

C++ vs Java vs C#

In this lesson, we'll get to know the difference between the three popular object-oriented languages, i.e., C++, Java, and C#.

WE'LL COVER THE FOLLOWING ^

- Java and C# (Pure Object-Oriented Language)
- Explanation of Code Snippets

Java and C# (Pure Object-Oriented Language)

Java emerged as an object-oriented language in the **90's**. In **Java**, even the main function has to be inside a class, unlike **C++**. Although the design goals for creating Java were different than that of **C++**, **Java** had the leverage of overcoming and rectifying all object-oriented issues the programmers identified in **C++**, such as the **diamond problem**. **C#** was introduced by Microsoft in 2001 as a Java competitor. It is also a pure object-oriented language and the syntax is much similar to that of Java.

Let's look at the implementation of **calculateSum** function in all three languages:



```
#include<iostream>
using namespace std;

int calculateSum(int p1, int p2){
    int sum = p1 + p2;
    return sum;
}

int main() {
    cout<<calculateSum(2,3); // calculates sum of given numbers
}
```



Explanation of Code Snippets

As you can see in codes above, in case of **C++** we simply wrote the `calculateSum` subroutine without making any class and called it in our `main` function just like in a procedural language. However, when you look at the codes of **Java** and **C#**, both are a bit more complex and extensive. If you would try removing some keywords like `Class` and `static` in these codes, it would generate an error and the codes won't compile, for **Java** and **C#**. So if you intend to write even a simple add/sum function in **C#** or **Java**, you cannot do so in any other simpler way. Hence the choice of language is programmer and problem dependent.

Java and **C#** have emerged as an object-oriented language **C++** still fills up a huge job market as well as is a primary building block in academics. Having said this, there are still some people who are skeptical about the use of object-oriented language. They believe it makes the overall program size more complex. Well, as we have also shown in our example above, even writing a simple sum function became an extensive task. They also argue that the details of object-oriented programming, its syntax, and peculiarities, are difficult for the programmer to learn, hence resulting in a relatively slow learning curve. Nonetheless, the object-oriented paradigm is a well-established programming practice and this course is all designed to make you familiar with it.

Before we formally start our course, let's take a short **C++** quiz to test your basic knowledge. Don't worry we don't grade you :)