Introduction

1. Objective

- What is computer science?
- What is a computer?
- What are the main components of computer hardware?
- What is science?
- Where is the science in computer science?
- What is an algorithm?
- What is software development?
- What is computer programming?
- What is a computer programming language?

2. Computing History

- The Von Neumann architecture is still the key model: CPU, memory, long-term storage, communication bus, and peripherals
- First computers were hard-wired, had to physically exchange cables to create different programs
- Early systems included ENIAC, ILLIAC, and UNIVAC (late 1940's)
- Machines cost hundreds of thousands of dollars, require tens of thousands of vacuum tubes and relays, huge space requirements, breakdowns frequent

3. Computer Taxonomy

- Supercomputers:
  - The world fastest Mid 1970's Cray Research develops the CRAY I
  - Parallel operation

- Mainframes:
  - Still large enough to fill a small room
  - Best at large amounts of I/O
  - Supports 50 to 1000 simultaneous users
  - Massive storage

- Minicomputer:
  - Can fit in a closet
  - Supports 2 to 50 simultaneous users

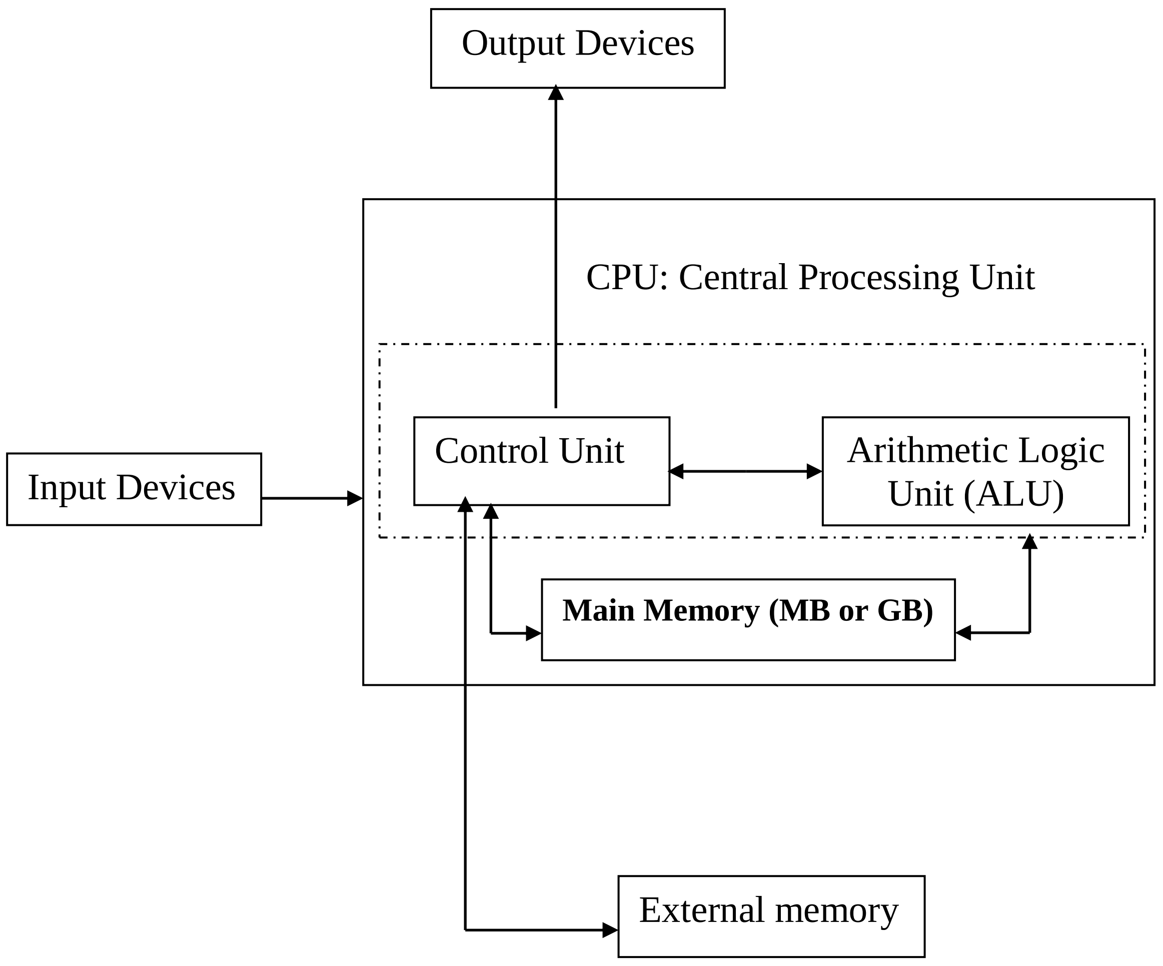
- Microcomputer:
  - Workstations, Sun
  - Desktop sized
  - Generally supports one user
  - Portables, laptops, notebooks

