COMPILERS

LECTURE 0
COURSE INTRODUCTION & PLANE

Prof. Dr. Hala Abdel-El Galil
Dr. Ahmed Hesham Mostafa

Text Books

- Main reference:
  - Compilers – Principles, Techniques and Tools, Second Edition by Alfred V. Aho, Ravi Sethi, Jeffery D. Ullman
- others references:
  - Modern compiler construction in Java 2nd edition
  - Advanced Compiler Design and Implementation by Muchnick
  - Cooper & Torczon, *Engineering a Compiler*

Course Contents

- Ch1. Introduction
- Ch3.Lexical analysis (Scanning)
- Ch4. Syntax Analysis (Parsing)
- Ch5. Syntax Directed Translation (Semantic Analysis)
- Ch6. Intermediate Code Generation

Course Aims and Objectives

The Objectives of this course is to explore the principles, algorithms, and data structures involved in the design and construction of compilers.

Understand the Lexical and Syntax phases of a compiler.

Understand transformation of source code to parse tree.

Course Aims and Objectives

Learn Regular Expressions

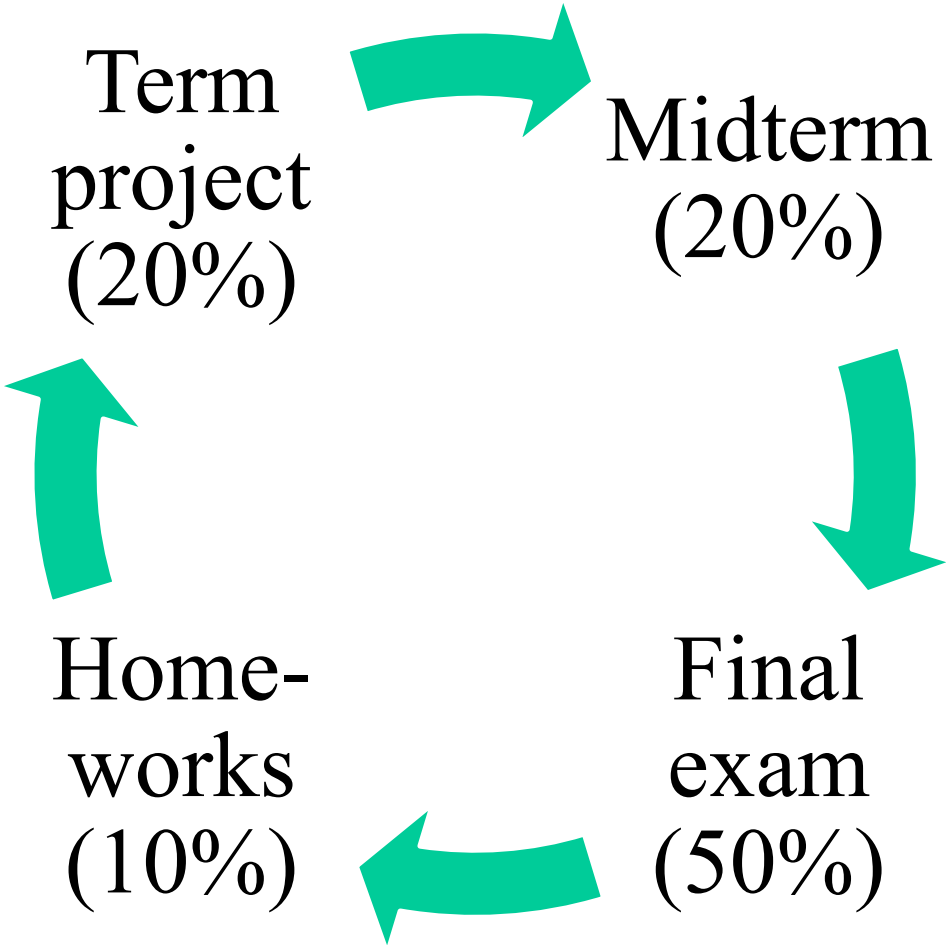
Converting Regular Expression to Finite automata

