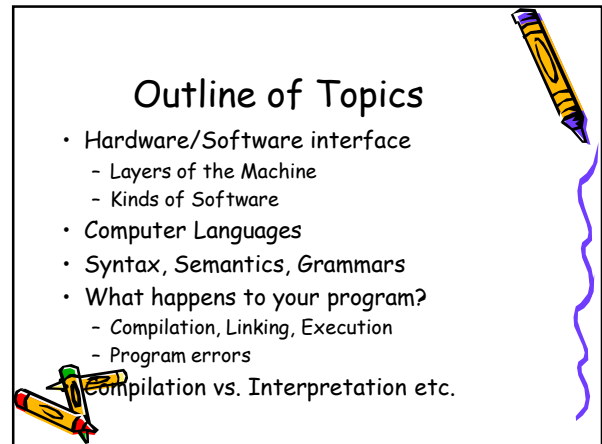
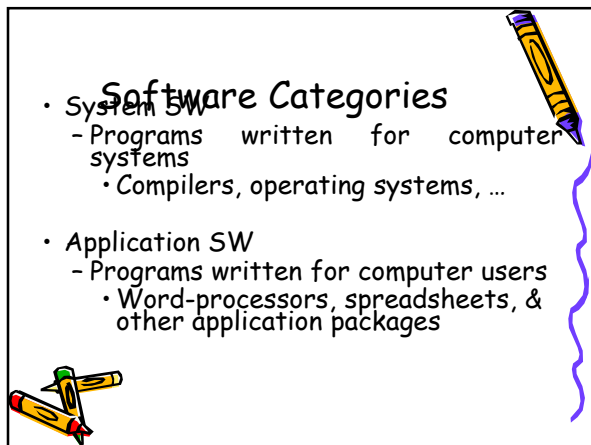
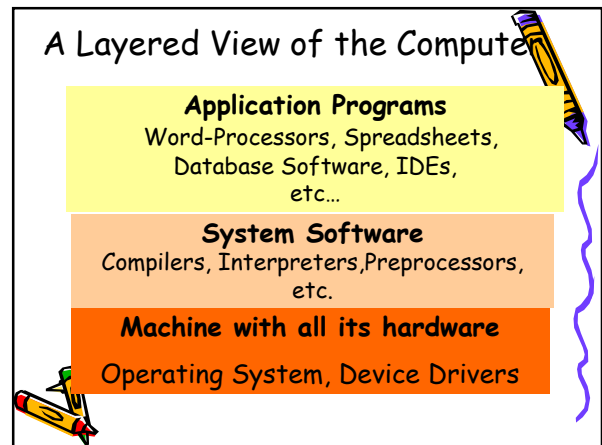
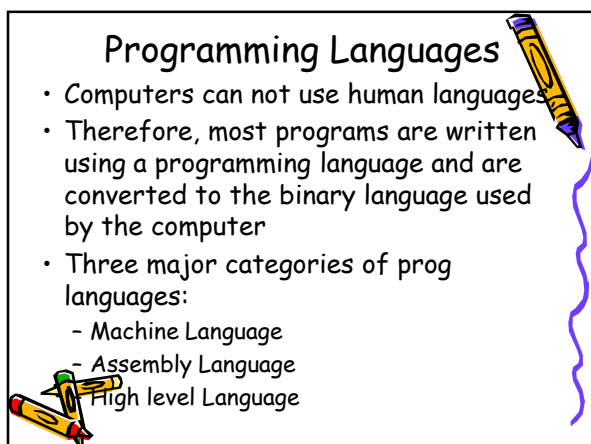


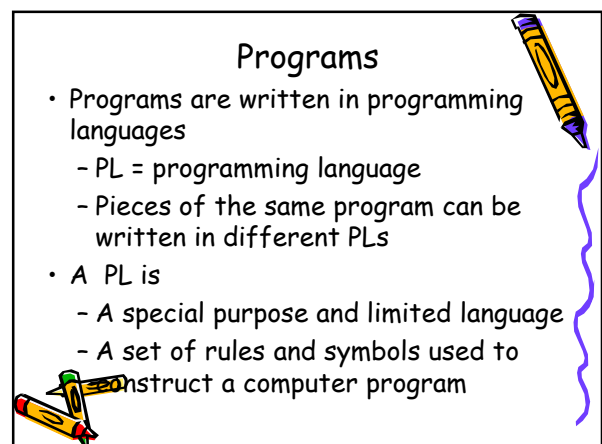
Introduction to Computer Programs

- 
- ## Outline of Topics
- Hardware/Software interface
 - Layers of the Machine
 - Kinds of Software
 - Computer Languages
 - Syntax, Semantics, Grammars
 - What happens to your program?
 - Compilation, Linking, Execution
 - Program errors
 - Compilation vs. Interpretation etc.

