1. **Platform independency**

* A programming language is said to be platform independent if it can be executed in any operating system, in case of java
* the java compiler will convert the code written in human understandable language to byte code, bytecode is sent to Java virtual machine (JVM)
* which resides in the RAM of any operating system. JVM recognizes the platform it is on and converts the bytecodes into machine code based on Operating system, It runs on the logic of “Write once, run anywhere”
* Hence java is called platform independent language.

1. **Differences Between java and C language**

|  |  |
| --- | --- |
| JAVA | C |
| java is Object oriented programming language  Java is high level language.  Java has a feature of threading  Java does not support pointers.  it is automatically managed by a garbage collector  Java memory allocation can be done by the new keyword | c is Procedural Programming Language  C is middle level language  C does not support threading  C supports pointer  C garbage collection needs to managed manually  C language memory allocation can be done by malloc |

1. **Syntax of main**

* A **main() method in java** is an entry point to start the execution of a program
* Every Java application has at least one [class](https://www.scientecheasy.com/2020/07/java-classes-objects.html/) and at least one main method
* classes needs to have a main() method to get things started.
* Therefore, java main() method is the starting place of your program

Syntax=> public static void main(String[] args)

* **public:** it is an access specifier The public modifier makes it accessible from anywhere in the application.
* **static:** The [static modifier](https://www.scientecheasy.com/2020/06/java-static-variable.html/) makes it a class method so that it can be called using the class name without [creating an object of the class](https://www.scientecheasy.com/2020/06/how-to-create-object-in-java.html/)
* **void:** The return type of the main method is void which means that it does not return a value to its caller.  You must specify void when you declare the main method.
* **main:** It is the name of a method where execution will start. In Java, the main method is called by JVM.
* **String[ ] args:** The main method accepts one argument of type String array (String[ ]). The square brackets [ ] represent the array of strings that is passed as an argument to this method
* args is the name of its parameter. You can use any parameter name as you wish.
* If a program does not contain the main method in a class, Java compiler will compile it but cannot run it.

1. **Java Application is Robust**

* [Strong type](http://en.wikipedia.org/wiki/Strong_typing) checking mechanism of Java also helps in making Java Robust
* Compiler checks the program that whether there are any errors or not and interpreter ascertains about the [run time error](http://en.wikipedia.org/wiki/Run_time_%28program_lifecycle_phase%29) and makes the system secure from crash.
* Java also does not support pointers which are the major & constant source of mistakes in other languages
* multiple inheritance can’t be exhibited in Java making it a strong & robust language
* **Robust code**” means that your program takes into consideration all possibilities, and that there is no existence of error – all situations are handled by the code and result in valid state. So now we understand that why java is called a robust language.

1. **Local Variables**

* Local variables are declared in methods, constructors, or blocks.
* Local variables are created when the method, constructor or block is entered and the variable will be destroyed once it exits the method, constructor, or block.
* Access modifiers cannot be used for local variables.
* There is no default value for local variables, so local variables should be declared and an initial value should be assigned before the first use.

1. **architectural neutral**

* Java application runs the same bytecodes regardless of any environment (Operating System)
* the compiler generates an architecture-neutral object file format
* An architecture-neutral object file format meaning that compiled Java code (bytecode) can run on many processors given the presence of a JVM
* The JVM is the main component of making the java a architectural neutral part
* The JVM takes your compiled platform-neutral byte code and interprets it to run platform-specific machine code.

1. Portability

* Portability refers to the ability to run a program on different machines
* There are two aspects of portability

Potable=platform independent + architectural neutral

* The primitive data type memory is fixed

1. Unicode

* Unicode is a computing industry standard designed to consistently and uniquely encode characters used in written languages throughout the world
* Unicode is a 16-bit character encoding standard and is capable to represent almost every character of well-known languages of the world.
* Before Unicode=
* ASCII-US

**System software and application Software**

* System Software is a set of programs that control and manage the operations of computer hardware. It also helps application programs to execute correctly
* System software makes the operation of a computer more fast, effective, and secure. Example: Operating system, programming language, Communication software, etc
* Application Software is a program that does real work for the user. It is mostly created to perform a specific task for a user.
* Application Software acts as a mediator between the end-user and System Software.
* You can also install multiple Application Software on a single System Software

1. **Compiler and interpreter**

* A Java compiler is a program that takes the text file work of a developer and [compiles](https://whatis.techtarget.com/definition/compiler) it into a platform-independent [Java](https://www.theserverside.com/definition/Java) file. Java compilers include the Java Programming Language Compiler (javac)
* 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)
* 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.
* At run time, the compiler first parses (analyzes) all of the language statements syntactically and then, in one or more successive stages or "passes,” builds the output code
* Generally, Java compilers are run and pointed to a programmer’s code in a text file to produce a [class](https://whatis.techtarget.com/definition/class) file for use by the Java virtual machine ([JVM](https://www.theserverside.com/definition/Java-virtual-machine-JVM)) on different platforms

**Interpreter**

* **Java interpreter** is a computer program (system software) that implements the JVM. It is responsible for reading and executing the program. It is designed in such a way that it can read the source program and translate the source code instruction by instruction. **It converts the high-level program into assembly language** (machine language).
* To convert the byte code into machine code, we deploy the .class file on the [Java Virtual Machine (JVM)](https://www.javatpoint.com/jvm-java-virtual-machine). The JVM converts that code into machine code using the Java interpreter. The JVM uses the interpreter at runtime, after that it execute the code on the host machine
* Java interpreter converts or translates the bytecode into the machine-understandable format i.e. machine code, after that the machine code interacts with the operating system

1. **Assembler**

* An assembler is a type of computer program that interprets software programs written in assembly language into machine language, code and instructions that can be executed by a computer.
* An assembler enables software and application developers to access, operate and manage a computer's hardware architecture and components.
* An assembler is sometimes referred to as the compiler of assembly language. It also provides the services of an interpreter.