

CSC/COS 111/101

LECTURE 4

SOFTWARE

ZUBAIR ADAM
Dr. Zubair Adam

OVERVIEW

- ▣ ***Introduction to & History of Computer***
- ▣ ***Basic Computer Configuration***
- ▣ ***Hardware (Input, Storage and Output Devices)***
- ▣ Software (Operating System, Application Software, etc.)
- ▣ Internet
- ▣ Using personal computers as effective problem-solving tools for the present and future.
- ▣ Computer application areas and technological trends.



Introduction to Software and Programming Languages

□ System Software

- Operating Systems
- Types of Operating Systems
- Language Translators
- Utility Software

□ Application Software

- Word Processing Packages
- Spread sheet Packages
- Graphic Packages
- Database Packages
- Desktop Publishing Packages
- Games Packages
- Communication Packages
- User Programs
- Integrated Packages

□ Computer Languages

- Machine Languages
- Low-Level Languages
- High-Level Languages

Computer Software

- Computer software is the set of programs that makes the hardware perform a set of tasks in particular order.
- Hardware and software are complimentary to each other. Both have to work together to produce meaningful results.
- Computer software is classified into two broad categories;
 - System Software and
 - Application Software.

A Layered View of the Computer

Hardware/Software interface

Application Programs

Word-Processors, Spreadsheets,
Database Software, IDEs,
etc...