Learn about Finite Automata

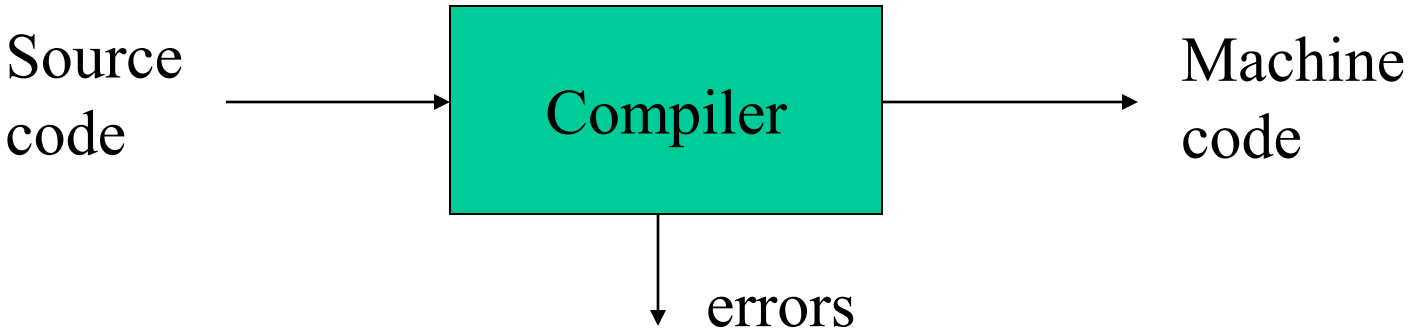
Lexical and Syntax analysis combination

The Aim of this course to learn and implement the Front Phase of compiler while the bakend phase is advanced topic

