluates to true. Java provides three ways for executing the loops. While all the ways provide similar basic functionality, they differ in their syntax and condition checking time.

   java provides Three types of Conditional statements this second  type is loop statement .

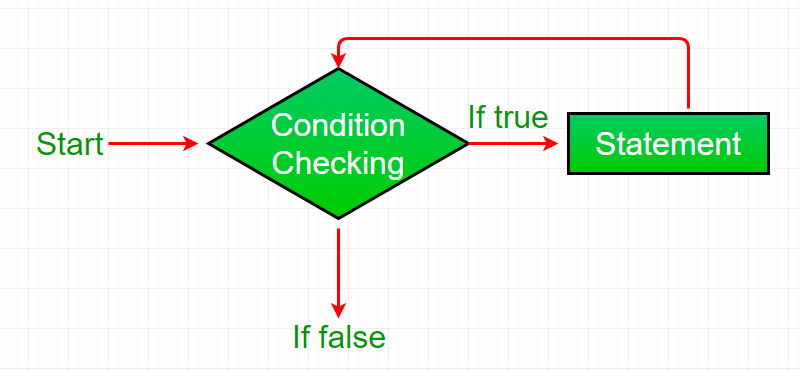
**while loop:** A while loop is a control flow statement that allows code to be executed repeatedly based on a given Boolean condition. The while loop can be thought of as a repeating if statement.

**Syntax :**

while (boolean condition)

{ loop statements... }

**Flowchart**



**Fig 22:** flowchart for while loop

* While loop starts with the checking of Boolean condition. If it evaluated to true, then the loop body statements are executed otherwise first statement following the loop is executed. For this reason it is also called **Entry control loop**
* Once the condition is evaluated to true, the statements in the loop body are executed. Normally the statements contain an update value for the variable being processed for the next iteration.
* When the condition becomes false, the loop terminates which marks the end of its life cycle.

**For loop:** for loop provides a concise way of writing the loop structure. Unlike a while loop, a for statement consumes the initialization, condition and increment/decrement in one line thereby providing a shorter, easy to debug structure of looping.

**Syntax:**

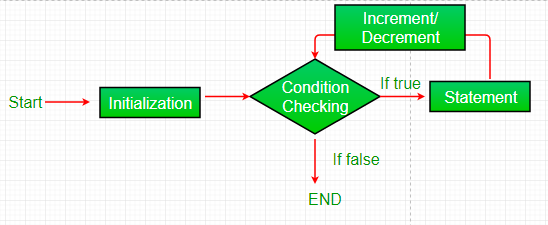
for (initialization condition; testing condition;increment/decrement)

{

statement(s)

}

**Flowchart:**



**Fig 23:** flowchart for for loop

* + **Initialization condition:**Here, we initialize the variable in use. It marks the start of a for loop. An already declared variable can be used or a variable can be declared, local to loop only.
  + **Testing Condition:** It is used for testing the exit condition for a loop. It must return a boolean value. It is also an **Entry Control Loop** as the condition is checked prior to the execution of the loop statements.
  + **Statement execution:** Once the condition is evaluated to true, the statements in the loop body are executed.
  + **Increment/ Decrement:**It is used for updating the variable for next iteration.
  + **Loop termination:**When the condition becomes false, the loop terminates marking the end of its life cycle.

**Do while:** do while loop is similar to while loop with only difference that it checks for condition after executing the statements, and therefore is an example of **Exit Control Loop.**

**Syntax:**

do

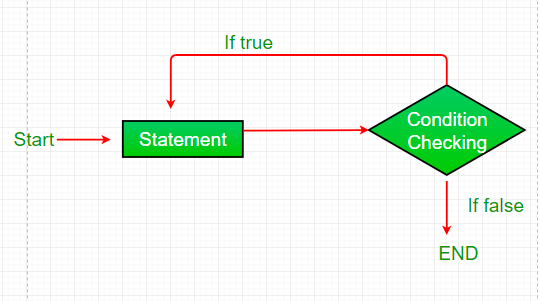
{

statements..

}

while (condition);

**Flowchart:**

[](https://media.geeksforgeeks.org/wp-content/uploads/loop3.png)

**Fig 24:** flowchart for do while loop

* do while loop starts with the execution of the statement(s). There is no checking of any condition for the first time.
* After the execution of the statements, and update of the variable value, the condition is checked for true or false value. If it is evaluated to true, next iteration of loop starts.
* When the condition becomes false, the loop terminates which marks the end of its life cycle.
* It is important to note that the do-while loop will execute its statements atleast once before any condition is checked, and therefore is an example of exit control loop

**6.5 Conditional Statements**

The Java *if statement* is used to test the condition. It checks boolean condition: *true* or *false*. There are various types of if statement in Java.