System Software

Compilers, Interpreters, Preprocessors, etc.
Operating System, Device Drivers

Machine with all its hardware

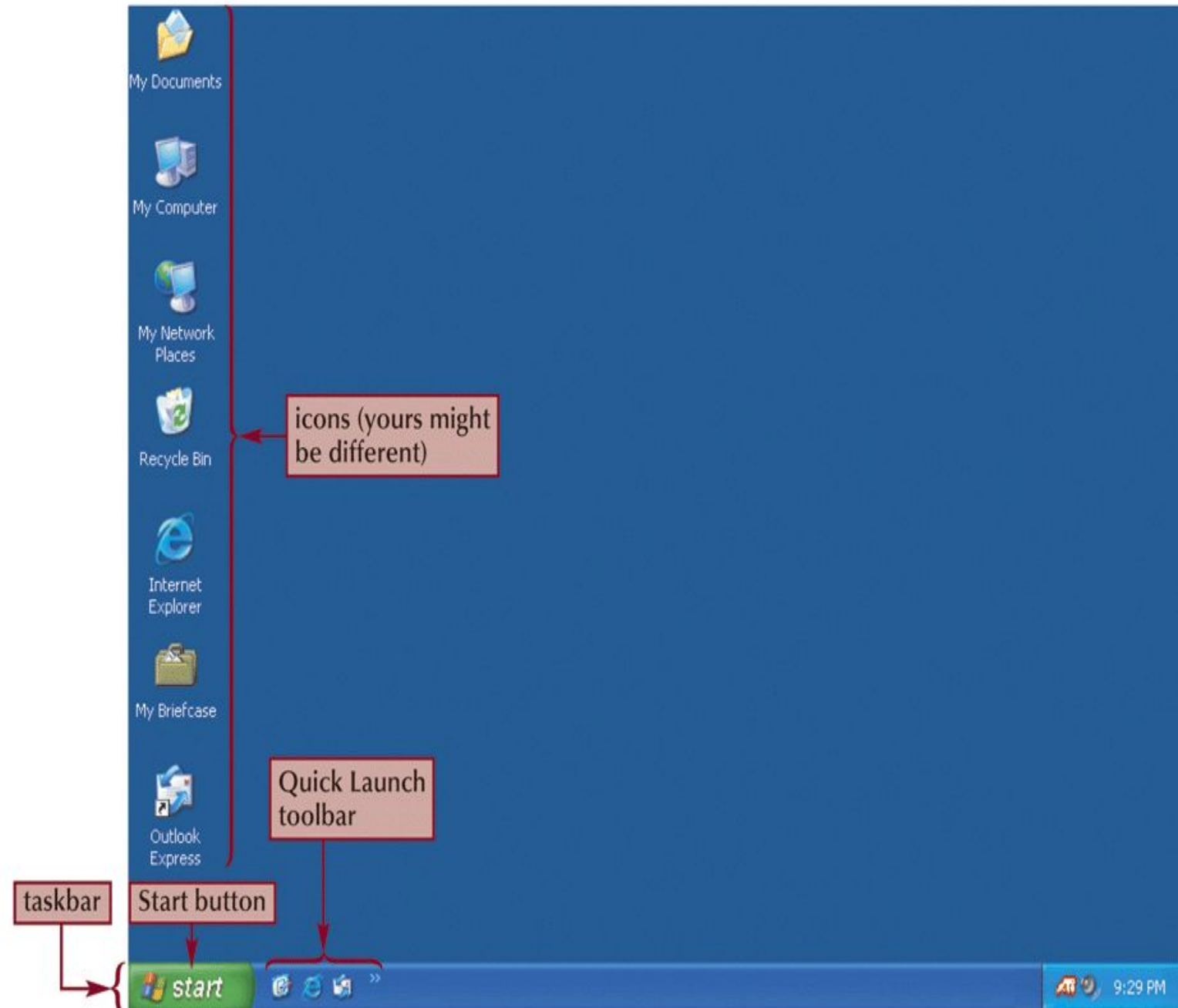
Systems Software

- Set of programs that coordinates activities and functions of the hardware and various other programs.
- The software that controls everything that happens in a computer.
- Background software, manages the computer's internal resources
 - Programs written for computer systems
 - **Compilers, operating systems, ...**
- System software consists of a group of programs that control the operations of a computer equipment including functions
 - managing memory,
 - managing peripherals,
 - loading, storing,
 - an interface between the application programs and the computer.

- MS DOS (Microsoft's Disk Operating System), UNIX are examples of system software
- **System Software manages the fundamental operations of your computer**
 - Operating system
 - System resource
 - Multitasking
 - Utilities
 - Programming Languages

