

## A Brief Overview On C and Java

Although C and Java has similar features, they have very different languages. C is a procedural oriented programming language. Procedural programming is a programming language that is derived from structure programming and based upon the concept of calling functions. It follows a step-by-step approach in order to break down a task into a set of variables and routines via a sequence of instructions. This languages uses top-down approach. Top-down approach is a programming style that forms the basis of traditional procedural languages, where the design begins with identifying complex parts and then dividing them into smaller parts. There is no access specifiers in procedural programming like private, public, protected. Procedural programming does not have any proper way of hiding data like encapsulation. Therefore, it is less secure than object oriented programming. In procedural programming, overloading is not possible. In procedural programming, there is no concept of inheritance because they do not have classes. In procedural programming, the function is more important than the data. Procedural programming is based on the unreal world because procedural programming languages does not have objects and classes. Procedural programming is used for designing medium-sized programs. Procedural programming uses the concept of procedure abstraction. **Example:** In C when we write from math library `sqrt`; `sqrt(4)` we really don't care about how `sqrt` function is implemented in math library. we have no idea how that `sqrt(4)` actually gets computed. We just care about functioning of `sqrt` here. There is no code reusability present in procedural programming.

In contrast, Java is an object oriented programming language. Object-oriented programming is a computer programming design philosophy or methodology that organizes- models software design around data or objects rather than functions and logic. In object-oriented programming, the program is divided into small parts called objects. Object-oriented programming follows a bottom-up approach. In a bottom-up approach, the basic elements of the system are first specified in detail. These elements are then Decoupled together to form larger subsystems and the subsystems are connected at many levels until a complete higher-level system is created. Object-oriented programming has access specifiers like private, public, protected. Object-oriented programming provides data hiding (encapsulation) so it is more secure. Overloading is possible in object-oriented programming like function overloading and operator overloading. In object-oriented programming, the concept of inheritance is used. In object-oriented programming, data is more important than function. Object-oriented programming is based on the real world because they have objects and classes. Object-oriented programming is used for designing large and complex programs. Object-oriented programming uses the concept of data abstraction. **Example:** In Java, `HashMap` provides methods to store and retrieve key-value pairs from the map. So only information we need to use the hash map is to identify it's use case of it in our application. How it is implemented inside, we do not have to know it. It offers code reusability by using the feature of inheritance.

On the other hand if we compare this two language as syntax, semantically and efficiency; C language has 32 keywords, But Java language has 52 keywords. C language does not have Boolean keyword we can only use it with `<stdbool.h>` library. Instead of Boolean C uses Integer, 0 for false and nonzero for true. In Java, Boolean is Java's own type and stores value true or false. In C char is usually 8 bit ASCII against In Java, char is 16 bit UNICODE. In C Array declarations like `int *a = malloc(N * sizeof(*a))` and arrays don't know their own size but in Java `int[] a = new int[N]` and with array name.length procedure array show's it's size. Identifiers in C are the user-defined words. It can be composed of uppercase letters, lowercase letters, underscore, or digits, but the starting letter should be either an underscore or an alphabet. Identifiers cannot be used as keywords. Rules for constructing identifiers in C: The first character of an identifier should be either an alphabet or an underscore, and