* if statement
* if-else statement
* if-else-if ladder
* nested if statement

**Java if Statement**

The Java if statement tests the condition. It executes the *if block* if condition is true.

**Syntax:**

1. **if**(condition){
2. //code to be executed
3. }



**Fig 25:** flowchart for if statement

**Java if-else Statement**

The Java if-else statement also tests the condition. It executes the *if block* if condition is true otherwise *else block* is executed.

**Syntax:**

1. **if**(condition){
2. //code if condition is true
3. }**else**{
4. //code if condition is false
5. }



**Fig 26:** flowchart for if else statement

**Java if-else-if ladder Statement**

The if-else-if ladder statement executes one condition from multiple statements.

**Syntax:**

1. **if**(condition1){
2. //code to be executed if condition1 is true
3. }**else** **if**(condition2){
4. //code to be executed if condition2 is true
5. }
6. **else** **if**(condition3){
7. //code to be executed if condition3 is true
8. }
9. ...
10. **else**{
11. //code to be executed if all the conditions are false
12. }



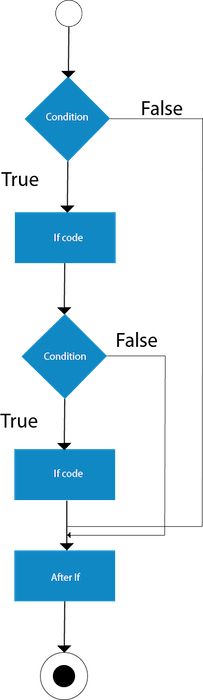
**Fig 27:** flowchart for else if ladder

**Java Nested if statement**

The nested if statement represents the *if block within another if block*. Here, the inner if block condition executes only when outer if block condition is true.

**Syntax:**

1. **if**(condition){
2. //code to be executed
3. **if**(condition){
4. //code to be executed
5. }
6. }



**Fig 28:** flowchart for nested if statement

**Java Switch Statement**

The Java *switch statement* executes one statement from multiple conditions. It is like if-else if ladder statement. The switch statement works with byte, short, int, long, enum types, String and some wrapper types like Byte, Short, Int, and Long. Since Java 7, you can use strings in the switch statement.

In other words, the switch statement tests the equality of a variable against multiple values.

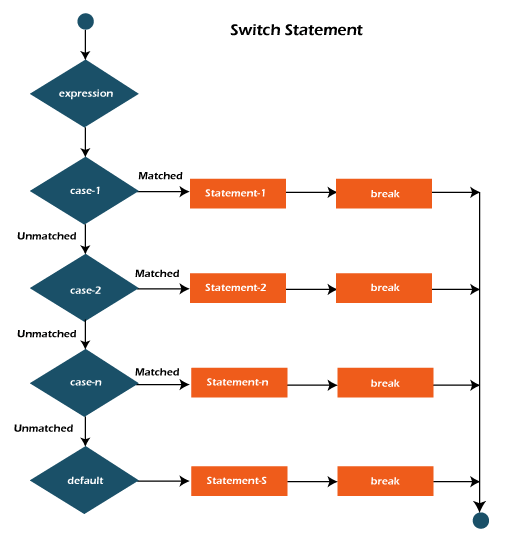
Points to Remember

* There can be *one or N number of case values* for a switch expression.
* The case value must be of switch expression type only. The case value must be *literal or constant*. It doesn't allow variables.
* The case values must be *unique*. In case of duplicate value, it renders compile-time error.
* The Java switch expression must be of *byte, short, int, long (with its Wrapper type), enums and string*.
* Each case statement can have a *break statement* which is optional. When control reaches to the break statement, it jumps the control after the switch expression. If a break statement is not found, it executes the next case.
* The case value can have a *default label* which is optional.

**Syntax:**

1. **switch**(expression){
2. **case** value1:
3. //code to be executed;
4. **break**;  //optional
5. **case** value2:
6. //code to be executed;
7. **break**;  //optional
8. ......
10. **default**:
11. code to be executed **if** all cases are not matched;
12. }

**Flowchart of Switch Statement**



**Fig 29:** flowchart for switch statement

**6.6 Operators in java**

Operators in Java are the symbols used for performing specific operations in Java. Operators make tasks like addition, multiplication, etc which look easy although the implementation of these tasks is quite complex.