- Personal Digital Assistant (PDA) or Handheld Computer:
  - Palm, Visor, TRG,
  - Palm size
  - More than just an organizer
  - Special-purpose computers
  - Embedded

- Dedicated use
- Not easily programmable, firmware

4. Components of a computer

- Main memory
- Secondary memory
- Central Processor unit
- I/O devices
- Computer networks



- Bit - an acronym for Binary Digit. A bit is either 0 or 1.

- Byte - a group of 8 bits. 00000000 through 11111111.

- Counting in binary and decimal.

Dec	Binary
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000

- Central Processing Unit – CPU
    - The component that executes instructions in order to process input data and generate output.

- Arithmetic Logic Unit-ALU.
    - Performs arithmetic and logical operations. Stores data temporarily in an accumulator or registers.

- Control Unit

        - Responsible for retrieving instructions to be executed, retrieving necessary data, and sending both to the ALU for processing.

- Secondary Storage
    - Stores information that needs to be retrieved later. Different than primary storage (RAM).
    - Additional Storage (larger)
    - Permanent
    - Examples: Hard Disks, CD-ROMs, Etc.

5. Computer Software

- Software is instructions and associated data, stored in electronic format, that direct the computer to perform some task.

- Categories of software:
    - System
    - Applications

- Categories of Systems software
    - Operating systems
    - Utility programs

- Categories of Applications software
    - Productivity software
    - Education
    - Entertainment

6. Computer Networks

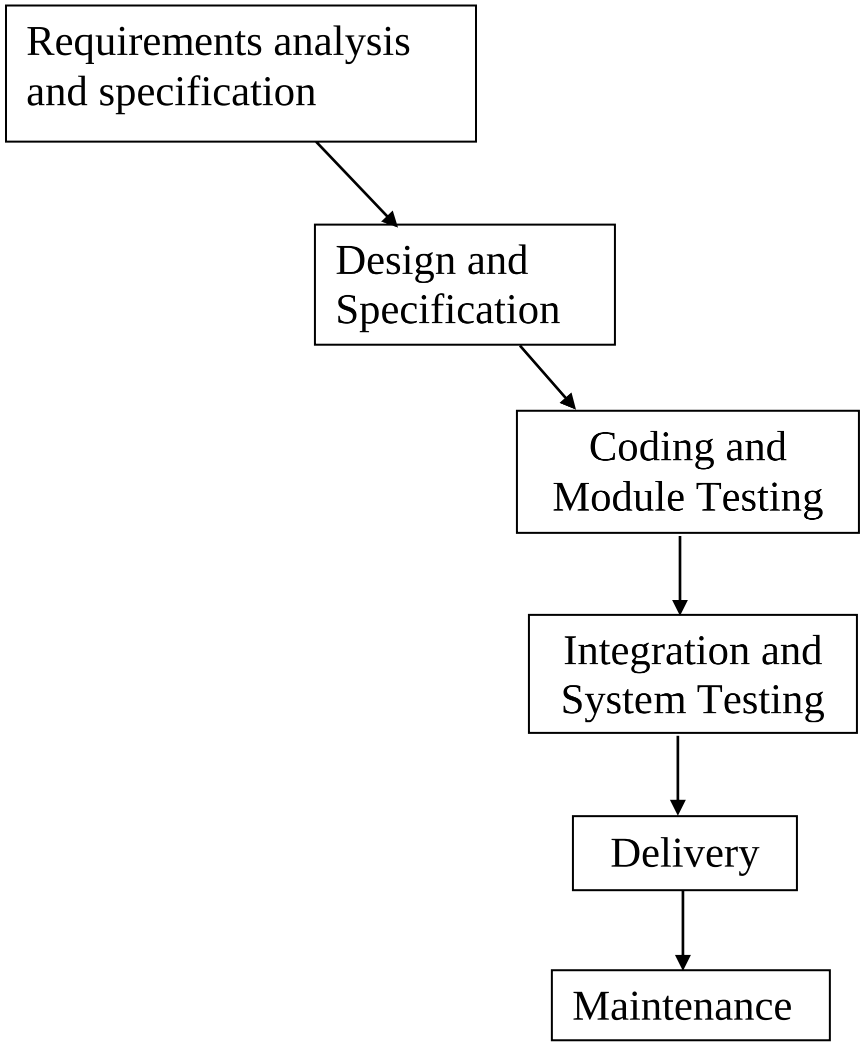
- Why Network?
- Classifications of Networks: LAN, WAN, Etc.
- Connecting to the Network

