

Analysis of C and Java's Similarities and Differences

C and Java languages remain popular today as they were in the past. Languages are used with this program in many areas, especially desktop software. Both are used for similar applications even if the Coding Standard is different. In this article, I will compare C and Java programming languages.

Although they are similar languages, it is aimed to compare their languages with two different procedural and object-oriented programs. When I compared C and Java Programming languages, I was faced with the problem of deciding what to compare as it was not clear what constitutes each language and what constitutes its environment. It is not enough to say that each language is specified in its documentation, as the designers of the two languages have different philosophies about what is pertaining to a language. C is a structured and procedure-oriented programming language while Java is an object-oriented programming language. These structural features highlight the advantages and disadvantages of the C and Java programming languages. In this article, I will examine them in detail based on their structural differences in C and Java programming languages..

Syntax

First of all, I covered the syntax. Keywords are reserved words in programming language, these words are detected by the compiler. There are 32 keywords in C language. There are exactly 67 keywords in the Java language. Some keywords are different. Some keywords in C language but not in Java are: Union, Struct. On the contrary, some of the keywords in Java but not in C. These are: Class, Interface, New, Abstract . However, some keywords are the same in order to increase intelligibility and readability as well as writability in the code. For example, to express a numeric value, the expression "int", which comes from the root of the word integer, is used in both programming languages. Variables expressed with the same keywords may not be of the same sizes in programming languages. While the char expression used in defining characters in C language generally takes up 8 bit ASCII, in Java it takes up 16 bit UNICODE. Punctuation marks are generally the same in C and Java languages. Semicolon, Colons, Commas, Apostrophes, Quotation marks, Braces, Brackets, Parentheses are used as punctuation marks. As you know, there are libraries in programming languages to access these libraries, there are expression differences in C and Java, but they mean functionally similar things. Libraries are called with #include in C language, but they are called with import in Java. Calling the functions of the library we added in our code differs in C and Java languages. For example in C language #include "math.h" x = pow(5,3); In Java, x = Math.sqrt(2.2);Printing an expression as output to the terminal differs in C and Java. printf("sum = %d", x); while expressing as System.out.println("sum = " + x);.

In C and Java languages, there is a table 1.1 to write to get the output of hello.

hello, world	<pre>#include<stdio.h> int main(void) { printf("Hello\n"); return 0; }</pre>	<pre>public class HelloWorld { public static void main(String[] args) { System.out.println("Hello"); } }</pre>
--------------	--	--

Defining constants is different in Java and C languages. In the Java programming language, we define it as a constant with the final keyword that we add in front of the data type. In C language, define or const is used. #define is a preprocessor directive. It is used to define micro substitution Its syntax is - :,#define token value It should not be enclosed with a (;) semicolon. Whereas, Constants are used to make variables constant such that never change during execution once defined. Constants are also called literals. Its syntax is -: const type constant_name; It should be enclosed with a (;) semicolon

As I said before, C has procedural paradigm. However, Java has object-oriented programming paradigm. OOP is a programming paradigm based on the concept of "objects", which can contain data and code: data in the form of fields (often known as attributes or properties), and code, in the form of procedures (often known as methods). OOP brings some new concepts that are not in C programming. These are called “Encapsulation”, “Polymorphism”, “Inheritance” and “Abstraction”

Encapsulation is one of the four fundamental OOP concepts. The other three are inheritance, polymorphism, and abstraction. Encapsulation in Java is a mechanism of wrapping the data (variables) and code acting on the data (methods) together as a single unit. In encapsulation, the variables of a class will be hidden from other classes, and can be accessed only through the methods of their current class. Therefore, it is also known as data hiding.

Polymorphism is the ability of an object to take on different forms. In Java, polymorphism refers to the ability of a class to provide different implementations of a method, depending on the type of object that is passed to the method.

Inheritance is one of the core concepts of object-oriented programming (OOP) languages. It is a mechanism where you can derive a class from another class for a hierarchy of classes that share a set of attributes and methods.

Abstraction is one of the key concepts of object-oriented programming (OOP) languages. Its main goal is to handle complexity by hiding unnecessary details from the user. That enables the user to implement more complex logic on top of the provided abstraction without understanding or even thinking about all the hidden complexity.

Operators

Operators are very important to the programming language. We take certain actions with operators.

There are operators common to both Java and C languages. Arithmetic operators, Comparison Operators and Logical Operators are some of them. These are almost the same. Arithmetic Operators +, -, *, /, %, ++, --. Comparison Operators ==, !=, >, <, >=, <=. Logic Operators &&, ||, !. There are several operators in C that are not in the Java programming language. & returns the address of the variable, * indicates the variable. As in every programming language, keywords cannot be used in definitions. Both languages have their own rules while defining. You can see more detail below.

	Unary Operators	Arithmetic Operators	Shift Operators	Relational Operators
C	+ - ! ~ ++ -- * & sizeof	+ - * / %	<< >>	< <= > >= == !=
Java	+ - ! ~ ++ --	+ - * / %	<< >> >>>	< > <= >= instanceof == !=

	Bitwise Operators	Logical Operators	Ternary Operators	Assignment Operators	Comma Operator	Some Extra Operators
C	& ^	&&	? :	= += -= *= /= %= >>= <<= &= ^= =	,	() [] . - >
Java	& ^	&&	? :	= += -= *= /= %= &= ^= = <<= >>= >>>=	,	

Semantics

I wrote the semantics in this part. C and Java languages have common data types as in other programming languages. I classified and analyzed the data types according to their properties. Integer, short, float, double and long data types are used to perform numerical operations. The operation is performed by selecting the appropriate data type according to the purpose of use.

