**Java Features:-**JAVA has provided the following features.

1. **Simple**
2. **Object Oriented**
3. **Platform independent**
4. **Arch Neutral**
5. **Portable**
6. **Robust**
7. **Secure**
8. **Dynamic**
9. **Distributed**
10. **Multi-Threaded**
11. **Interpretive**
12. **High Performance**

**1) Simple:**

1)Java is using all the simplified syntaxes from C and C++.

2)Java is a simple programming language, because, Java applications will take less memory and less execution time.

3) Java has removed all most all the confusion-oriented features like pointers, multiple inheritance..... etc

**2) Object Oriented:**

Java is an object-oriented programming language, because, JAVA is able to store data in the form of Objects only.

**3) Platform Independent: -**Java is platform independent programming Language, because, Java allows its applications to compile on one operating system and execute on another operating system.

**4) Arch Neutral:-** Java is an Arch Neutral Programming language, because, Java allows its applications to compile on one H/W Arch and to execute on any H/W Arch.

**5) Portable:-** Java is a portable programming language, because, JAVA is able to run its applications under all the operating systems and under all the H/W Systems.

**6) Robust:** Java is a Robust programming language because Java is having very good memory management systems like garbage collection and dynamic memory allocation.

**7) Secure:** Java is a very good Secure programming language because of byte code.

To provide explicit security for the Java applications we are having a very good predefined library in the form of ***java.security*** package.

**8) Dynamic:-**Java is dynamic technology it follows dynamic memory allocation (at runtime the memory is allocated).

**9) Distributed:-**

By using java it is possible to develop distributed applications.

**10) Multi-Threaded:-**Java provide support of multithreading.

**11) Interpretive**::JAVA is both a compilative programming language and an Interpretive programming language.

**12) High Performance:-**JAVA is a high-performance programming language due to its rich set of features like Platform independence, Arch Neutral, Portable, Robust, and Dynamic,......

**Difference between C and Java Language.**

|  |  |
| --- | --- |
| C language | JAVA Language |
| C is a middle-level language and C is a structural and procedure-oriented programming language. | **Java is a high-level language and it is an object-oriented programming language** |
| It is compiled language and after compilation, it generates .exe file. | **It is compiled and interpreter language and after compilation, it generates a .class file.** |
| We can save the C language source file with .c extension. | **We can save the Java language source file with .java extension.** |
| It translates the code into machine language so that the machine can understand the code, so it is more secure. | **It translates the code into a bytecode that is executed by the JVM so, it is more secure.** |
| C is a platform-dependent language. | **Java is a platform-independent language.** |
| C language does not support garbage collection, exception handling, and multithreading | **Java language support garbage collection, exception handling, and multithreading.** |
| It supports the concept of the pointer. | **It does not support the concept of pointers because of security.** |
| There are 32 reserved keywords in C. | **There are 53 reserved keywords in Java.** |

**public modifier:-**

**The public modifier is applicable for variables, methods, and classes.**

**All packages are able to access public members.**

**Default modifier:-**

**It is applicable for variables, methods, and classes.**

**We are able to access default members only within the package and it is not possible to access outside the package.**

**Default access is also known as package-level access.**

**If we have not specified any modifier then it’s treated as default.**

**private modifier:-**

**private modifier applicable for methods and variables.**

**We are able to access private members only within the class and it is not possible to access them even in child classes.**

**protected modifier:-**

**The protected modifier is applicable for variables, and methods.**

**We are able to access protected members within the package and it is possible to access outside packages also but only direct child classes.**

**But in the outside package, we can access protected members only by using child reference. If we try to use parent reference, we will get compile time error.**