

How does a compiler work?The type and extent of optimization depends on the compiler, although most compilers aim to enhance the program's speed, efficiency and resource utilization, e.g., CPU or memory. The lexemes are then tokenized, i.e., they are organized into meaningful character sequences called tokens, in preparation for the next steps of syntax analysis and semantic analysis. The compiler interprets the meaning of the tokens created during lexical analysis and verifies that the code's syntax is correct, based on the rules for the source language. During this step, the compiler typically creates abstract syntax trees that represent the logical structures of specific code elements. Compilers vary in the methods they use for analyzing and converting source code to output code. The compiler splits the source code into lexemes, which are individual code fragments that represent specific patterns in the code. The compiler generates the final output code, using the optimized IR code, to create code that the machine can finally execute. Intermediate representation (IR) code generation. Optimization. 5.7.9. 11. 13.