then it can be followed by any of the character, digit, or underscore. It should not begin with any numerical digit. In identifiers, both uppercase and lowercase letters are distinct. Therefore, we can say that identifiers are case sensitive. Commas or blank spaces cannot be specified within an identifier. Keywords cannot be represented as an identifier. The length of the identifiers should not be more than 31 characters. Identifiers should be written in such a way that it is meaningful, short, and easy to read. In Java; Identifiers are used to name a variable, constant, function, class, and array. It is usually defined by the user. It uses letters, underscores, or a dollar sign as the first character. The label is also known as a special kind of identifier that is used in the goto statement. The identifier name must be different from the reserved keywords. It can not contain white spaces. Difference between Java and C rules; In Java identifiers can be start with \$ sign. There is no limit on the length of the identifier but it is advisable to use an optimum length of 4 – 15 letters only. They both have the same type operators. This operator types are: Arithmetic Operators, Relational Operators, Shift Operators, Logical Operators, Bitwise Operators, Ternary or Conditional Operators, Assignment Operator, Misc Operator. Additional C has sizeof() operator for return the size of a variable, & operator for return the address of a variable, \* for pointer to a variable and Java has instanceof operator and this operator is used only for object reference variables. The operator checks whether the object is of a particular type (class type or interface type). These operators are subtypes of miscellaneous operators. Let's look at making constant variables for both C and Java languages. A constant is a value assigned to the variable which will remain the same throughout the program. In C, There are two ways of declaring constant: Using const keyword, Using #define pre-processor. In Java, the final modifier represents that the value of the variable cannot be changed. It also makes the primitive data type immutable or unchangeable. But If we declare a variable as static, all the objects of the class (in which constant is defined) will be able to access the variable and can be changed its value. To overcome this problem, we use the final modifier with a static modifier. When the variable defined as final, the multiple instances of the same constant value will be created for every different object which is not desirable. When we use static and final modifiers together, the variable remains static and can be initialized once. Therefore, to declare a variable as constant, we use both static and final modifiers. It shares a common memory location for all objects of its containing class. Let's look at strings for both C and Java. String in C programming is a sequence of characters terminated with a null character '\0'. Strings are defined as an array of characters. The difference between a character array and a string is the string is terminated with a unique character '\0'. Declaring a string is as simple as declaring a one-dimensional array and there are a lot of way. Some of them like assigning a string literal without size char str[] = "Alperen", assigning a string literal with a predefined size: char str[50] = "AlperenKara"; Important note there is an extra terminating character which is the Null character ('\0') used to indicate the termination of a string that differs strings from normal character arrays. In Java: String objects are stored in a special memory area known as string constant pool to make Java more memory efficient (because no new objects are created if it exists already in string constant pool). There are two ways to create a string in java. These are string literal String s = "Alperen", Using new keyword String s = new String ("AlperenKara"); In java, string objects are immutable. Immutable simply means unmodifiable or unchangeable. Once string object is created its data or state can't be changed but a new string object is created. Finally let's compare for last token of the array separators. In C: Square brackets [ ]: The opening and closing brackets represent the single and multidimensional subscripts. Simple brackets ( ): It is used in function declaration and function calling. For example, printf() is a pre-defined function. Curly braces { }: It is used in the opening and closing of the code. It is used in the opening and closing of the loops. Comma (,): It is used for separating for more than one statement and for example, separating function parameters in a function call, separating the variable when printing the value of more than one variable using a single printf statement. Hash/pre-processor (#): It is used for pre-processor directive. It basically denotes that we are using the header file. Asterisk (\*): This symbol is used to represent pointers and also used as an operator for multiplication. Tilde (~): It is used as a destructor to free memory. Period (.): It is used to access a member of a structure or a union. And in Java Square Brackets []: It is used to define array elements. A pair of square brackets represents the

single-dimensional array, two pairs of square brackets represent the two-dimensional array. Parentheses (): It is used to call the functions and parsing the parameters. Curly Braces {}: The curly braces denote the starting and ending of a code block. Comma (,): It is used to separate two values, statements, and parameters. Assignment Operator (=): It is used to assign a variable and constant. Semicolon (;): It is the symbol that can be found at end of the statements. It separates the two statements. Period (.): It separates the package name from the sub-packages and class. It also separates a variable or method from a reference variable. Because of there is no pointers in Java asterisk (\*) and tilde (~) are don't included in Java. Both of them use same comment type. Line Oriented: It begins with a pair of forwarding slashes (//).Block-Oriented: It begins with /\* and continues until it founds \*/.

On the other hand we can compare them as semantics. Semantic, refers to the meaning associated with any statement in the programming language. If there is a problem with semantic, it referred to as semantic error. It is generally encountered at run time. It occurs when a statement is syntactically valid but does not do what the programmer intended. This type of error is tough to catch. There are two areas of semantics that are logical semantics and lexical semantics. Most of the semantics are case-insensitive. In C when declaring a variable, C does not force programmer to bind variable on initializing it. When variable is initialized it can has a garbage value on it. C has both pass by value and pass by reference. C is all about pointers. C has strong support for pointers and we can do a lot of useful programming using pointers. C supports the goto statement, but its use should be minimized in order to prevent the consequences of using it in a program. In contrast Java; Java force programmer to bind variables on initializing it. Java has always pass by value not pass by reference. Java has limited support for pointers. Initially, Java was completely without pointers but later versions started providing limited support for pointers. We cannot use pointers in Java as much as we can use in C. Java does not have goto statements.