## Types of Operators in Java

There are multiple types of operators in Java all are mentioned below:

1. Arithmetic Operators
2. Unary Operators
3. Assignment Operator
4. Relational Operators
5. Logical Operators
6. Ternary Operator
7. Bitwise Operators
8. Shift Operators
9. instance of operator

### ****1. Arithmetic Operators****

They are used to perform simple arithmetic operations on primitive data types.

* **\* :**Multiplication
* **/ :**Division
* **% :**Modulo
* **+ :**Addition
* **– :**Subtraction

**2. Unary Operators**

Unary operators need only one operand. They are used to increment, decrement, or negate a value.

* **– :** **Unary minus**, used for negating the values.
* **+ :** **Unary plus** indicates the positive value (numbers are positive without this, however). It performs an automatic conversion to int when the type of its operand is the byte, char, or short. This is called unary numeric promotion.
* **++ :** **Increment operator**, used for incrementing the value by 1. There are two varieties of increment operators.
  + **Post-Increment:**Value is first used for computing the result and then incremented.
  + **Pre-Increment:**Value is incremented first, and then the result is computed.
* **– –  : Decrement operator**, used for decrementing the value by 1. There are two varieties of decrement operators.
  + **Post-decrement:**Value is first used for computing the result and then decremented.
  + **Pre-Decrement: The value** is decremented first, and then the result is computed.
* **! : Logical not operator**, used for inverting a boolean value.

**3. Assignment Operator**

**‘=’** Assignment operator is used to assign a value to any variable. It has right-to-left associativity, i.e. value given on the right-hand side of the operator is assigned to the variable on the left, and therefore right-hand side value must be declared before using it or should be a constant.

The general format of the assignment operator is:

**Variable = value;**

In many cases, the assignment operator can be combined with other operators to build a shorter version of the statement called a **Compound Statement**. For example, instead of a **=** a+5, we can write a **+=** 5.

* **+=**, for adding the left operand with the right operand and then assigning it to the variable on the left.
* **-=**, for subtracting the right operand from the left operand and then assigning it to the variable on the left.
* **\*=**, for multiplying the left operand with the right operand and then assigning it to the variable on the left.
* **/=**, for dividing the left operand by the right operand and then assigning it to the variable on the left.
* **%=**, for assigning the modulo of the left operand by the right operand and then assigning it to the variable on the left.

**4. Relational Operators**

These operators are used to check for relations like equality, greater than, and less than. They return boolean results after the comparison and are extensively used in looping statements as well as conditional if-else statements. The general format is,

Variable **relation\_operator** value

Some of the relational operators are-

* **==, Equal to**returns true if the left-hand side is equal to the right-hand side.
* **!=, Not Equal to**returns true if the left-hand side is not equal to the right-hand side.
* **<, less than:**returns true if the left-hand side is less than the right-hand side.
* **<=, less than or equal to**returns true if the left-hand side is less than or equal to the right-hand side.
* **>, Greater than:**returns true if the left-hand side is greater than the right-hand side.
* **>=, Greater than or equal to**returns true if the left-hand side is greater than or equal to the right-hand side.

**5. Logical Operators**

These operators are used to perform “logical AND” and “logical OR” operations, i.e., a function similar to AND gate and OR gate in digital electronics. One thing to keep in mind is the second condition is not evaluated if the first one is false, i.e., it has a short-circuiting effect. Used extensively to test for several conditions for making a decision. Java also has “Logical NOT”, which returns true when the condition is false and vice-versa

Conditional operators are:

* **&&, Logical AND:**returns true when both conditions are true.
* **||, Logical OR:**returns true if at least one condition is true.
* **!, Logical NOT:**returns true when a condition is false and vice-versa

**6. Ternary operator**

The ternary operator is a shorthand version of the if-else statement. It has three operands and hence the name Ternary.

The general format is:

Condition **?** if true **:** if false

The above statement means that if the condition evaluates to true, then execute the statements after the ‘?’ else execute the statements after the ‘:’.

**7. Bitwise Operators**

These operators are used to perform the manipulation of individual bits of a number. They can be used with any of the integer types. They are used when performing update and query operations of the Binary indexed trees.

* **&, Bitwise AND operator:**returns bit by bit AND of input values.
* **|, Bitwise OR operator:**returns bit by bit OR of input values.
* **^, Bitwise XOR operator:**returns bit-by-bit XOR of input values.
* **~, Bitwise Complement Operator:**This is a unary operator which returns the one’s complement representation of the input value, i.e., with all bits inverted.