7. Problem solving and programming:

- It is a very challenging task
- It takes a plain English description of a problem and transforms it to a digital computer solution
- Successful solutions require the following:
    - Capture the requirements of the problem right from the beginning
    - Have a clear definition of what is the set of inputs and what is the set of outputs
    - Limit the scope of the problem: Do not solve the universal problem.
    - Use any information about the problem: formulas, equations, etc.

- Have a complete test plan for your solution.

8. The Software Development method


Requirements analysis and specification
Design and Specification
Coding and Module Testing
Integration and System Testing
Delivery
Maintenance

9. Programming languages

- A Programming language is a notational system for describing tasks/computations in a machine and human readable form.

- Most computer languages are designed to facilitate certain operations and not others: numerical computation, or text manipulation, or I/O.

- More broadly, a computer language typically embodies a particular *programming paradigm*.

- Every language has syntax and semantics:

    - **Syntax**: The syntax of a program is the form of its declarations, expressions, statements and program units.

    - **Semantic**: The semantic of a program is concerned with the meaning of its program.

- Machine language:
    - The lowest level language: The language of the CPU.
    - It consists of binary strings that represent Instructions: 0's and 1's.

- Assembly languages:

    - Abbreviations for machine language

- High-level languages:
    - Use program statements - words and algebra-type expressions.
    - Developed in the 50's and 60's.
    - After a program is written in one of the high-level languages, it must be either **compiled** or **interpreted**.