Figure 26

Windows XP starting screen

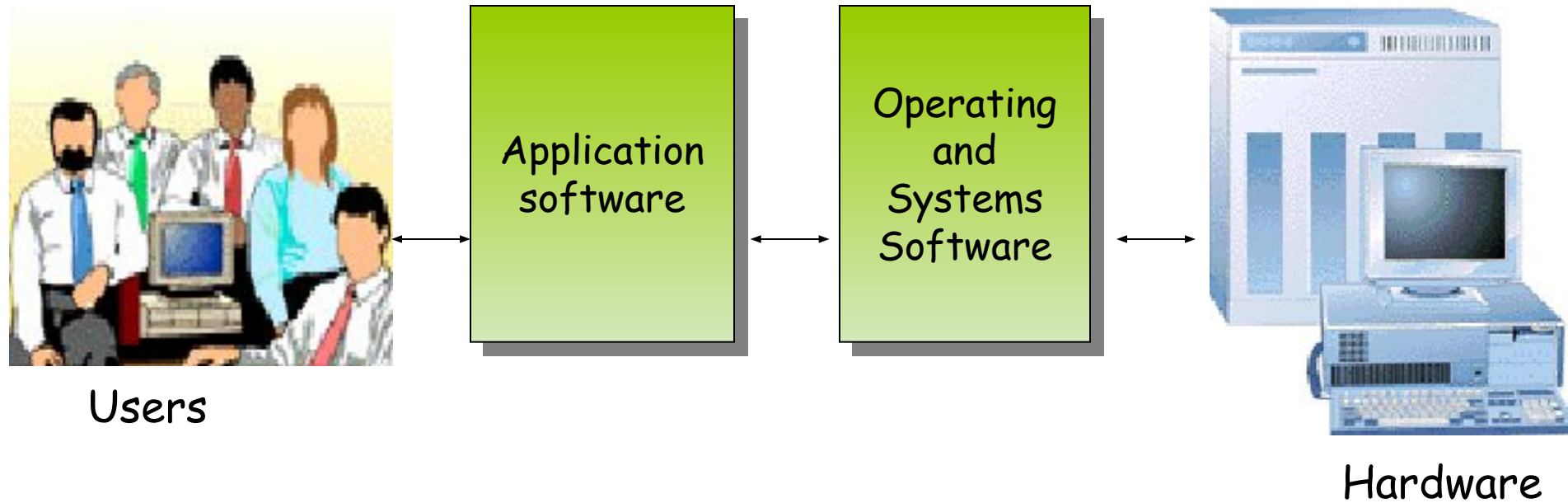


Operating Systems or OS

- The operating system (OS) controls all other software and allows the hardware devices to work properly.
- Without an operating system, a user cannot run an application program on their computer, unless the application program is self booting.
- Some popular operating systems are:
 - **Microsoft Windows** - for PCs
 - **Mac OS** - for Apple computers
 - **Linux** - for very large network computers
 - **Handheld operating systems** - for PDAs, MP3 players, and cell phones

- Is software, consisting of programs and data, that runs on computers, manages computer hardware resources, and provides common services for execution of various application software.
- The OS is the most important type of system software in a computer system.
- Provides several essential services:
 - Loading & running application programs
 - Allocating memory & processor time
 - Providing input & output facilities
 - Managing files of information

Role of the Operating System



Operating System Functions

An operating system is primarily a resource manager

Design is tied to the hardware and software resources the operating system must manage

- Perform common computer hardware functions
 - Ex: Get input from keyboard
- Provide a user interface
 - Command-based user interface (ms-dos)
 - Graphical-user interface (windows)
- Provide a degree of hardware independence
 - Application program interface
- Manage system memory
 - Convert logical view to physical view

Operating System Functions

- Manage processing tasks
 - Allocate computer resources
 - Multitasking (run more than once application at a time)
 - Time-sharing (allow multiple access to a system)
- Provide networking capability
 - Enable connection to the Internet
- Control access to system resources
 - Authentication
- Manage files
 - Access to files

- Perform common computer hardware functions:
Processors, secondary storage (such as hard disks), other I/O devices, processes, threads, databases
- Provide a user interface
- Provide a degree of hardware independence
- Manage system memory
- Manage processing tasks
- Provide networking capability
- Control access to system resources
- Manage files

Types of Operating Systems

Today's computers are:

1. **Multi-user** - A multi-user operating system allows for multiple users to use the same computer at the same time and different times.
2. **Multiprocessing** - An operating system capable of supporting and utilizing more than one computer processor.
3. **Multitasking** - An operating system that is capable of allowing multiple software processes to run at the same time.
4. **Multithreading** - Operating systems that allow different parts of a software program to run concurrently.

Programs

❖ Programs are written in programming languages

- ✧ PL = programming language
- ✧ Pieces of the same program can be written in different PLs
 - Languages closer to the machine can be more efficient
 - As long as they agree on how to communicate

❖ A PL is

- ✧ A special purpose and limited language
- ✧ A set of rules and symbols used to construct a computer program
- ✧ A language used to interact with the computer

Language Translators

- A programming language (PL) is a vocabulary and set of grammatical rules for instructing a computer or computing device to perform specific tasks.
- PL refers to high-level languages, such as BASIC, C, C++, COBOL, Java, FORTRAN, Ada, and Pascal.
- Each PL has a unique set of keywords (words that it understands) and a special syntax for organizing program instructions.
- A programmer uses PL to develop software programs, scripts, or other sets of instructions for computers to execute.
- PL share similarities, each has its own syntax.