**8. Shift Operators**

These operators are used to shift the bits of a number left or right, thereby multiplying or dividing the number by two, respectively. They can be used when we have to multiply or divide a number by two. General format-

Number **shift\_op** number\_of\_places\_to\_shift;

* **<<, Left shift operator:**shifts the bits of the number to the left and fills 0 on voids left as a result. Similar effect as multiplying the number with some power of two.
* **>>, Signed Right shift operator:**shifts the bits of the number to the right and fills 0 on voids left as a result. The leftmost bit depends on the sign of the initial number. Similar effect to dividing the number with some power of two.
* **>>>, Unsigned Right shift operator:**shifts the bits of the number to the right and fills 0 on voids left as a result. The leftmost bit is set to 0.



**Fig 29:** operators in java

**Chapter 7**

**ADVANTAGES AND DISADVATAGES**

**OF JAVA**

Java is a general-purpose, robust, secure, and object-oriented programming language. It is a high-level language, I.e., its syntax uses English like language. It was developed by Sun Microsystems in the year 1995. It is now maintained and distributed by Oracle. Java has its runtime environment and API; therefore, it is also called a platform.

Java is used in a large number of applications over the years. However, it has various advantages and disadvantages given below.

**7.1 Advantages:**

**1. Simple**

Java is a simple programming language since it is easy to learn and easy to understand. Its syntax is based on C++, and it uses automatic garbage collection; therefore, we don't need to remove the unreferenced objects from memory. Java has also removed the features like explicit pointers, operator overloading, etc., making it easy to read and write.

Backward Skip 10sPlay VideoForward Skip 10s

**2. Object-Oriented**

Java uses an object-oriented paradigm, which makes it more practical. Everything in Java is an object which takes care of both data and behavior. Java uses object-oriented concepts like object, class, inheritance, encapsulation, polymorphism, and abstraction.

**3. Secured**

Java is a secured programming language because it doesn't use Explicit pointers. Also, Java programs run inside the virtual machine sandbox. JRE also provides a classloader, which is used to load the class into JVM dynamically. It separates the class packages of the local file system from the ones that are being imported from the network.

**4. Robust**

Java is a robust programming language since it uses strong memory management. We can also handle exceptions through the Java code. Also, we can use type checking to make our code more secure. It doesn't provide explicit pointers so that the programmer cannot access the memory directly from the code.

**5. Platform independent**

Java code can run on multiple platforms directly, I.e., we need not compile it every time. It is right once, runs anywhere language (WORA) which can be converted into byte code at the compile time. The byte code is a platform-independent code that can run on multiple platforms.

**6. Multi-Threaded**

Java uses a multi-threaded environment in which a bigger task can be converted into various threads and run separately. The main advantage of multi-threading is that we need not provide memory to every running thread.

**7.2 Disadvantages**

**1. Performance**

Java needs to be interpreted during runtime, which allows it to run on every operating system, but it also makes it perform slower than the languages like [C](https://www.javatpoint.com/c-programming-language-tutorial) and [C++](https://www.javatpoint.com/cpp-tutorial). On the other hand, the C++ program needs to be compiled on each operating system, directly to binary and therefore runs faster.

**2. Memory consumption**

Java program consumes more memory since it runs on top of Java virtual machine.

**3. Cost**

Java programming language is a bit costly due to its higher processing and memory requirements. We need better hardware to run the Java program.

**4. Less machine interactive**

Java lacks when it comes to interacting directly with machines, making it less viable for the software that needs to run quickly and run directly with the machine, as explicit pointers are also missing in Java.

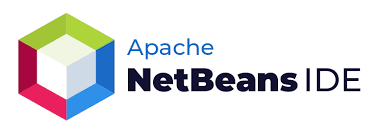
**5. Garbage collection**

Java provides automatic garbage collection that cannot be controlled by the programmer. It doesn't provide the methods like delete() and free() to free the memory.

However, due to the various disadvantages, Java is one of the most used programming languages due to its advantages, making it platform-independent, secure and a robust programming language.

**Chapter 8**

**NETBEANS IDE**

****

**Fig 30:** netbeans IDE

NetBeans IDE is a free and open source integrated development environment for application development on Windows, Mac, Linux, and Solaris operating systems.

The IDE simplifies the development of web, enterprise, desktop, and mobile applications that use the Java and HTML5 platforms. The IDE also offers support for the development of PHP and C/C++ applications.

NetBeans IDE offers first-class tools for Java web, enterprise, desktop, and mobile application development. It is consistently the first IDE to support the latest versions of the JDK, Java EE, and JavaFX. It provides smart overviews to help you understand and manage your applications, including ouf-of-the-box support for popular technologies such as Maven.

