* **What is volatile memory and non-volatile memory?**

**=> -Volatile memory is a type of memory in which data contained in the memory is lost whenever the power is turned off.**

**RAM is a example of volatile memory.**

**-Non-volatile memory is a type of memory in which data contained in the memory remains intact and will not get lost whenever the power is turned off.**

**ROM is a example of non-volatile memory.**

* **What is Low-level, Mid-level and High-level language?**

**=> -The language which is easily readable and understandable for computer, is known as low-level language / machine language.**

**-The language which is less readable and less understandable, known as mid level language / assembly language.**

**Example: microprocessor instruction set**

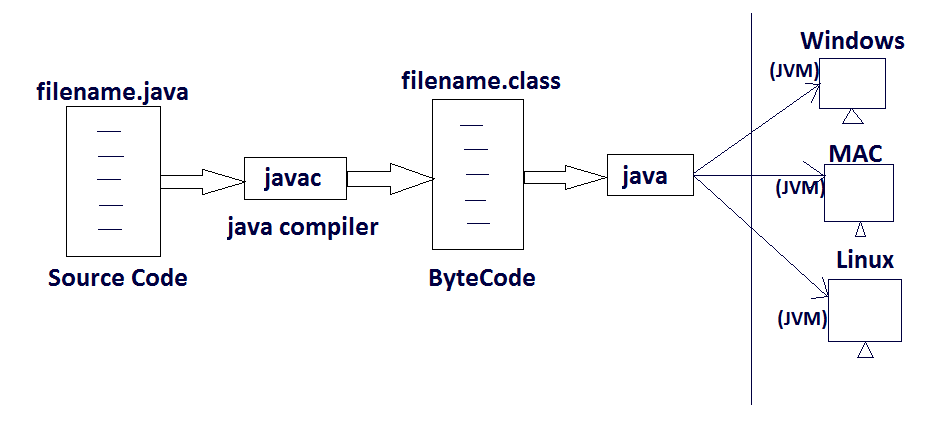
**-The language which is highly readable and understandable by human, is known as high-level language.**

**Example: java, c#, python etc.**

* **What is native machine language?**

**=> Machine level language along with the system architecture, is known as native machine language.**

* **Explain program execution architecture.**

**=> **

**A file where instructions are written using java programming language is known as source code (i.e .java file).**

**Java source code converted into an intermediate language, called as bytecode with the help of java compiler by javac command (i.e .class file).**

**The bytecode can be executed with the help of JVM by java command.**

* **Why java is called platform independent language?**

**=> Java is called platform independent programming language as it works on the principle of “Compile Once, Run Anywhere” i.e, CORA.**

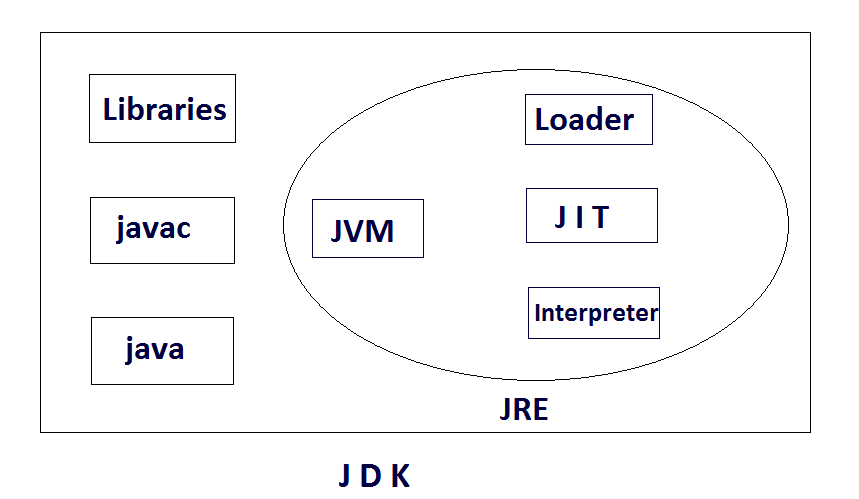
**If we have written some code in java and we have compiled that code, we can execute that code on any platform. The only thing we need is to install the corresponding JVM (Java Virtual Machine).**