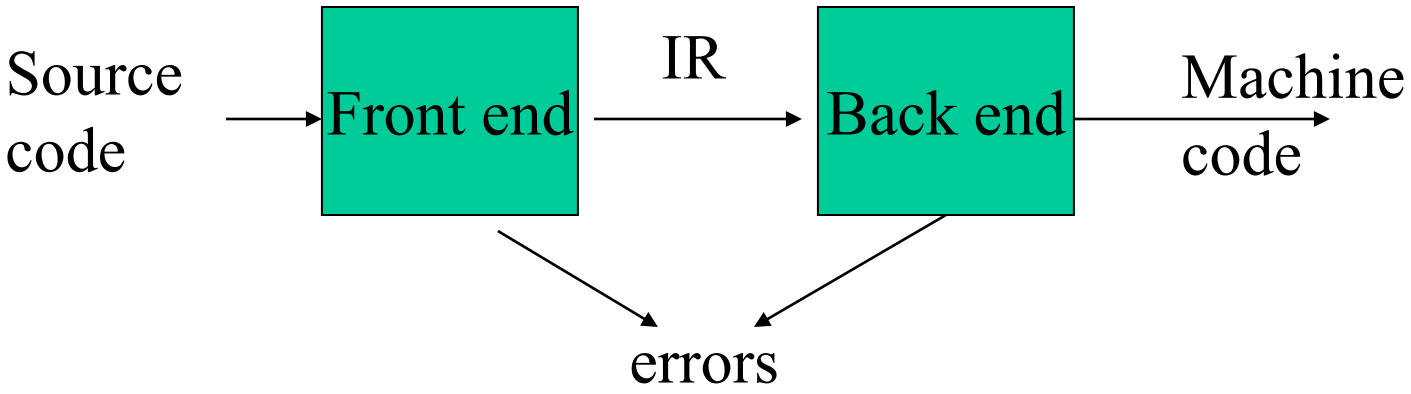
Grading policy



Abstract view



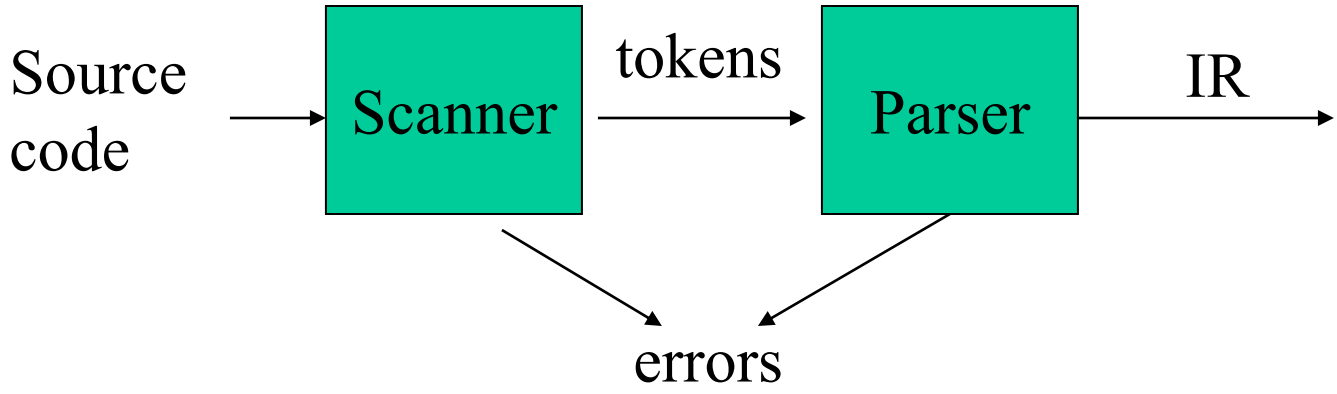
Front-end, Back-end division



Front end maps legal code into IR(Intermmidate Representation)