**NetBeans** is an integrated development environment(IDE) for Java. NetBeans allows applications to be developed from a set of modular software components called *modules*. NetBeans runs on Windows, macOS, Linux and Solaris. In addition to Java development, it has extensions for other languages like PHP, C, C++, HTML5 and JavaScript. Applications based on NetBeans, including the NetBeans IDE, can be extended by third party developers.

NetBeans began in 1996 as Xelfi (word play on *Delphi*), a Java IDE student project under the guidance of the Faculty of Mathematics and Physics at Charles University in Prague. In 1997, Roman Staněk formed a company around the project and produced commercial versions of the NetBeans IDE until it was bought by Sun Microsystems in 1999. Sun open-sourced the NetBeans IDE in June of the following year. Since then, the NetBeans community has continued to grow. In 2010, Sun (and thus NetBeans) was acquired by Oracle Corporation.

**Modularity:** All the functions of the IDE are provided by modules. Each module provides a **Modularity:** All the functions of the IDE are provided by modules. Each module provides a well-defined function, such as support for the Java language, editing, or support for the CVS versioning system, and SVN. NetBeans contains all the modules needed for Java development in a single download, allowing the user to start working immediately. Modules also allow NetBeans to be extended. New features, such as support for other programming languages, can be added by installing additional modules. For instance, Sun Studio, Sun Java Studio Enterprise, and Sun Java Studio Creator from Sun Microsystems are all based on the NetBeans IDE.

**License:** The IDE is licensed under the Apache License 2.0. Previously, from July 2006 through 2007, NetBeans IDE was licensed under Sun's Common Development and Distribution License (CDDL), a license based on the Mozilla Public License (MPL). In October 2007, Sun announced that NetBeans would henceforth be offered under a dual license of the CDDL and the GPL version 2 licenses, with the GPL linking exception for GNU Classpath. Oracle has donated NetBeans Platform and IDE to the Apache Foundation where it underwent incubation and graduated as a top level project in April 2019.

well-defined function, such as support for the Java language, editing, or support for the CVS versioning system, and SVN. NetBeans contains all the modules needed for Java development in a single download, allowing the user to start working immediately. Modules also allow NetBeans to be extended. New features, such as support for other programming languages, can be added by installing additional modules. For instance, Sun Studio, Sun Java Studio Enterprise, and Sun Java Studio Creator from Sun Microsystems are all based on the NetBeans IDE.

**License:** The IDE is licensed under the [Apache License 2.0](https://en.wikipedia.org/wiki/Apache_License_2.0). Previously, from July 2006 through 2007, NetBeans IDE was licensed under Sun's [Common Development and Distribution License](https://en.wikipedia.org/wiki/Common_Development_and_Distribution_License) (CDDL), a license based on the [Mozilla Public License](https://en.wikipedia.org/wiki/Mozilla_Public_License) (MPL). In October 2007, Sun announced that NetBeans would henceforth be offered under a [dual license](https://en.wikipedia.org/wiki/Dual_license) of the CDDL and the [GPL](https://en.wikipedia.org/wiki/GPL) version 2 licenses, with the [GPL linking exception](https://en.wikipedia.org/wiki/GPL_linking_exception) for [GNU Classpath](https://en.wikipedia.org/wiki/GNU_Classpath).[[10]](https://en.wikipedia.org/wiki/NetBeans#cite_note-10) Oracle has donated NetBeans Platform and IDE to the Apache Foundation where it underwent incubation and graduated as a top level project in April 2019.

**Chapter 9**

**DATABASE**

**9.1 XAMPP**



**Fig 31:** xampp

**XAMPP** ([/ˈzæmp/](https://en.wikipedia.org/wiki/Help:IPA/English) or [/ˈɛks.æmp/](https://en.wikipedia.org/wiki/Help:IPA/English))[[2]](https://en.wikipedia.org/wiki/XAMPP#cite_note-kaiseidlerinterview-2) is a [free and open-source](https://en.wikipedia.org/wiki/Free_and_open-source) [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) [web server](https://en.wikipedia.org/wiki/Web_server) [solution stack](https://en.wikipedia.org/wiki/Solution_stack) package developed by Apache Friends,[[2]](https://en.wikipedia.org/wiki/XAMPP#cite_note-kaiseidlerinterview-2) consisting mainly of the [Apache HTTP Server](https://en.wikipedia.org/wiki/Apache_HTTP_Server), [MariaDB](https://en.wikipedia.org/wiki/MariaDB" \o "MariaDB) [database](https://en.wikipedia.org/wiki/Database), and [interpreters](https://en.wikipedia.org/wiki/Interpreter_(computing)) for scripts written in the [PHP](https://en.wikipedia.org/wiki/PHP) and [Perl](https://en.wikipedia.org/wiki/Perl) [programming languages](https://en.wikipedia.org/wiki/Programming_language).[[3]](https://en.wikipedia.org/wiki/XAMPP#cite_note-x_mariadb-3)[[4]](https://en.wikipedia.org/wiki/XAMPP#cite_note-4) Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server possible.