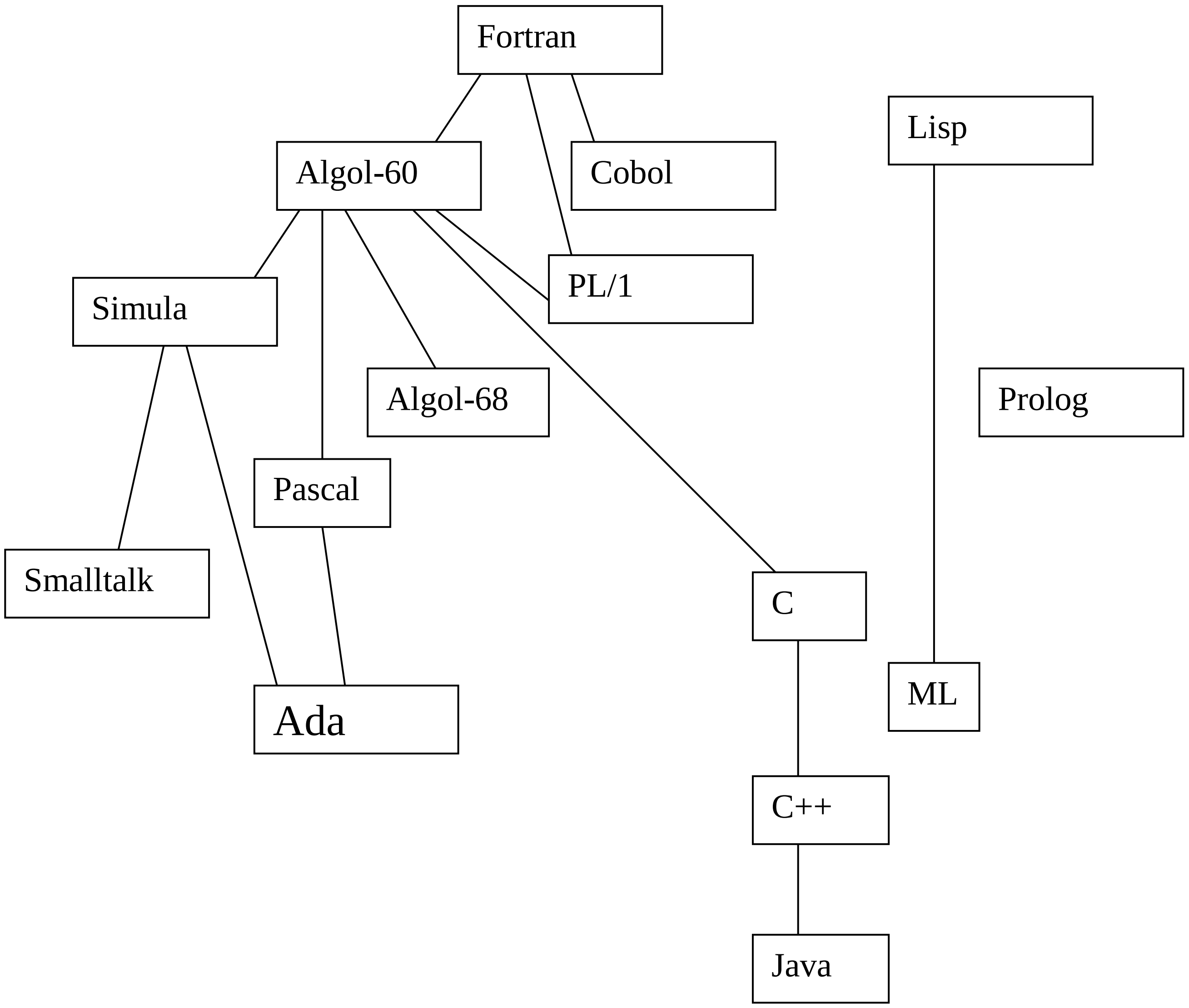
10. Programming paradigms

- The paradigms are not exclusive, but reflect the different emphasis of language designers. Most practical languages embody features of more than one paradigm.

- **Classification:**

Imperative/ Algorithmic	Declarative		Object-Oriented
	Functional Programming	Logic Programming	
Algol Cobol PL/1 Ada C Modula-3	Lisp Haskell ML Miranda APL	Prolog	Smalltalk Simula C++ Java

- Language History:



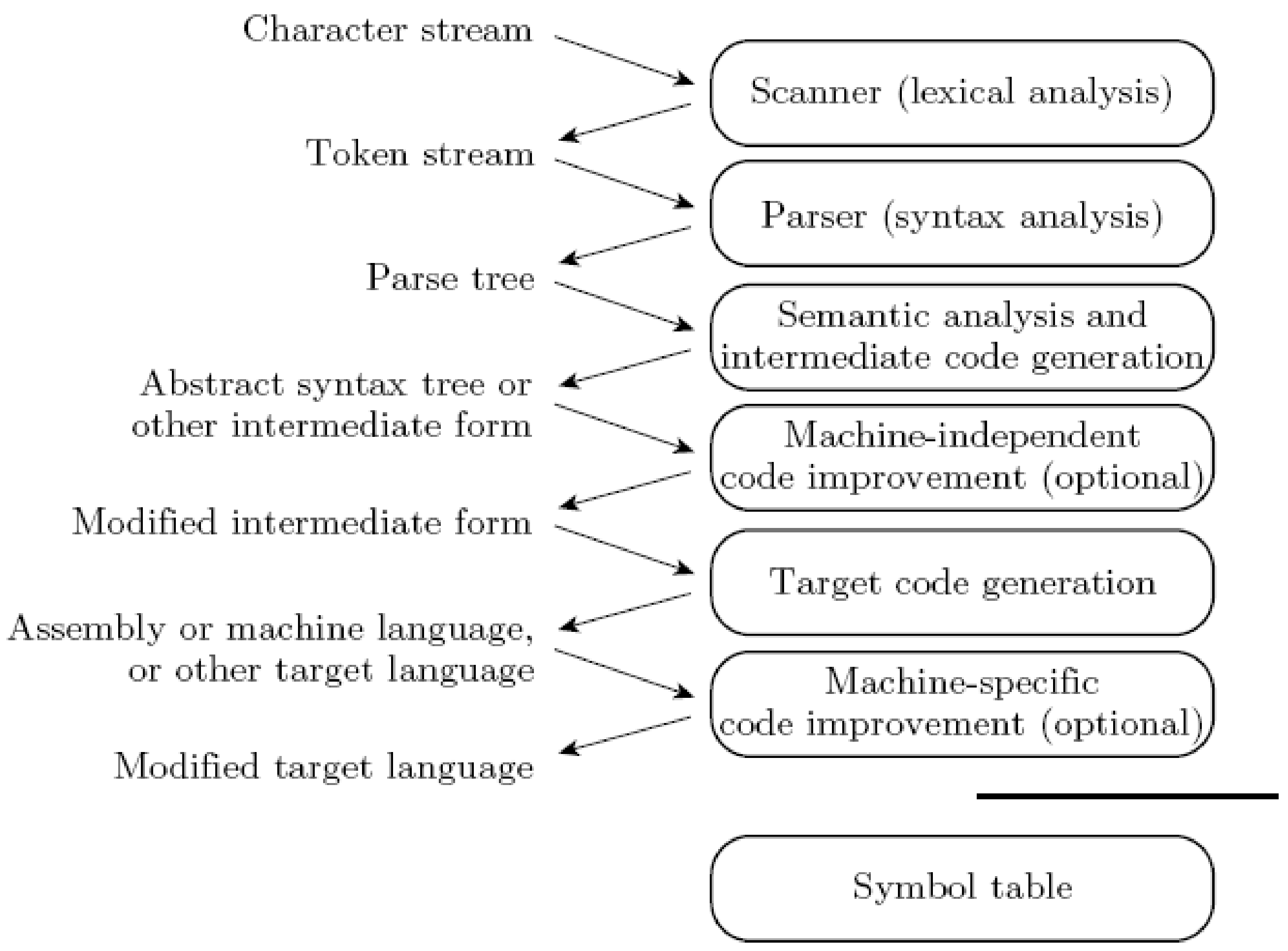
11. Imperative paradigms

- It is based on commands that update variables in storage. The Latin word *imperare* means “to command”.
    - The language provides statements, such as *assignment statements*, which explicitly change the *state* of the memory of the computer.
    - This model closely matches the actual executions of computer and usually has high execution efficiency.
- Many people also find the imperative paradigm to be a more natural way of expressing themselves.

12. Running Your Program

- **Interpreter:**
    - An **interpreter** program translates the program statements into machine language one line at a time
- **Compiler**:
    - A **compiler** program rewrites the program into machine language that the CPU can understand. This is done all at once and the program is saved in this new form. A compiled program is generally considerably larger than the original.

