

# Compiler Design - Architecture

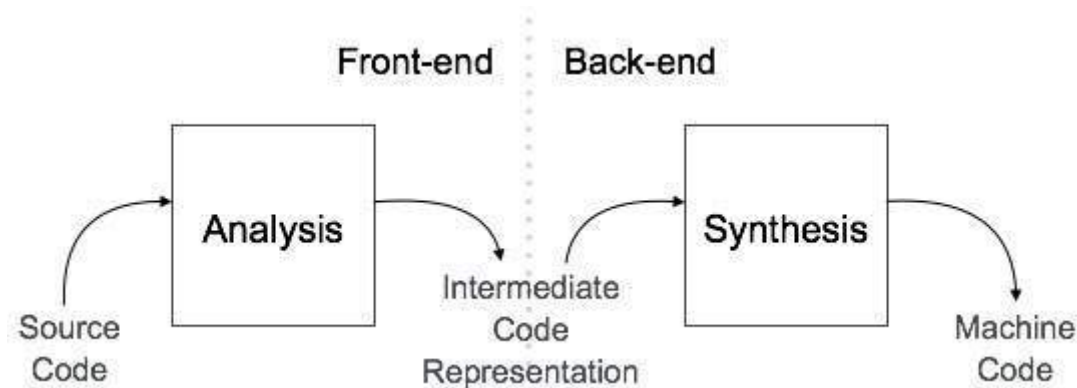
[https://www.tutorialspoint.com/compiler\\_design/compiler\\_design\\_architecture.htm](https://www.tutorialspoint.com/compiler_design/compiler_design_architecture.htm)

Copyright © tutorialspoint.com

A compiler can broadly be divided into two phases based on the way they compile.

## Analysis Phase

Known as the front-end of the compiler, the **analysis** phase of the compiler reads the source program, divides it into core parts and then checks for lexical, grammar and syntax errors. The analysis phase generates an intermediate representation of the source program and symbol table, which should be fed to the Synthesis phase as input.



## Synthesis Phase

Known as the back-end of the compiler, the **synthesis** phase generates the target program with the help of intermediate source code representation and symbol table.

A compiler can have many phases and passes.

- **Pass** : A pass refers to the traversal of a compiler through the entire program.
- **Phase** : A phase of a compiler is a distinguishable stage, which takes input from the previous stage, processes and yields output that can be used as input for the next stage. A pass can have more than one phase.