As a result; both languages has a lot of advantages and some disadvantages. With memory allocation and pointers in C, programmers can write very optimized and fast programs. For thus C programming language can be used for system programming as well as Application programming. The compiler transforms the C code into machine code, all at once. C supports Preprocessors. C is used in limited areas like Embedded Systems, Systems programming, Open source software, Control systems. In Java; Java handles allocation automatically with automatic garbage collection. This prevents memory leaks. When executing the code for the first time, it is interpreted. However, when a piece of code gets executed frequently, it is compiled to the machine code in real-time and further execution uses the compiled version. Java can be used in a lot of area because it is a high level language. This areas are Desktop GUI Applications, Web Applications, Mobile Applications, Enterprise Applications, Scientific Applications, Web Servers & Applications Servers, Embedded Systems etc.

#### References:

<https://www.geeksforgeeks.org/differences-between-procedural-and-object-oriented-programming/>

<https://www.javatpoint.com/procedural-programming-vs-object-oriented-programming>

[https://tr.wikipedia.org/wiki/Yukar%C4%B1dan\\_a%C5%9Fa%C4%9F%C4%B1ya\\_ve\\_a%C5%9Fa%C4%9F%C4%B1dan\\_yukar%C4%B1ya\\_tasar%C4%B1m#:~:text=Yukar%C4%B1dan%20a%C5%9Fa%C4%9F%C4%B1ya%2C%20tasar%C4%B1m%C4%B1n%20karma%C5%9F%C4%B1k%20par%C3%A7alar%C4%B1n,tan%C4%B1mlayan%20bir%20ana%20yordam%20yazmakt%C4%B1r.](https://tr.wikipedia.org/wiki/Yukar%C4%B1dan_a%C5%9Fa%C4%9F%C4%B1ya_ve_a%C5%9Fa%C4%9F%C4%B1dan_yukar%C4%B1ya_tasar%C4%B1m#:~:text=Yukar%C4%B1dan%20a%C5%9Fa%C4%9F%C4%B1ya%2C%20tasar%C4%B1m%C4%B1n%20karma%C5%9F%C4%B1k%20par%C3%A7alar%C4%B1n,tan%C4%B1mlayan%20bir%20ana%20yordam%20yazmakt%C4%B1r.)

<https://medium.com/@rshkntshrm/power-of-procedural-abstraction-249fd7529473>

<https://introcs.cs.princeton.edu/java/faq/c2java.html>

<https://stackoverflow.com/questions/1921539/using-boolean-values-in-c>

<https://www.javatpoint.com/tokens-in-c>

<https://www.javatpoint.com/java-tokens>

[https://www.geeksforgeeks.org/java-identifiers/#:~:text=Rules%20for%20defining%20Java%20Identifiers&text=The%20only%20allowed%20characters%20for,%5B0%2D9%5D\).](https://www.geeksforgeeks.org/java-identifiers/#:~:text=Rules%20for%20defining%20Java%20Identifiers&text=The%20only%20allowed%20characters%20for,%5B0%2D9%5D).)

[https://www.tutorialspoint.com/cprogramming/c\\_operators.htm](https://www.tutorialspoint.com/cprogramming/c_operators.htm)

[https://www.tutorialspoint.com/java/java\\_basic\\_operators.htm](https://www.tutorialspoint.com/java/java_basic_operators.htm)

<https://www.javatpoint.com/operators-in-java>

<https://www.javatpoint.com/c-operators>

<https://www.javatpoint.com/java-constant>

<https://www.geeksforgeeks.org/strings-in-java/>

<https://www.geeksforgeeks.org/strings-in-c/>

[https://www.geeksforgeeks.org/difference-between-java-and-c-language/#:~:text=C%20is%20a%20compiled%20language,JVM%20\(Java%20Virtual%20Machine\).](https://www.geeksforgeeks.org/difference-between-java-and-c-language/#:~:text=C%20is%20a%20compiled%20language,JVM%20(Java%20Virtual%20Machine).)

<https://techdifferences.com/difference-between-c-plus-plus-and-java.html>

<https://www.softwaretestinghelp.com/cpp-vs-java/>

<https://medium.com/swlh/a-performance-comparison-between-c-java-and-python-df3890545f6d>

<https://www.quora.com/What-kinds-of-applications-is-C-used-for>

<https://www.softwaretestinghelp.com/real-world-applications-of-java/>