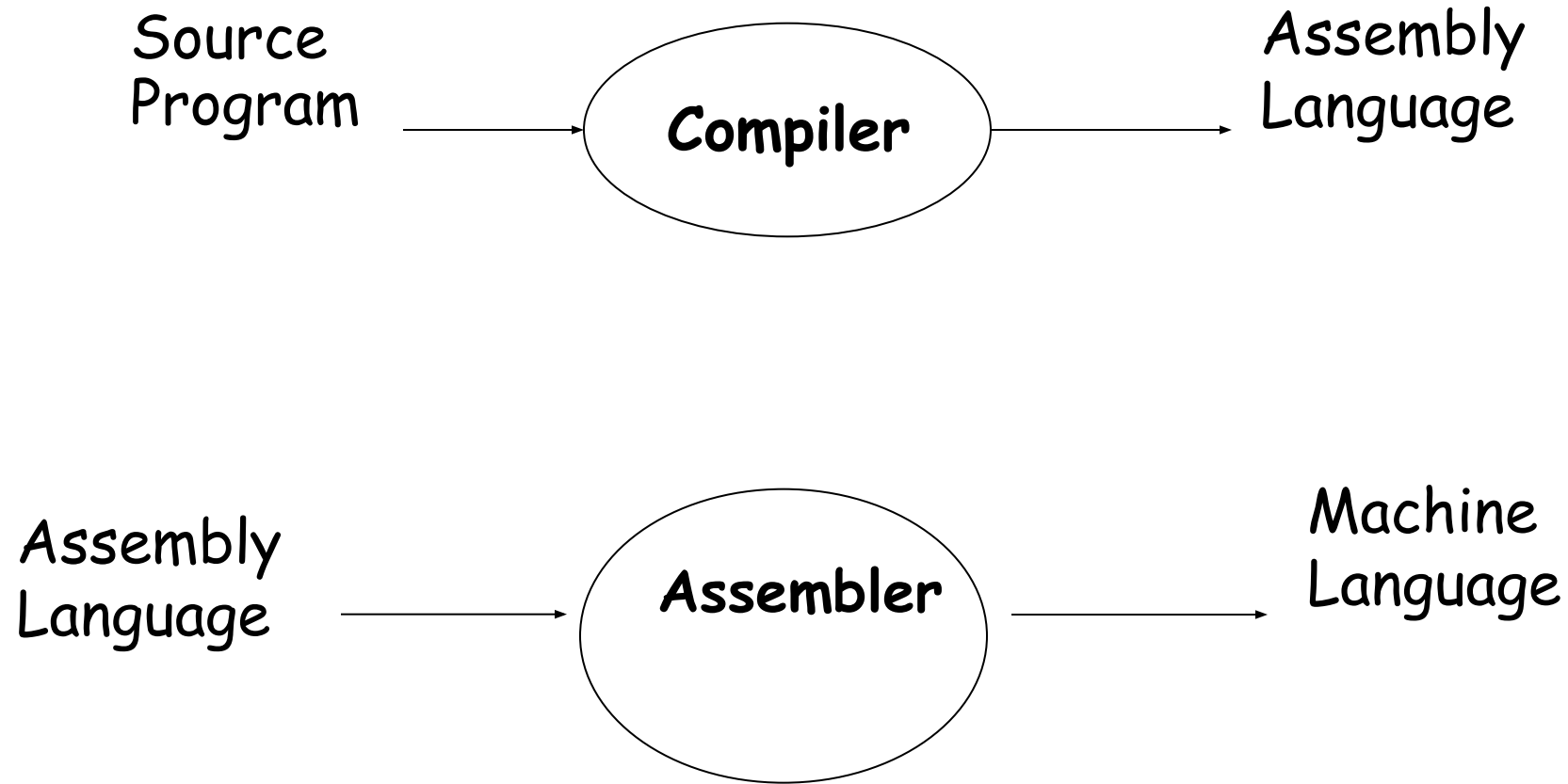
Language Translators

- A programmer learns the languages rules, syntax, and structure, they write the source code in a text editor or IDE.
- The programmer often compiles the code into machine language that can be understood by the computer.
- Scripting languages, which do not require a compiler, use an interpreter to execute the script.

Assemblers

- An assembler is a program that converts assembly language into machine code.
- It takes the basic commands and operations from assembly code and converts them into binary code that can be recognized by a specific type of processor.
- Assemblers are similar to compilers in that they produce executable code.
- However, assemblers are more simplistic since they only convert low-level code (assembly language) to machine code.