**Java code can be run on multiple platforms e.g, Windows, Linus, MacOS etc. Java source code is compiled by the compiler (javac) and converted into ByteCode. This ByteCodes are platform independent code because it can be run on multiple platforms so this is “Write Once, Run Anywhere” i.e, WORA.**

* **What is JDK?**

**=> JDK is the acronym of Java Development Kit.**

**JDK consist of many components which are required for compilation and execution of java program.**

****

**It includes:**

**Libraries: Consists of predefined java programs.**

**Javac: used to compile the java source code to bytecode.**

**Java: used to create or invoke the JRE.**

**JRE: Java Runtime Environment is an environment to created for successful execution of the java bytecode.**

**A JRE consist of some components, such as:**

**JVM: Java Virtual Machine, which is needed for execution of a bytecode.**

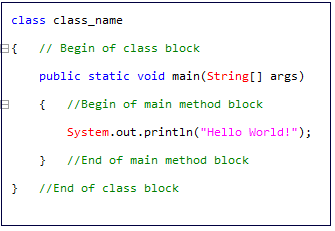
**JIT: Just In Time compiler, is used to convert the bytecode into native machine language.**

**Class Loader: it is used to load the class file from secondary memory to execution area.**

**Interpreter: It is used to interpret the instruction, generated instruction by instruction.**

* **Structure of java program?**

**=>**

****

**Class, which contains main method is the only mandatory class, that we have to define in our java program for execution.**

**Class\_name, which is used in java file is given as name for the class file generated by java. (i.e, Class\_name.java => javac => Class\_name.class)**

**Public, is an access specifier & access modifier in java. We declared main as public so that all other classes can get access to it.**

**Static, we declare main method as static so that it gets invoked without creating an instance of a class. It can be directly invoked by the class.**

**Void, represent the main method does not return any value.**

**Main, method is a entrypoint where a program starts its execution.**

**String[] args, String is a predefined built-in class in java. Here we declare an array of objects of String class.**

**System, predefined built in class.**

**Out, is the output stream object.**

**Println, method belongs to System class, used to print an output.**

* **Execution starts from?**

**=> When a java class is executed, first main method is called.**

**To execute a program in java, the JVM needs some starting point and that point in the program is the 'main' method.**

**There is a reason for that method to be static.**

**The main method is static so that it gets invoked without creating an instance of a class.**

**It can be directly invoked by the class.**

* **What is statically typed and dynamically typed language?**

**=> Statically typed language perform type checking at compile time. E.g, Java etc.**

**Dynamically typed language perform type checking at run time. E.g, Python etc.**

* **What is value?**
* **what is an identifier?**

**=> The name given by the java programmer for the elements of java programming languages, such as:**

**i) variable**

**ii) method**

**iii) class**

**iv) interface**

**-are known as identifier.**

* **Rules of identifier.**

**=> -identifiers can be alphanumeric,**

**-identifiers can not be started with numeric values as number,**

**-we can not use white space with identifier,**

**-'$' & '\_' are the only two special characters which are allowed with identifiers in java,**

**-an identifier can be started with '$' & '\_' (i.e $test, \_test)**

**-single '\_' is an invalid identifier (but it was valid till jdk 1.7)**

**-single '$' is an valid identifier**

**-keyword can not be used as identifier**

* **Uses of “+” operator in java?**

**=> “+” operator in java has two behaviors, such as:**

1. **Addition:**

**-adding numeric values or charector values is known as Addition operator.**

1. **Concatenation:**

**-merging of two values is known as Concatenation operator.**

* **What is a datatype? different premitive datatypes available in java.**

**=># A datatype is a named memory allocation, which are used to create a block of memory to store date.**

**A datatype is a predefined keyword in java, that specifies the size and type of values that can be stored in an identifier.**