For example, if we want to use a number of 2 or 4 bytes, we use short or integer. We use floats for non-integer numbers and the size of the float is 4 bytes. We use double for numbers that are not integers but larger. But if we are going to use larger numbers, we use the long data type. To use data types in Java we use what we use in C, but since Java is an object-oriented programming language, it offers certain advantages. In order to use data types more efficiently, there are special classes for those data types. While numeric data types are normally kept in the stack, the data types we create as classes are kept in the heap. One of its advantages is that it makes conversion processes easier. For example, let's see how we translate a number that we created as a String in c and java languages;

C	Java
<pre>#include <stdio.h> #include <stdlib.h> #include <math.h> int main(){ char sayi[6] = "125469"; int size = 6; int result=0; int x=6; for(int i=0;i<size;++i){ int number = sayi[i] - 48; result += number*pow(10,x-1); --x; } printf("%d",result); return 0; }</pre>	<pre>public static void main(String args[]) { String sayi = new String("123456"); int number = Integer.parseInt(sayi); System.out.println(number); }</pre>

Another data type is char. We keep a single character with char as usage is common in c and java languages. Next we have the String data type. String keyword does not exist in c language, it is a java specific data type. To create a string expression in C language, a char array is created. We may need to show where the char array we created in C language ends with '\0'. In Java, String is a class. This class provides us with many methods.

C	Java
<pre>char *c = (char *)malloc(50 *sizeof(char)); strcpy(c,"12345"); char c[] = "abcd"; char c[50] = "abcd"; char c[] = {'a', 'b', 'c', 'd', '\0'}; char c[5] = {'a', 'b', 'c', 'd', '\0'};</pre>	<pre>String c = "12345"; String c=new String("12345"); char[] ch={'1','2','3','4','5'}; String s=new String(ch);</pre>

Another data type that is not in C language but in Java language is boolean. Specifies boolean logical expressions. Returns true, false data. In C language, this has no direct equivalent. Integer type flag is defined, if flag is 0, it returns false. In other languages except 0, it returns true.

The other rule for semantic is statements. Statements in C and Java programming languages are very similar. We can classify for three parts. First part is that conditional statement. If statement contains if, else and if else and switch – case statements. Second part is loop statements. Both C and Java contain

for, while, and do – while loops. One loop that includes in Java but does not include in C programming. Its called for-each loop. The for-each loop is used to traverse array or collection in Java. It is easier to use than simple for loop because we don't need to increment value and use subscript notation. It works on the basis of elements and not the index. It returns element one by one in the defined variable. The last part is flow control statement. These part is the same. There are 3 keywords for flow statement. These are return, continue and break.

Efficiency:

C language is faster than Java. There are some reasons for this situation. The main reason is that Java uses an interpreter of intermediate code, C uses a compiler to generate native machine code.

The second reason is that OO languages have some overhead for object management. But this all depends on what you are doing.

Memory Allocation:

Memory allocation is different for C and Java languages. For C language, it supports two kinds of memory allocation through the variables in its programs: First one is that static allocation is what happens when you declare a static or global variable. Each static or global variable defines one block of space, of a fixed size. The space is allocated once, when your program is started and is never freed. The second one is that. Dynamic memory allocation is a technique in which programs determine as they are running where to store some information. You need dynamic allocation when the amount of memory you need, or how long you continue to need it, depends on factors that are not known before the program runs. For java programming languages, memory allocation in java refers to the process where the computer programs and services are allocated dedicated to virtual memory spaces. The Java Virtual Machine divides the memory into Stack and Heap Memory. For Java Virtual Machine, executing an application in its maximum potential can happen from stack and heap memory. Every time a new variable or object is declared, the memory allocates memory dedicated to such operations.

Learning Curve :

It is not correct to make general definitions about people's aptitude for programming languages. The predisposition of individuals to programming languages may vary from person to person. Statistical data about people's inclination towards programming languages in general can give us an idea about the general situation. In this regard, we can consider surveys of professional and regular developers around the world. When we look at the 2022 survey of Stackoverflow, 10817 respondents stated that they like the JAVA programming language with a percentage of 45.75%. When we look at the C programming language, 5433 people stated that they like this language with 39.68%. When the participants were asked which programming language they prefer to develop with, the preference for Java language was higher than C language. When we look at the survey in general, the preference for object-oriented programming languages has been higher. Today, object-oriented programming languages are preferred due to definitions and easier expression. Object-oriented programming languages are preferred in the events held by organizations such as Code.Org around the world, and daily life objects and code pieces are exemplified. In this way, the learning curve of the programming language has increased drastically.

Conclusion :

In conclusion, this essay has compared differences and similarities between C and Java programming languages. Based on these comparisons, We have stressed the different philosophies for these two programming languages. As part of the comparison, we assessed the suitability of C and Java for programming in some different parts.