Compilation into Assembly Language



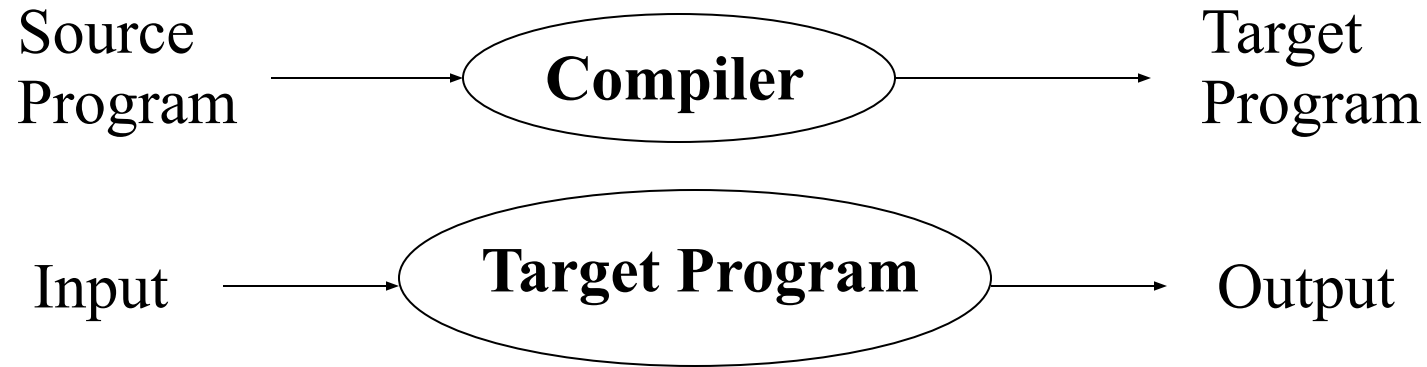
Compilers

- A compiler is a software program that transforms high-level source code that is written by a developer in a high-level programming language into a low level object code (binary code) in machine language, which can be understood by the processor.
- The process of converting high-level programming into machine language is known as compilation.
- The processor executes object code, which indicates when binary high and low signals are required in the arithmetic logic unit of the processor.
- The stages in compilation include:
 - Lexical Analysis
 - Syntax Analysis
 - Semantic Analysis
 - Intermediate code generation
 - Code optimization
 - Code generation

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.
- Is distinguished from an assembler by the fact that each input statement, in general, correspond to more than one machine instruction.

Compilation



- Compiler translates source into target (a machine language program)
- Compiler goes away at execution time
- Compiler is itself a machine language program, presumably created by compiling some other high-level program
- Machine language, when written in a format understood by the OS is **object code**