**Datatypes in java are classified in to two types, i.e,**

1. **Primitive: Primitive datatypes are predefined datatypes of java.**

**1. Number**

**->Integer**

**-byte (1byte)**

**-short (2bytes)**

**-int (4bytes)**

**-long (8bytes)**

**->Floating**

**-float (4bytes)**

**-double (8bytes)**

**2. Charector**

**-char (2bytes)**

**3. Boolean**

**-boolean (1bit)**

1. **Non-primitive: Non-primitive datatypes are user defined/derived datatypes of java.**
2. **Classes**
3. **Interfaces**
4. **Arrays**

* **What is a variable?**

**=> A variable is a named memory location which is used to store values or data of the program.**

**A variable is a piece of memory that can contain a data value. A variable thus has a datatype.**

**We can achieve reuse ability of data with the help of variable.**

* **What do you mean by variable declaration? why do we need it?**

**=># It is a statement which is written to specify the type of data that should be stored in the given variable.**

**Syntax : datatypes Variablename ;**

**#Variable declaration statement is used to create a block of memory to store particular type of value.**

**Variable / identifier is the name given to the block of memory created.**

* **what is variable initialization? Why do we need it?**

**=> # Initialization is a statement which is written to store the value using assignment operator in a variable.**

**Syntax : variable = value ;**

**#It is used to initialize a memory with a value.**

* **What is variable utilization? Why do we need it?**

**=> # Utilization is a statement which is written to use the values of the variable to perform same operation.**

**# The utilization of the value present in the variable changes from one requirement to another requirement.**

* **what are the types of variables are there in java and explain all.**

**=> There are two types of variable, i.e,**

**1. Local variable**

**-A variable declare within a method definition (method signature and method body) inside a class is known as local variable.**

**-local variable don’t have any default values.**

**2. Global variable**

**-A variable declare outside a method block inside a class is known as global variable.**

**-global variables have default value depends on the datatype.**

**i)Static variable**

**-A variable declare inside a class, outside a method block and prefix with the static keyword is known as static variable.**

**ii)Non static variable**

**-A variable declare inside a class, outside a method block and without the static keyword is known as non-static variable.**

* **what is a keyword?**

**=> The predefined compiler aware words are keywords.**

**The keywords are reserved words which have predefined meaning in java programming language.**

**i.e, class, static, public, void, true, false etc**

**Rules:**

**Keywords are always written in lower case.**

* **String is a datatype in java.**

**=> String is a predefined, built in class in java. Here we declare an array of objects of string class. It helps us to take command line argument. Though string is a array of charactors so it is not a premitive datatype.**

* **Can we have a local variable with a same name,within a same block?**

**=> No, two local variables with a same name can not be possible inside a same block.**

* **What are conditional statements? Explain different types of conditional statesment.**

**=> #In java, the statements that allow us to check condition** **is known as conditional statements.**

**# 5types:**

1. **Simple if:**
2. **If-else**
3. **Else-if ladder**
4. **Nested if**
5. **switch**

* **What is loops? Explain different types of loop.**

**=>#A loop statements allows us to execute a statements or group of statements multiple times.**

**#types:**

1. **While loop**
2. **Do-while loop**
3. **For loop**

* **What is a break? why do we use?**

**=># Break is a control transfer statement, which can be used either in switch or loop block.**

**Break should be the last statement to be used in a block.**

**# Break statement transfer the control outside the current loop or switch block.**

* **what is continue? why do we use?**

**=> # Continue is a control transfer statement, which can be used either in loop block.**

**Continue should be the last statement to be used in a block.**

**# Continue statement transfer the control to the begining of the current loop block.**

* **diff bw break and continue?**

**=>**

* **what is a method, explain method signature?**

**=> # Method is a block of instruction, which perform a specific activity as a task.**

**# Method name followed by the formal arguments of a method is considered as a method signature.**

* **what is method precidence?**
* **what is recurssion?**

**=> A method calling itself is known as recursion.**