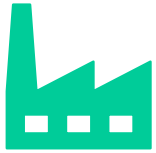
Back end maps IR onto target machine

Front end



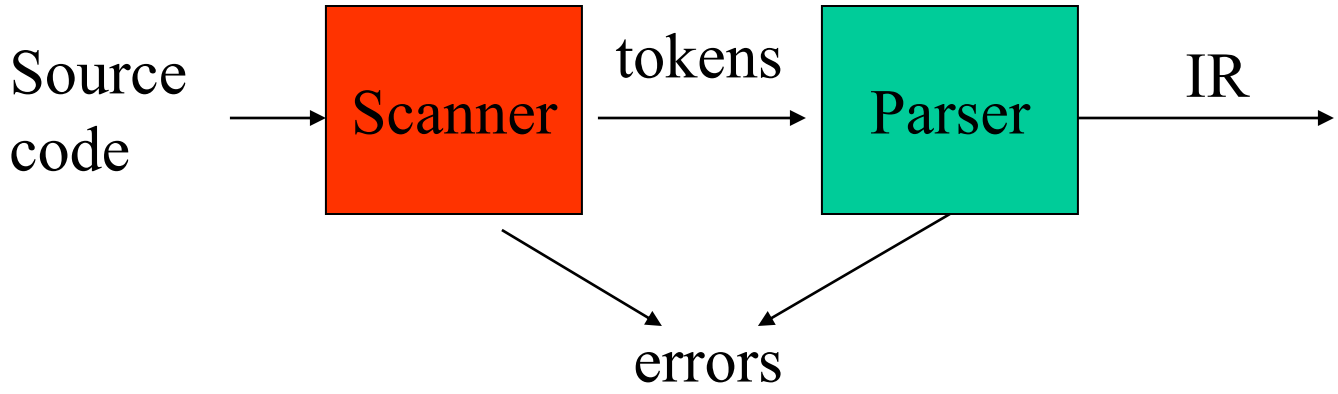
Recognize legal code

Report errors



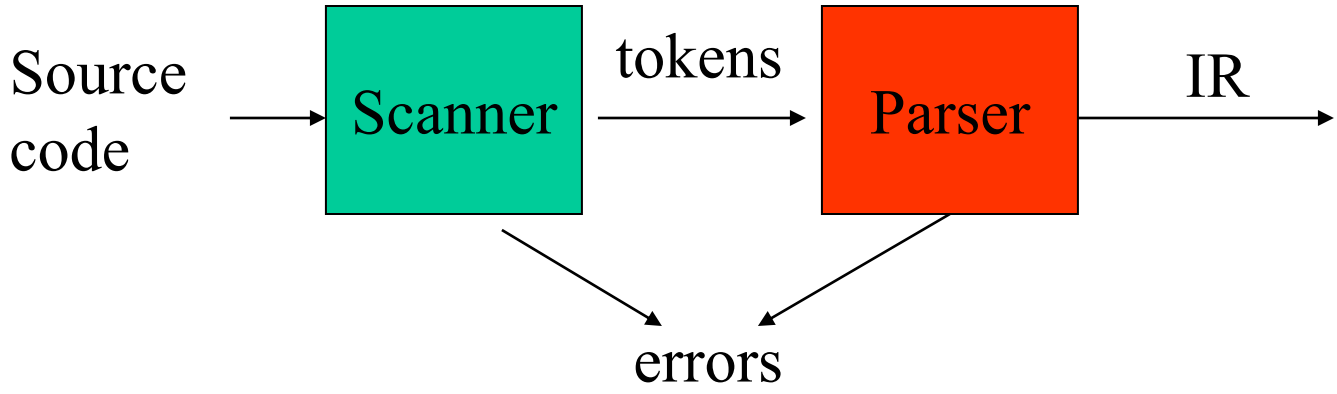
Produce IR

Front end



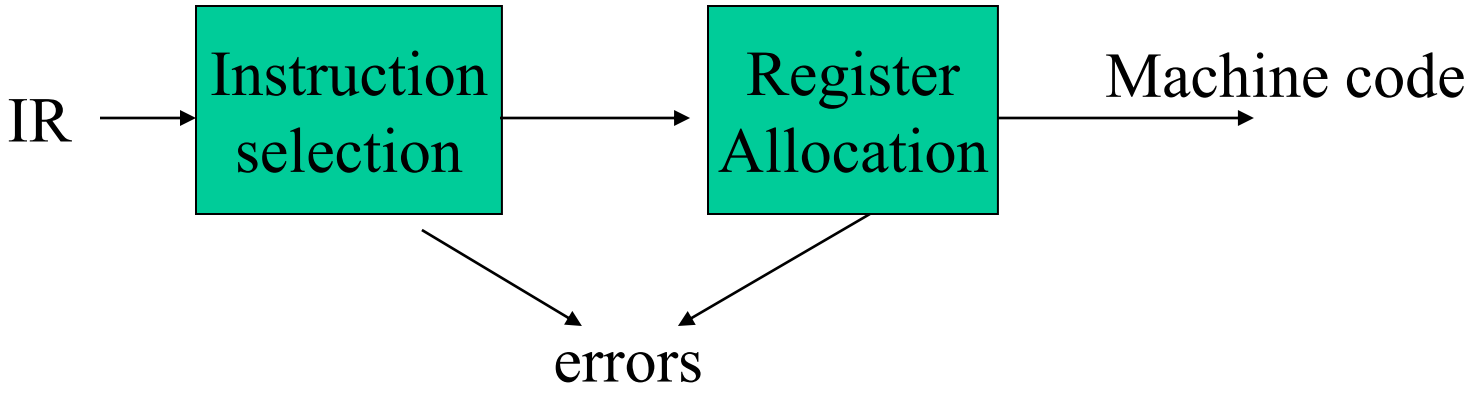
- Scanner:
  - Maps characters into tokens – the basic unit of syntax
    - x = x + y becomes <id, x> = <id, x> + <id, y>
  - Typical tokens: number, id, +, -, *, /, do, end
  - Eliminate white space (tabs, blanks, comments)

Front end



- Parser:
  - Recognize context-free syntax
  - Guide context-sensitive analysis
  - Construct IR
  - Produce meaningful error messages
  - Attempt error correction

Back end



Translate	Choose	Decide	Ensure
Translate IR into target machine code	Choose instructions for each IR operation	Decide what to keep in registers at each point	Ensure conformance with system interfaces

Academic Integrity

We want a cooperative group working together to do great stuff!