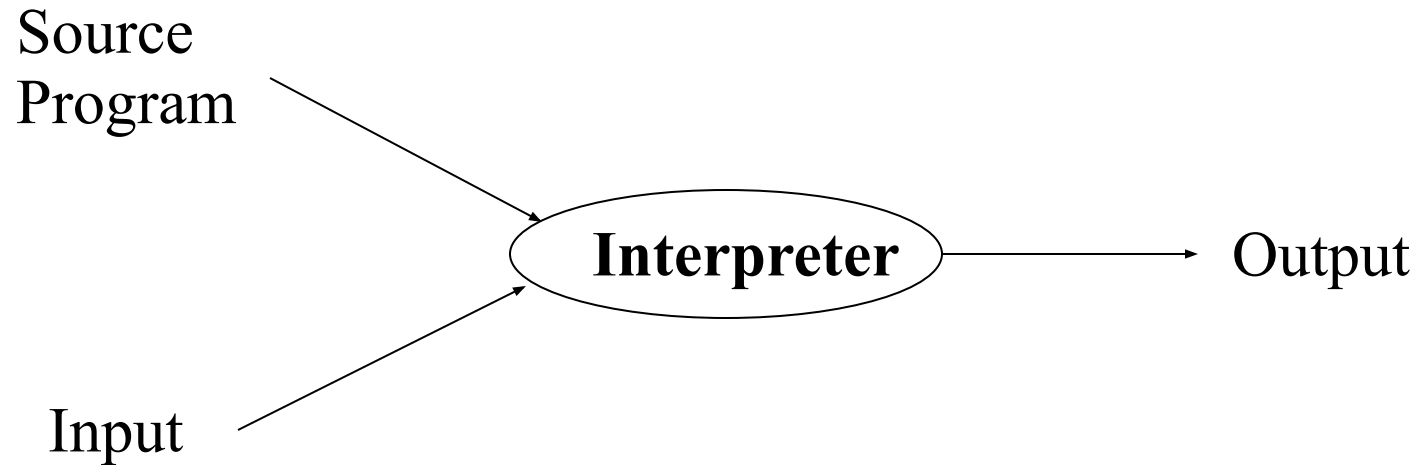
Interpreters

- An interpreter is a computer program that is used to directly execute program instructions written using one of the many high-level programming languages.
- The interpreter transforms the high-level program into an intermediate language that it then executes, or it could parse the high-level source code and then performs the commands directly, which is done line by line or statement by statement.
- Programming languages are implemented in two ways:
interpretation and compilation.

Interpreters

- An interpreter transforms or interprets a high-level programming code into code that can be understood by the machine (machine code) or into an intermediate language that can be easily executed as well.
- The interpreter reads each statement of code and then converts or executes it directly.
- In contrast, an assembler or a compiler converts a high-level source code into native (compiled) code that can be executed directly by the operating system.

Interpretation



- The interpreter stays around during execution
- It reads and executes statements one at a time

Compilation vs. Interpretation

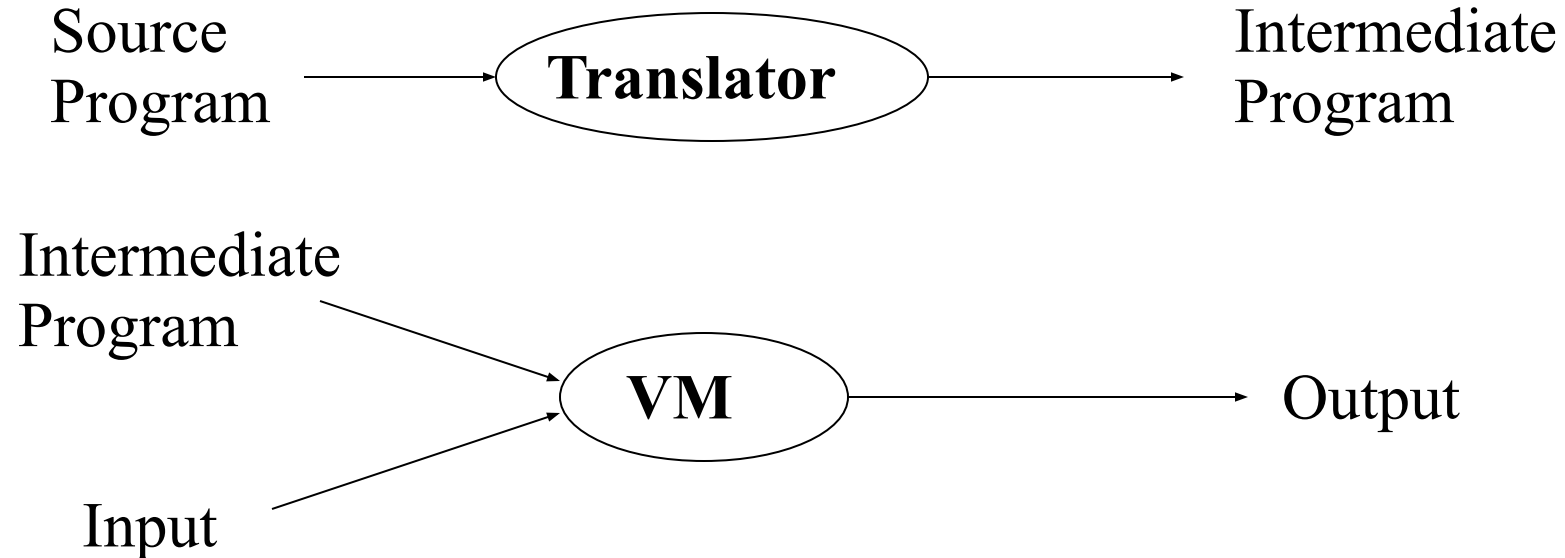
❖ Compilation:

- ✧ Syntax errors caught before running the program
- ✧ Better performance
- ✧ Decisions made once, at compile time

❖ Interpretation:

- ✧ Better diagnostics (error messages)
- ✧ More flexibility
- ✧ Supports **late binding** (delaying decisions about program implementation until runtime)
 - Can better cope with PLs where type and size of variables depend on input
- ✧ Supports creation/modification of program code on the fly (e.g. Lisp, Prolog)

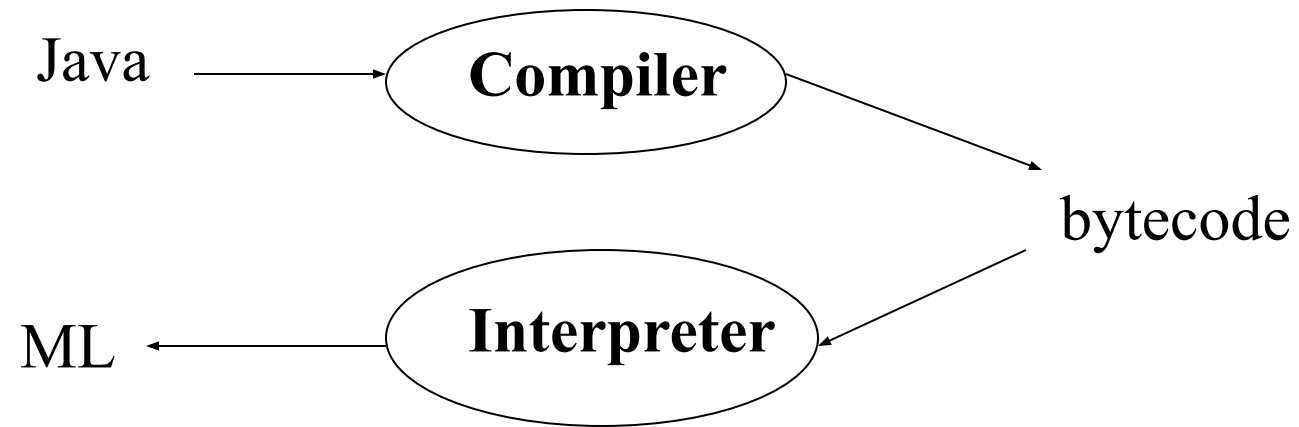
Mixture of C & I



- Many programming languages implement this
- Interpreter implements a Virtual Machine (VM).

JAVA

For portability:



For flexibility: Just In Time (JIT) compiler translates bytecode into ML just before execution

Utility Software

- It helps you maintain your computer and keep it in good running condition
- Utility software is software designed to help to analyse, configure, optimize or maintain a computer.
- It is used to support the computer infrastructure - in contrast to application software, which is aimed at directly performing tasks that benefit ordinary users.
- Utilities often form part of application systems.

For example a batch job may run user-written code to update a database and may then include a step that runs a utility to back up the database, or a job may run a utility to compress a disk before copying files.

□ File Conversion

File conversion is the process of converting a file into another type.

For example:

- transferring a file used in Microsoft Word to Corel Word Perfect.
- It overs data transfer from any medium to another medium, making an exact copy or simultaneously editing and Validating data.
- a copy from a hard disk to a diskette.

□ File Copy

File copying is the creation of a new file which has the same content as an existing file.

For example:

- File copying is the creation of a new copy file which has the same content as an existing file.

□ Housekeeping Operations

These include programs to clear areas of storage, writing file labels, updating common data e.g. data.

