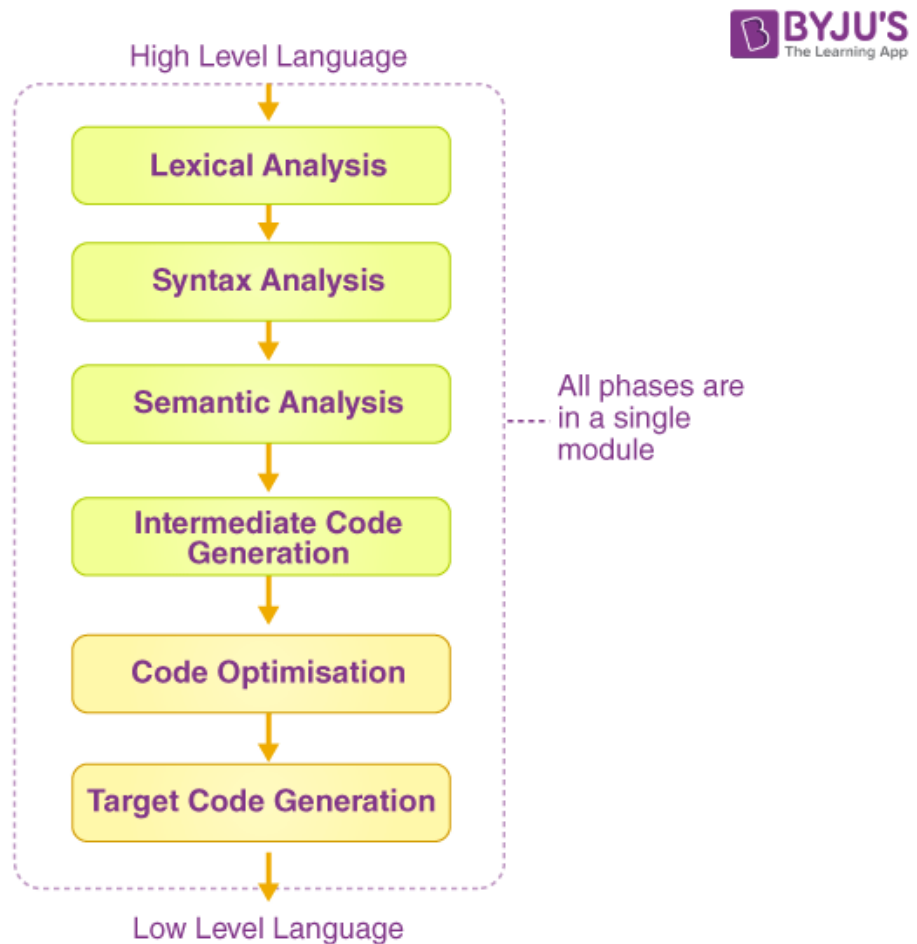


Assignment 4

1. How does a compiler work

A compiler takes the program code (source code) and converts the source code to a machine language module called an object file. Another specialized program, called a linker, combines this object file with other previously compiled object files and libraries to create an executable file.

The compilation process goes through multiple stages before the machine code is produced at the end.



2. Types of compiler:

1. **Cross Compilers:** Produce an executable machine code for a platform but, this platform is not the one on which the compiler is running.
2. **Bootstrap compilers:** These compilers are written in a programming language that they have to compile.

3. **Transcompiler:** Convert the source code of one programming language to the source code of another programming language.
4. **Single Pass Compilers:** When all the phases of compiler design is in a single module. Source code converts directly into machine code.
5. **Two Pass Compiler:** A processor that runs through the program to be translated twice is considered a two-pass compiler.
6. **Multipass Compiler:** A program's source code or syntax tree is processed many times by the multipass compiler. It breaks down a huge programme into numerous smaller programmes and runs them all at the same time.
7. **Incremental Compilers:** A compiler which executes the recompilation of only a changed source instead of compiling the complete source code.