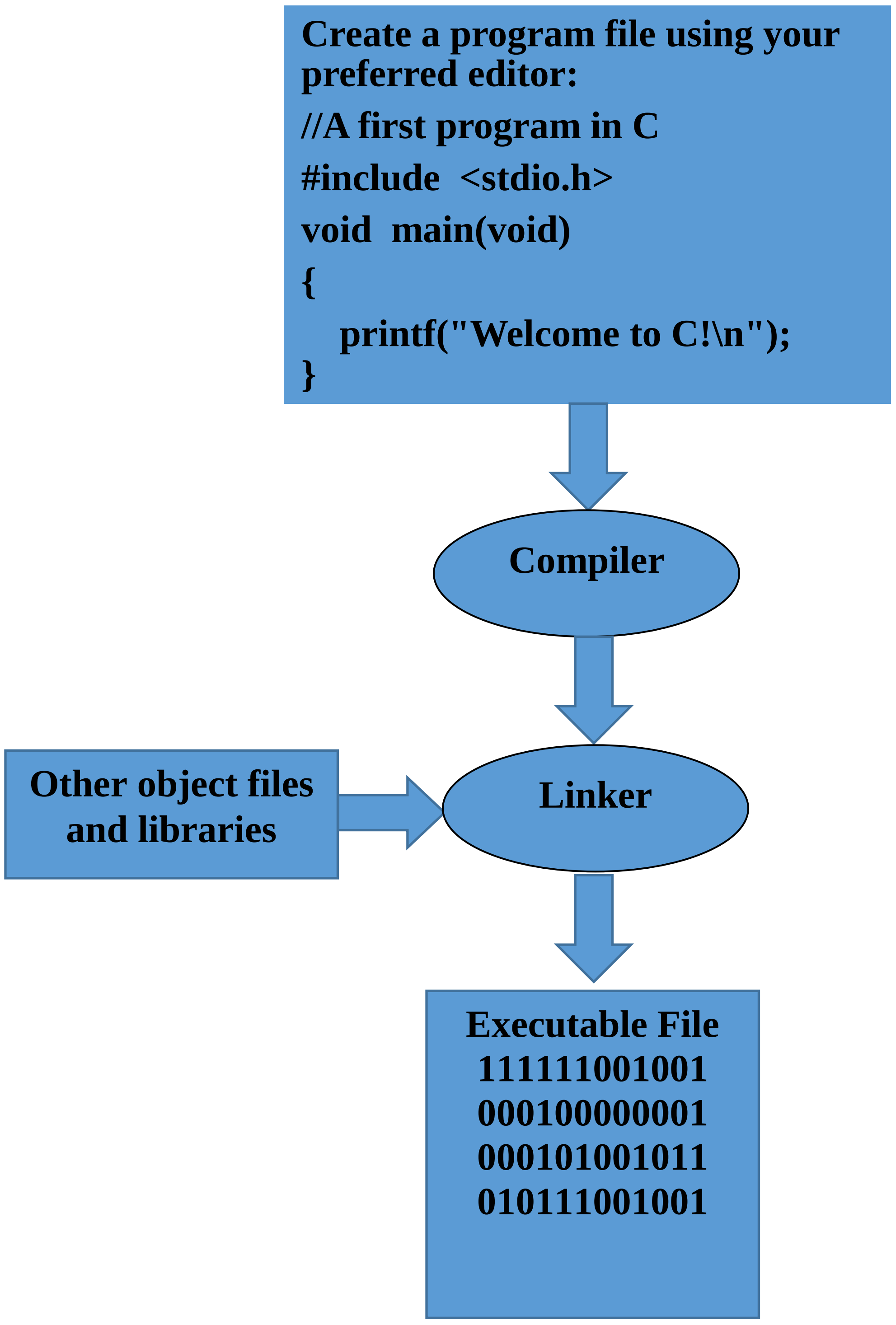
- Phase of Compilation (From Scott's class notes)


Character stream
Scanner (lexical analysis)
Token stream
Parser (syntax analysis)
Parse tree
Semantic analysis and intermediate code generation
Abstract syntax tree or other intermediate form
Machine-independent code improvement (optional)
Modified intermediate form
Target code generation
Assembly or machine language, or other target language
Machine-specific code improvement (optional)
Modified target language
Symbol table

13. Processing of a high-level language program

- What is a computer program?
    - **A computer program** is a set of detailed directions telling the computer exactly what to do, one step at a time. A program can be as short as one line of code, or as long as several millions lines of code.

- Steps to execute a program:



Programming Errors

- **Syntax** Errors: You need to speak the language!
- **Run Time** Errors: illegal operations?
- **Semantic** Errors: You need to review your algorithm!