1. Lookup: Imagine you have a big list of names. When you're given a name, you want to quickly check if it's on that list or not. That's what lookup does. It tells you whether a given name is already in the list or not.
2. Insert: If you have a new name that's not on the list, you need a way to add it. Inserting a name means putting it onto the list.
3. Access: Once a name is on the list, there's usually more information associated with it, like what type of thing it represents (like a variable or a function), or where it's used in the program. Access allows you to get to that extra information when you have the name.
4. Modify: Sometimes, you might need to change or add more information about a name that's already on the list. Modifying lets you do that. For example, you might want to add the datatype of a variable.
5. Delete: If a name is no longer needed or if it's causing conflicts, you might want to remove it from the list. Deleting allows you to take names off the list.

Related posts:

1. What are the types of passes in compiler ?
2. Discuss the role of compiler writing tools. Describe various compiler writing tools.
3. What do you mean by regular expression ? Write the formal recursive definition of a regular expression.
4. How does finite automata useful for lexical analysis ?
5. Explain the implementation of lexical analyzer.
6. Write short notes on lexical analyzer generator.
7. Explain the automatic generation of lexical analyzer.
8. Explain the term token, lexeme and pattern.
9. What are the various LEX actions that are used in LEX programming ?

10. Describe grammar.
11. Explain formal grammar and its application to syntax analyzer.
12. Define parse tree. What are the conditions for constructing a parse tree from a CFG ?
13. Describe the capabilities of CFG.
14. What is parser ? Write the role of parser. What are the most popular parsing techniques ? OR Explain about basic parsing techniques. What is top-down parsing ? Explain in detail.
15. What are the common conflicts that can be encountered in shift-reduce parser ?
16. Differentiate between top-down and bottom-up parser.Under which conditions predictive parsing can be constructed for a grammar ?
17. Differentiate between recursive descent parsing and predictive parsing.
18. What is the difference between S-attributed and L-attributed definitions ?
19. What is intermediate code generation and discuss benefits of intermediate code ?
20. Define parse tree. Why parse tree construction is only possible for CFG ?
21. What are the symbol table requirements ? What are the demerits in the uniform structure of symbol table ?