AIM:- Introduction: objective, scope and outcome of the course

objective:- The laboratory course is intended to make experiments on the basic techniques of compiler cons. and tools that can be used to perform Syntax directed translation of a high-level programming lang. into an executable code.

This will deeper insights into the more advanced semantics aspects of programming lang. code generation and object orientation.

Scope:- The scope of this course is to explore the principle, algo. and data structure involved in the design and const. of compiler

outcomes:-

1. understand the working of ex and yacc compiler for debug
2. understand the working of lex and define the role of lexical analyzer.
3. understand and use context free grammer and parse tree
4. Learn & amp;
5. Develop program for solving larger problems
6. Learn how to write programs that execute faster

Introduction of CD

compiler is a software which converts a program written in H/L (high Level Lang.) source Lang. to LLL. (object / Target / HL)

cross compiler that run on A and produces a code for mac. B

- Phases of a compiler:-

There are two major phases of compilation, which is turn have many parts. each of them take input from the o/p of previous level and work in a coordinated way

Analysis phase :- An semantic tree representation is created from the given source code :- 1. Lexical Analyzer 2. Syntax Analyzer.

3. Semantic Analyzer:- Lexical analyzer divides the program into tokens syntax analyzer recognizes "sentences" in the prog. using syntax of lang. and sematic analyzer checks static semantic of each other.

Synthesis phase :- It has three parts:-

4. Intermediate code Generator

5. code optimizer

6. code generator

Intermediate code generator generates "abstract" code. code optimizer the abstract code, and final code generator translates abstract intermediate code into specific machine instruction.