- 
- ## Software Categories
- System SW
 - Programs written for computer systems
 - Compilers, operating systems, ...
 - Application SW
 - Programs written for computer users
 - Word-processors, spreadsheets, & other application packages




- 
- ## Programming Languages
- Computers can not use human languages
 - Therefore, most programs are written using a programming language and are converted to the binary language used by the computer
 - Three major categories of prog languages:
 - Machine Language
 - Assembly Language
 - High level Language

- 
- ## Programs
- Programs are written in programming languages
 - PL = programming language
 - Pieces of the same program can be written in different PLs
 - A PL is
 - A special purpose and limited language
 - A set of rules and symbols used to construct a computer program

Computer Languages

- Machine Language
 - Natural language of a particular computer
 - Primitive instructions built into every computer
 - The instructions are in the form of binary code
 - Any other types of languages must be translated down to this level
 - Machine-dependent


Not portable



Assembly Language

- English-like abbreviations representing elementary computer operations.
- The computer cannot understand assembly language -
- a program called assembler is used to convert assembly language programs into machine code
- **Example:**


Load	Price
Add	Tax
Store	Cost



High-level language


- A programming language which use statements consisting of English-like keywords such as "FOR", "PRINT" or "IF", ... etc.
- Much easier to program than in assembly language.
- Data are referenced using descriptive names
- Operations can be described using familiar symbols
- Example:

Cost := Price + Tax




Compiling Source Code

- A program written in a high-level language is called a *source program* (or *source code*).
- Program called a *compiler* is used to translate the source program into a machine language program called an *object program*.
- The object program is often then linked with other supporting library code before the object can be executed on the machine.




10




Syntax & Semantics

- Syntax:
 - The structure of strings in some language. A language's syntax is described by a grammar.
 - Examples:
 - Binary number
 - <binary_number> = <bit> | <bit>
 - <binary_number>
 - <bit> = 0 | 1
 - Identifier
 - <identifier> = <letter> { <letter> | <digit> }
 - <letter> = a | b | ... | z
 - <digit> = 0 | 1 | ... | 9
- Semantics:
 - The meaning of the language



Syntax & Grammars

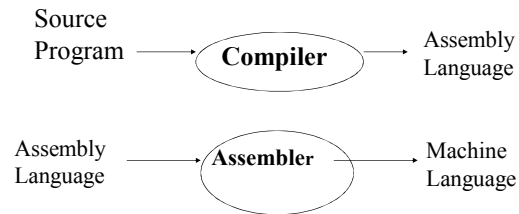
- Syntax descriptions for a PL are themselves written in a formal language.
- The formal language is not a PL but it can be implemented by a compiler to enforce grammar restrictions.
- Some PLs look more like grammar descriptions than like instructions.



Compilers & Programs

- **Compiler**
 - A program that converts another program from some source language (or high-level programming language / HLL) to machine language (object code).
 - Some compilers output assembly language which is then converted to machine language by a separate assembler.

Compilation into Assembly L



Compilers & Programs

- **Source program**
 - The form in which a computer program, written in some formal programming language, is written by the programmer.
 - Can be compiled automatically into machine code or executed by an interpreter.

Compilers & Programs

- **Object program**
 - Output from the compiler
 - Equivalent machine language translation of the source program
- **Executable program**
 - Output from linker/loader
 - Machine language program linked with necessary libraries & other files
 - Files usually have extension '.exe'

What is a Linker?


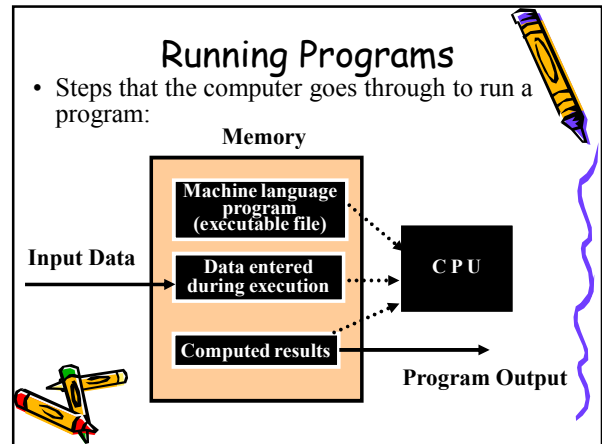
- A program that pulls other programs together so that they can run.
- Most programs are very large and consist of several modules.
- Even small programs use existing code provided by the programming environment called libraries.
- The linker pulls everything together, makes sure that references to other parts of the program (code) are resolved.

Compilers Vs Interpreters

- **Compilers**
 - Translate the program before it's executed.
 - When programs are compiled, they are translated all at once.
 - Compiled programs typically execute more quickly than interpreted programs, but have a slower translation speed


Compilers Vs Interpreters

- Interpreters
- Translate programs line-by-line instead of all at once (like compiled programs).
- Interpreted programs generally translate quicker than compiled programs, but have a slower execution speed.


Program Execution

- Steps taken by the CPU to run a program (instructions are in machine language):
 1. Fetch an instruction
 2. Decode (interpret) the instruction
 3. Retrieve data, if needed
 4. Execute (perform) actual processing
 5. Store the results, if needed



Program Errors

- Syntax Errors:
 - Errors in grammar of the language
- Runtime error:
 - When there are no syntax errors, but the program can't complete execution
 - Divide by zero
 - Invalid input data
- Logical errors:
 - The program completes execution, but delivers incorrect results
 - Incorrect usage of parentheses



Questions

