Is it legal to use the phrase "**Google**" as an "internal programming identifier to write computer software code?

Here's why it is generally permissible but also carries some risk:

Internal Use & Lack of Public Confusion: Trademark law is primarily concerned with preventing consumer confusion in the marketplace. If the term "google" is used strictly as an internal variable, function, or class name within your own non-public source code (e.g., google_search_result_parser), and never appears in the user interface, marketing, or public-facing materials in a way that suggests endorsement or sponsorship by Google, it is unlikely to be considered trademark infringement.

Nominative Fair Use: You are permitted to use another's trademark to refer truthfully to the trademark owner or its product (nominative fair use). For example, using "google" as part of a variable name to describe data retrieved from an interaction with a Google API would be a valid, descriptive use.

Best Practices and Risk Avoidance: While likely legal for strictly internal use, many developers and legal experts advise against using trademarked terms unnecessarily.

- Potential for Discovery: In the event of a lawsuit for any reason, internal code can be discoverable, which could lead to uncomfortable questions about the intent behind using the name, even if no infringement occurred.

- Code Clarity: Using descriptive, generic names (e.g., search_engine_results) instead of specific brand names often makes code more maintainable, flexible, and easier for other developers to understand.

- Avoiding "Hazy Cases": Choosing a unique name entirely avoids any potential legal ambiguity or the need to defend a "fair use" claim.

In summary, the use of "google" as an identifier in your private code is unlikely to lead to legal trouble, but using generic or unique identifiers is considered a best practice to avoid potential complications and ensure code clarity.

Is it legal to use the phrase "**leafBuilder**" as an "internal programming identifier to write computer software code?

Using the phrase "leafBuilder" as a purely internal programming identifier is legal because trademark law only applies to use "in commerce" in a way that is likely to cause consumer confusion about the source of goods or services. Internal identifiers not exposed to the public or customers generally do not meet this "in commerce" standard.

However, "leafBuilder" is an active, registered trademark. There are practical and legal risks to consider:

- Potential for Exposure: If the internal identifier is ever exposed in public-facing materials, such as in open-source code repositories, public documentation, or error messages displayed to users, it could be seen as use in commerce and lead to a claim of trademark infringement.

- Legal Scrutiny in Disputes: If a legal dispute arises for any other reason, the use of a third party's trademarked term, even internally, could lead to questions about intent or potential access to trade secrets, which can be a costly complication.

- Expansion of Business: If your software product were to grow into a commercial offering and "leafBuilder" was still used as a prominent identifier, it could cause confusion in the marketplace, especially if the trademark owner is in a related industry (software/web services seem relevant to the existing trademark owner).

Conclusion: While strictly internal use is likely permissible, it is highly advisable to choose a unique, non-trademarked identifier to avoid any potential legal entanglements and future complications. Consulting with a trademark attorney for your specific situation is the safest approach.