XAMPP's ease of deployment means a [WAMP](https://en.wikipedia.org/wiki/WAMP) or [LAMP](https://en.wikipedia.org/wiki/LAMP_(software_bundle)) stack can be installed quickly and simply on an operating system by a developer, with the advantage that common add-in applications such as [WordPress](https://en.wikipedia.org/wiki/WordPress) and [Joomla!](https://en.wikipedia.org/wiki/Joomla!" \o "Joomla!) can also be installed with similar ease using [Bitnami](https://en.wikipedia.org/wiki/Bitnami" \o "Bitnami).

The term XAMPP is an apparent [acronym](https://en.wikipedia.org/wiki/Acronym_and_initialism). However, there is no official acronym expansion specified on the Apache Friends website. Their homepage header reads "XAMPP Apache + MariaDB + PHP + Perl", indicating that this [abbreviation](https://en.wikipedia.org/wiki/Abbreviation) is a [recursive acronym](https://en.wikipedia.org/wiki/Recursive_acronym).

The term can be unofficially broken down as follows:

|  |  |
| --- | --- |
| **Letter** | **Meaning** |
| X | An [ideographic](https://en.wikipedia.org/wiki/Ideogram) letter referring to [cross-platform](https://en.wikipedia.org/wiki/Cross-platform)[[5]](https://en.wikipedia.org/wiki/XAMPP#cite_note-abbreviations.com-5) |
| A | [Apache](https://en.wikipedia.org/wiki/Apache_HTTP_Server),[[6]](https://en.wikipedia.org/wiki/XAMPP#cite_note-xampp-6) or its expanded form, Apache HTTP Server[[5]](https://en.wikipedia.org/wiki/XAMPP#cite_note-abbreviations.com-5) |
| M | [MariaDB](https://en.wikipedia.org/wiki/MariaDB)[[7]](https://en.wikipedia.org/wiki/XAMPP#cite_note-mariadb-7) (formerly [MySQL](https://en.wikipedia.org/wiki/MySQL)[[5]](https://en.wikipedia.org/wiki/XAMPP#cite_note-abbreviations.com-5)[[7]](https://en.wikipedia.org/wiki/XAMPP#cite_note-mariadb-7)) |
| P | [PHP](https://en.wikipedia.org/wiki/PHP)[[6]](https://en.wikipedia.org/wiki/XAMPP#cite_note-xampp-6)[[5]](https://en.wikipedia.org/wiki/XAMPP#cite_note-abbreviations.com-5) |
| P | [Perl](https://en.wikipedia.org/wiki/Perl)[[6]](https://en.wikipedia.org/wiki/XAMPP#cite_note-xampp-6)[[5]](https://en.wikipedia.org/wiki/XAMPP#cite_note-abbreviations.com-5) |

**9.2 MySql**

****

**MySQL** ([/ˌmaɪˌɛsˌkjuːˈɛl/](https://en.wikipedia.org/wiki/Help:IPA/English))[[5]](https://en.wikipedia.org/wiki/MySQL#cite_note-whatismysql-5) is an [open-source](https://en.wikipedia.org/wiki/Open-source_software) [relational database management system](https://en.wikipedia.org/wiki/Relational_database_management_system) (RDBMS).[[5]](https://en.wikipedia.org/wiki/MySQL#cite_note-whatismysql-5)[[6]](https://en.wikipedia.org/wiki/MySQL#cite_note-6) Its name is a combination of "My", the name of co-founder [Michael Widenius](https://en.wikipedia.org/wiki/Michael_Widenius)'s daughter My,[[7]](https://en.wikipedia.org/wiki/MySQL#cite_note-7) and "SQL", the acronym for [Structured Query Language](https://en.wikipedia.org/wiki/Structured_Query_Language). A [relational database](https://en.wikipedia.org/wiki/Relational_database) organizes data into one or more data tables in which data may be related to each other; these relations help structure the data. SQL is a language that programmers use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an [operating system](https://en.wikipedia.org/wiki/Operating_system) to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups.

