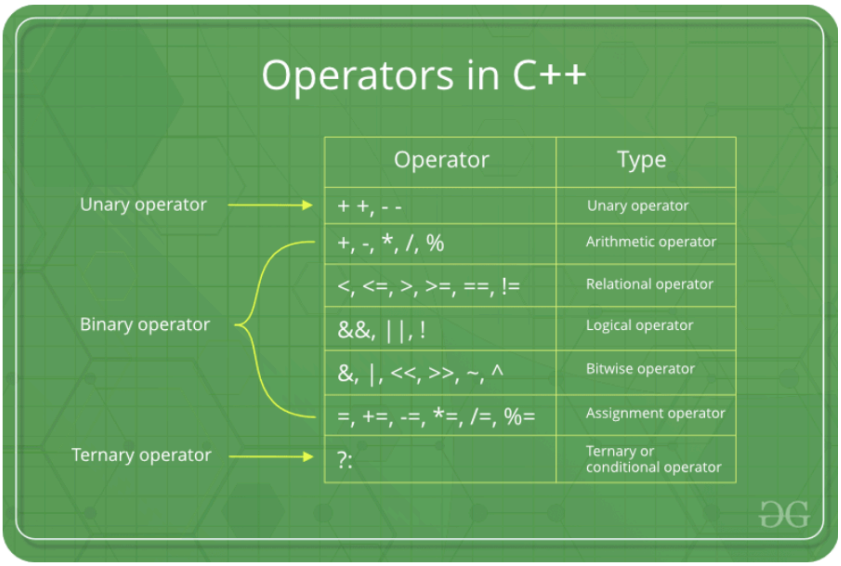
1. Does your language have keywords or reserved words? How many?
    a. Keywords can't be used for a variable name, function name, or any other identifiers. The total count of reserved keywords in C++ is 95.
2. What are the naming requirements for variables in your language? What about naming conventions? Are those enforced by the compiler/interpreter, or just standards in the community?
    a. In C++, naming conventions are not enforced by the compiler or interpreter. The compiler only checks for the correct syntax and structure of the code but does not care about the names of variables, functions, or classes as long as they follow the basic rules of valid identifiers (e.g., no spaces, starting with a letter or underscore, etc.).
    b. According to https://www.geeksforgeeks.org/naming-convention-in-c/ naming conventions used by the community are:
        i. The class name should be a noun.
        ii. Use upper case letters as word separators, and lower case for the rest of the word in the class name.
        iii. The first character in the class name must be in upper case.
        iv. No underscores ('_') are permitted in the class name.
        v. The private attribute name in class should be prepended with the character 'm'.
        vi. After prepending 'm', the same rules will be followed for the name as that for the class name.
        vii. Character 'm' also precedes other name modifiers also. For example, 'p' for pointers.
        viii. Each method/ function name should begin with a verb.
        ix. The first character of function/ method argument names should be lowercase. All words starting after the first letter should be in the upper case with class names.
        x. The variable name should begin with an alphabet.
        xi. Digits may be used in the variable name but only after the alphabet.
        xii. No special symbols can be used in variable names except for the underscore('_').
        xiii. No keywords can be used for variable names.
        xiv. Pointer variables should be prepended with 'p' and place the asterisk '*' close to the variable name instead of the pointer type.
        xv. Reference variables should be prepended with 'r'. This helps to differentiate between the method returning a modifiable object and the same method returning a non-modifiable object.
        xvi. Static variables should be prepended with 's'.
        xvii. The global constants should be all capital letters separated with '_'.

xviii. No special character is allowed in the file name except for underscore (‘_’) and dash (‘-‘).

xix. The file name should end with the .cc extension in the end or should end with the .cpp extension.

xx. Do not use filenames that already exist in /user/include. or any predefined header file name.

3. Is your language statically or dynamically typed?
    a. C++ is statically typed, meaning variables are bound to a data type during compilation. Also meaning the programmer has to declare the type of the variable when it is originally declared. Data_type variable_name.
4. Strongly typed or weakly typed?
    a. Generally a strongly typed language has stricter typing rules. Error and exceptions can be more likely to happen at compilation. An object may only be sent a message containing a method that is defined in the object's class, or in any of the classes that this object inherits from. C++ is a strongly typed language because the variables have to be specified before you declare them. https://atomicobject.com/oo-programming/object-oriented-typing
5. Are some variables mutable while others are immutable?
    a. https://www.geeksforgeeks.org/c-mutable-keyword/
    b. The keyword mutable is mainly used to allow a particular data member of const object to be modified.
6. What are the operators available for each data type?
    a. Different operators are available for each of the 6 data types in C++.
    b. https://www.geeksforgeeks.org/operators-in-cpp/
    c.



7. Are mixed type operations allowed? If so, how are they accommodated?

    a. Yes, C++ follows a set of rules when performing mathematical operations on variables of different data types. Data types have a ranked system so when conversion happens some get demoted/promoted.
    b. https://medium.com/@ctchalland/when-you-mix-nuts-and-bolts-type-conversion-c5f4f07e4d83#:~:text=The%20answer%20is%20yes%2C%20C%2B%2B,a%20float%20outranks%20an%20int.
8. At what point are identifier names and operator symbols bound in your language? For example if you declare a (variable, class name, function name), when is it bound to the type, address? When are operators (+,*, etc.) bound to their operations?
    a. Identifier names and operator symbols are bound during compilation. Ex. int x=1 the x variable is bound when compiled.
9. Explicitly typed or implicitly typed?
    a. C++ is an explicitly typed language. An explicitly typed language is when you have to provide as much detail as you can to variables, functions, etc while implicitly typed language is when the language figures out the details for you. An example of this would be when you are assigning a variable like x = 42, in C++ you would have to say the type it is like (int x = 42). Unlike in python where you can just say x = 42 and it would figure it out. https://www.incredibuild.com/blog/implicit-vs-explicit-in-programming-key-differences#:~:text=Take%20languages%20like%20C%2B%2B,understand%20exactly%20what's%20going%20on.