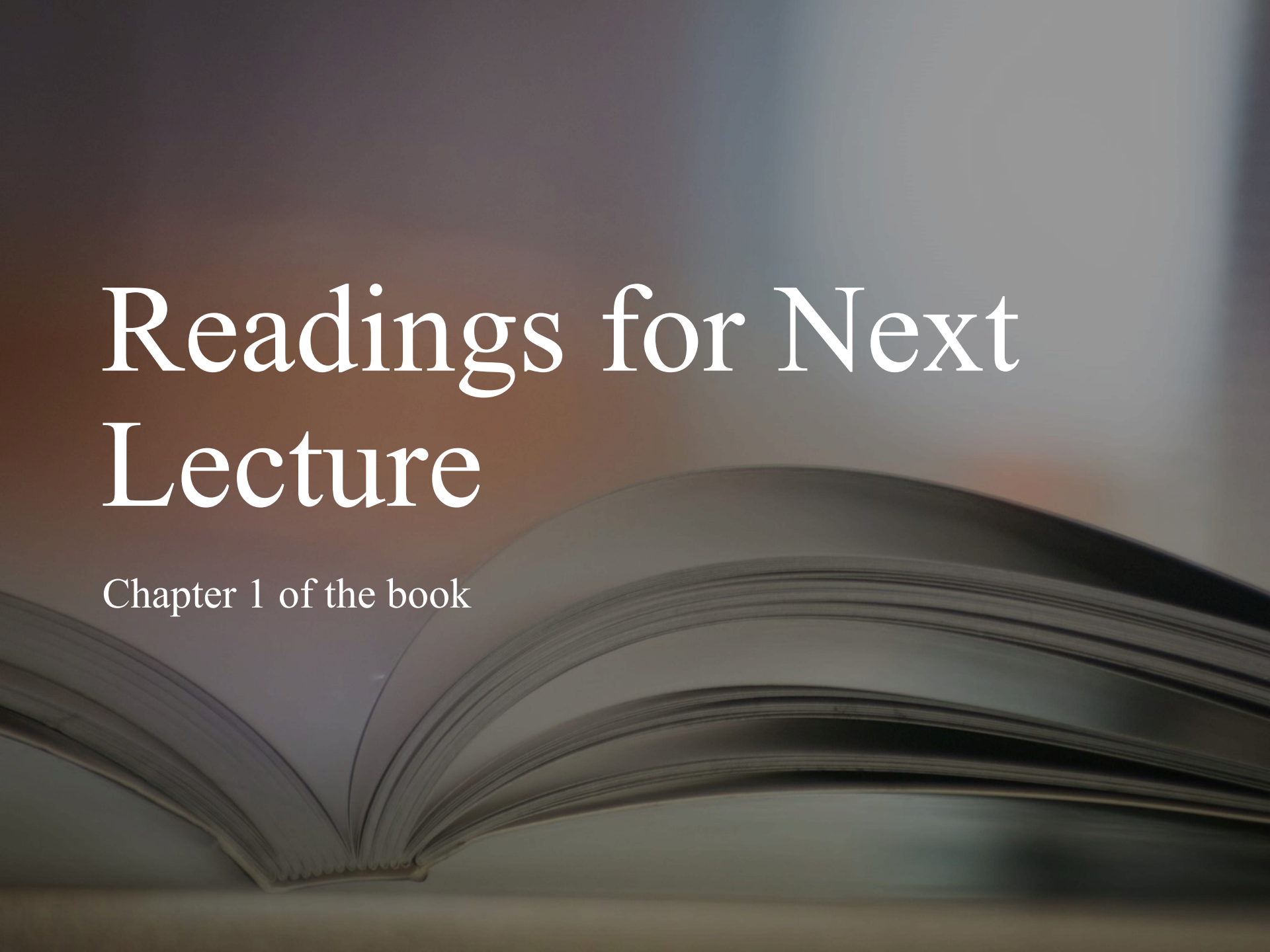
Possibilities include bounties for first person to solve vexing problems

But: you must never misrepresent work done by someone else as your own, without proper credit

OK to share ideas & help each other out, but your project should ultimately be created by your group & solo homework / tests should be your own

Readings for Next Lecture

Chapter 1 of the book



THANKS
SEE U NEXT LECTURE