MySQL is [free and open-source software](https://en.wikipedia.org/wiki/Free_and_open-source_software) under the terms of the [GNU General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License), and is also available under a variety of [proprietary](https://en.wikipedia.org/wiki/Proprietary_software) licenses. MySQL was owned and sponsored by the [Swedish](https://en.wikipedia.org/wiki/Sweden) company [MySQL AB](https://en.wikipedia.org/wiki/MySQL_AB), which was bought by [Sun Microsystems](https://en.wikipedia.org/wiki/Sun_Microsystems) (now [Oracle Corporation](https://en.wikipedia.org/wiki/Oracle_Corporation)).[[8]](https://en.wikipedia.org/wiki/MySQL#cite_note-sunacquire-8) In 2010, when [Oracle acquired Sun](https://en.wikipedia.org/wiki/Acquisition_of_Sun_Microsystems_by_Oracle_Corporation), Widenius [forked](https://en.wikipedia.org/wiki/Fork_(software_development)) the [open-source](https://en.wikipedia.org/wiki/Open-source) MySQL project to create [MariaDB](https://en.wikipedia.org/wiki/MariaDB" \o "MariaDB).[[9]](https://en.wikipedia.org/wiki/MySQL#cite_note-9)

MySQL has stand-alone clients that allow users to interact directly with a MySQL database using SQL, but more often, MySQL is used with other programs to implement applications that need relational database capability. MySQL is a component of the [LAMP](https://en.wikipedia.org/wiki/LAMP_(software_bundle)) [web application](https://en.wikipedia.org/wiki/Web_application) [software stack](https://en.wikipedia.org/wiki/Software_stack) (and [others](https://en.wikipedia.org/wiki/List_of_AMP_packages)), which is an acronym for [*Linux*](https://en.wikipedia.org/wiki/Linux)*,*[*Apache*](https://en.wikipedia.org/wiki/Apache_HTTP_Server)*, MySQL,*[*Perl*](https://en.wikipedia.org/wiki/Perl)*/*[*PHP*](https://en.wikipedia.org/wiki/PHP)*/*[*Python*](https://en.wikipedia.org/wiki/Python_(programming_language)). MySQL is used by many database-driven web applications, including [Drupal](https://en.wikipedia.org/wiki/Drupal" \o "Drupal), [Joomla](https://en.wikipedia.org/wiki/Joomla" \o "Joomla), [phpBB](https://en.wikipedia.org/wiki/PhpBB" \o "PhpBB), and [WordPress](https://en.wikipedia.org/wiki/WordPress). MySQL is also used by many popular [websites](https://en.wikipedia.org/wiki/Website), including [Facebook](https://en.wikipedia.org/wiki/Facebook),[[10]](https://en.wikipedia.org/wiki/MySQL#cite_note-10)[[11]](https://en.wikipedia.org/wiki/MySQL#cite_note-11) [Flickr](https://en.wikipedia.org/wiki/Flickr),[[12]](https://en.wikipedia.org/wiki/MySQL#cite_note-12) [MediaWiki](https://en.wikipedia.org/wiki/MediaWiki),[[13]](https://en.wikipedia.org/wiki/MySQL#cite_note-13) [Twitter](https://en.wikipedia.org/wiki/Twitter),[[14]](https://en.wikipedia.org/wiki/MySQL#cite_note-14) and [YouTube](https://en.wikipedia.org/wiki/YouTube).[[15]](https://en.wikipedia.org/wiki/MySQL#cite_note-15)

