C++ and Java comparison

There are many differences and similarities between the C++ programming language and Java. A list of top differences between C++ and Java are given below:

Comparison Index	**C++**	**Java**
Platform-independent	C++ is platform-dependent.	Java is platform-independent.
Mainly used for	C++ is mainly used for system programming.	Java is mainly used for application programming. It is widely used in window, web-based, enterprise and mobile applications.
Design Goal	C++ was designed for systems and applications programming. It was an extension of C programming language.	Java was designed and created as an interpreter for printing systems but later extended as a support network computing. It was designed with a goal of being easy to use and accessible to a broader audience.
Goto	C++ supports the goto statement.	Java doesn't support the goto statement.
Multiple inheritance	C++ supports multiple inheritance.	Java doesn't support multiple inheritance through class. It can be achieved by interfaces in java.
Operator Overloading	C++ supports operator overloading.	Java has method overloading but no operator overloading. The String class does use the + and += operators to concatenate strings and String expressions use automatic type conversion, but that's a special built-in case.
Pointers	C++ supports pointers. You can write pointer program in C++.	Java supports pointer internally. However, you can't write the pointer program in java. It means java has restricted pointer support in java.
Compiler and Interpreter	C++ uses compiler only. C++ is compiled and run using the	Java uses compiler and interpreter both. Java source code is converted into bytecode

	compiler which converts source code into machine code so, C++ is platform dependent.	at compilation time. The interpreter executes this bytecode at runtime and produces output. Java is interpreted that is why it is platform independent.
Call by Value and Call by reference	C++ supports both call by value and call by reference.	Java supports call by value only. There is no call by reference in java.
Structure and Union	C++ supports structures and unions.	Java doesn't support structures and unions.
Thread Support	C++ doesn't have built-in support for threads. It relies on third-party libraries for thread support.	Java has built-in thread support.
Documentation comment	C++ doesn't support documentation comment.	Java supports documentation comment (/** ... */) to create documentation for java source code.
Virtual Keyword	C++ supports virtual keyword so that we can decide whether or not override a function.	Java has no virtual keyword. We can override all non-static methods by default. In other words, non-static methods are virtual by default.
unsigned right shift >>>	C++ doesn't support >>> operator.	Java supports unsigned right shift >>> operator that fills zero at the top for the negative numbers. For positive numbers, it works same like >> operator.
Inheritance Tree	C++ creates a new inheritance tree always.	Java uses a single inheritance tree always because all classes are the child of Object class in java. The object class is the root of the inheritance tree in java.
Hardware	C++ is nearer to hardware.	Java is not so interactive with hardware.
Object-oriented	C++ is an object-oriented language. However, in C language, single root hierarchy is not possible.	Java is also an object-oriented language. However, everything (except fundamental types) is an object in Java. It is a single root hierarchy as everything gets derived from java.lang.Object.
Relationship	No strict relationship between class names and filenames. In C++, header files and	The strict relationship is enforced, e.g., the source code for class PayRoll has to be in PayRoll.java.

	implementation files are used for specific class.	
Input mechanism	I/O statements use in and cout, e.g., in » x; cout« y;	I/O input mechanism is a quite complex as it reads one byte at a time (System.in). Output is easy, e.g. System.out.println(x);
compatibility with other languages	Compatible with C source code, except some exceptional cases.	No backward compatibility with any previous language. The syntax is influenced by C/C++.
Access control and object protection	Flexible model with constant protection available	The cumbersome model encourages week encapsulation.
Concept	Write once compile anywhere	Write once run anywhere everywhere
Exception handling	In C++, you may exclude the attempt/get regardless of whether the capacity throws an exemption.	**Exception handling** is different because there are no destructors. In Java, try/catch must define if the function declares that it may throw an exception.

C++ Example

File: main.cpp

```
#include <iostream>
using namespace std;
int main() {
  cout << "Hello C++ Programming";
  return 0;
}
```

Java Example

File: Simple.java

```
class Simple{
    public static void main(String args[]){
     System.out.println("Hello Java");
    }
}
```