MySQL is written in [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [C++](https://en.wikipedia.org/wiki/C%2B%2B). Its SQL parser is written in [yacc](https://en.wikipedia.org/wiki/Yacc" \o "Yacc), but it uses a home-brewed [lexical analyzer](https://en.wikipedia.org/wiki/Lexical_analysis).[[16]](https://en.wikipedia.org/wiki/MySQL#cite_note-16) MySQL works on many [system platforms](https://en.wikipedia.org/wiki/System_platform), including [AIX](https://en.wikipedia.org/wiki/AIX_operating_system), [BSDi](https://en.wikipedia.org/wiki/BSD/OS" \o "BSD/OS), [FreeBSD](https://en.wikipedia.org/wiki/FreeBSD), [HP-UX](https://en.wikipedia.org/wiki/HP-UX), [ArcaOS](https://en.wikipedia.org/wiki/ArcaOS" \o "ArcaOS), [eComStation](https://en.wikipedia.org/wiki/EComStation" \o "EComStation), [IBM i](https://en.wikipedia.org/wiki/IBM_i), [IRIX](https://en.wikipedia.org/wiki/IRIX), [Linux](https://en.wikipedia.org/wiki/Linux), [macOS](https://en.wikipedia.org/wiki/MacOS" \o "MacOS), [Microsoft Windows](https://en.wikipedia.org/wiki/Microsoft_Windows), [NetBSD](https://en.wikipedia.org/wiki/NetBSD" \o "NetBSD), [Novell NetWare](https://en.wikipedia.org/wiki/Novell_NetWare), [OpenBSD](https://en.wikipedia.org/wiki/OpenBSD" \o "OpenBSD), [OpenSolaris](https://en.wikipedia.org/wiki/OpenSolaris" \o "OpenSolaris), [OS/2](https://en.wikipedia.org/wiki/OS/2) Warp, [QNX](https://en.wikipedia.org/wiki/QNX), [Oracle Solaris](https://en.wikipedia.org/wiki/Solaris_(operating_system)), [Symbian](https://en.wikipedia.org/wiki/Symbian" \o "Symbian), [SunOS](https://en.wikipedia.org/wiki/SunOS), [SCO OpenServer](https://en.wikipedia.org/wiki/SCO_OpenServer), SCO [UnixWare](https://en.wikipedia.org/wiki/UnixWare), Sanos and [Tru64](https://en.wikipedia.org/wiki/Tru64_UNIX). A port of MySQL to [OpenVMS](https://en.wikipedia.org/wiki/OpenVMS) also exists.[[17]](https://en.wikipedia.org/wiki/MySQL#cite_note-17)

The MySQL server software itself and the client libraries use [dual-licensing](https://en.wikipedia.org/wiki/Dual_license) distribution. They are offered under [GPL](https://en.wikipedia.org/wiki/GNU_General_Public_License) version 2, or a proprietary license.[[18]](https://en.wikipedia.org/wiki/MySQL#cite_note-18)

Support can be obtained from the official manual.[[19]](https://en.wikipedia.org/wiki/MySQL#cite_note-19) Free support additionally is available in different IRC channels and forums. Oracle offers paid support via its MySQL Enterprise products. They differ in the scope of services and in price. Additionally, a number of third party organisations exist to provide support and services.

MySQL has received positive reviews, and reviewers noticed it "performs extremely well in the average case" and that the "developer interfaces are there, and the documentation (not to mention feedback in the real world via Web sites and the like) is very, very good".[[20]](https://en.wikipedia.org/wiki/MySQL#cite_note-20) It has also been tested to be a "fast, stable and true multi-user, multi-threaded SQL database server".[[21]](https://en.wikipedia.org/wiki/MySQL#cite_note-21)

MySQL was created by a Swedish company, [MySQL AB](https://en.wikipedia.org/wiki/MySQL_AB), founded by [Swedes](https://en.wikipedia.org/wiki/Swedes) [David Axmark](https://en.wikipedia.org/wiki/David_Axmark), Allan Larsson and [Finnish](https://en.wikipedia.org/wiki/Finns) [Michael "Monty" Widenius](https://en.wikipedia.org/wiki/Michael_Widenius). Original development of MySQL by Widenius and Axmark began in 1994.[[22]](https://en.wikipedia.org/wiki/MySQL#cite_note-22) The first version of MySQL appeared on 23 May 1995. It was initially created for personal usage from [mSQL](https://en.wikipedia.org/wiki/MSQL" \o "MSQL) based on the low-level language [ISAM](https://en.wikipedia.org/wiki/ISAM), which the creators considered too slow and inflexible. They created a new [SQL](https://en.wikipedia.org/wiki/Structured_Query_Language) interface, while keeping the same [API](https://en.wikipedia.org/wiki/Application_programming_interface) as mSQL. By keeping the API consistent with the mSQL system, many developers were able to use MySQL instead of the (proprietarily licensed) mSQL antecedent.[[23]](https://en.wikipedia.org/wiki/MySQL#cite_note-23)

**Chapter 10**

**Chapter 11**

**BIBILOGRAPHY**

* [www.wikipedia.com](http://www.wikipedia.com)
* [www.javapoint.com](http://www.javapoint.com)
* [www.w3cschools.com](http://www.w3cschools.com)
* [www.geeksforgeeks.com](http://www.geeksforgeeks.com)
* [www.interviewgit.com](http://www.interviewgit.com)
* [www.simplilearn.com](http://www.simplilearn.com)
* [www.scaler.com](http://www.scaler.com)