Application Software

- Programs that help users solve particular computing problems
- Programs written for computer users and it enables you to perform specific computer tasks, such as document production, spreadsheet calculations, and database management
 - Word-processors, spreadsheets, & other application packages
- Software that can perform a specific task for the user, such as
 - Word processors- **example: Microsoft word**
 - Spreadsheets-- **example: Microsoft Excel**
 - Database managers-- **example: Microsoft Access**
 - Graphics-- **example: Photoshop**

Application Software - Basic Tools: Examples

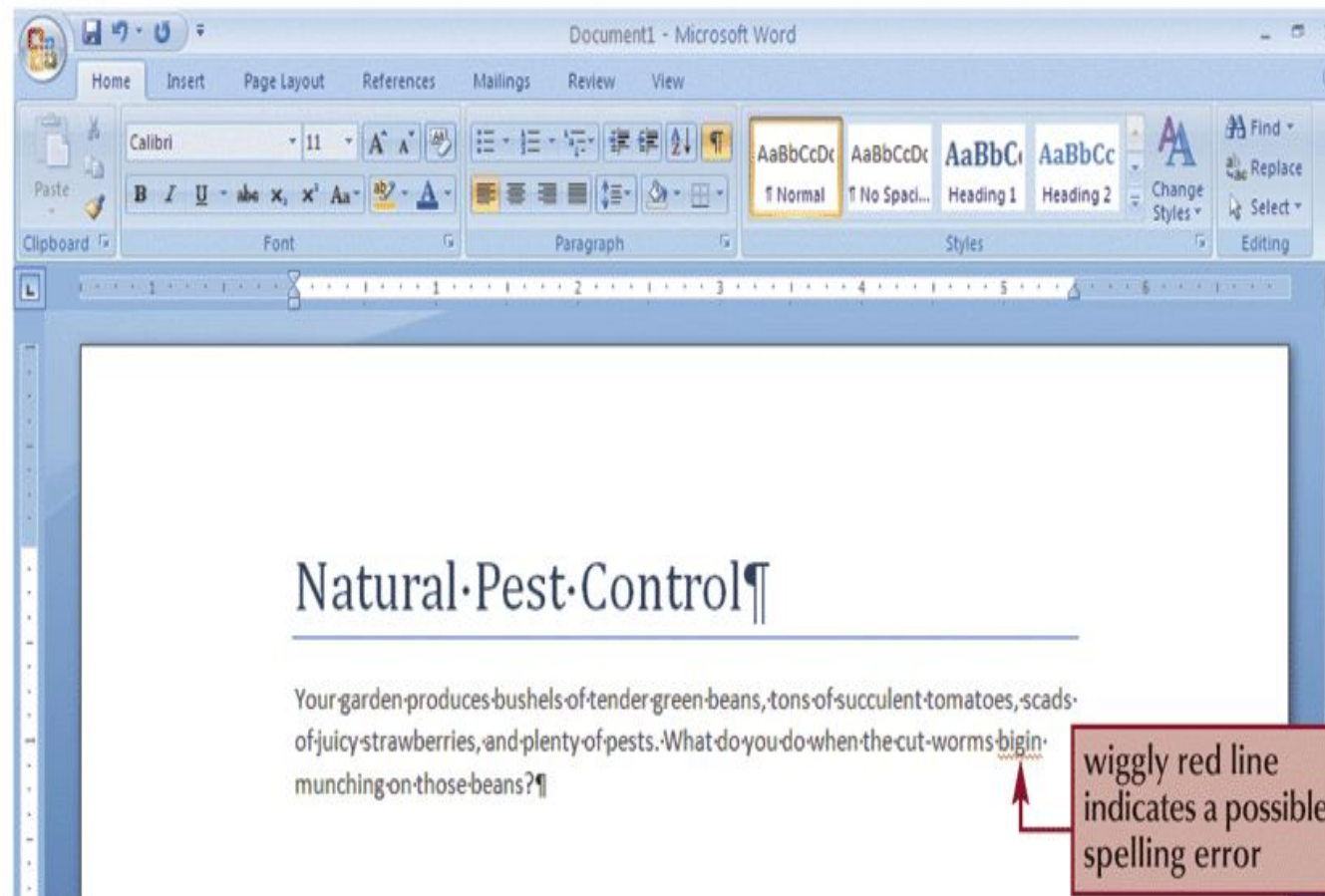
word processing, accounting, budgeting or payroll, fall under the category of application software. Word processors, spreadsheets, database management systems are all examples of general purpose application software

□ Word processing software:

- The main purpose of this software is to produce documents.
- MS-Word, Word Pad, Notepad and some other text editors are some of the examples of word processing software

Checking the spelling in a document

Figure 27

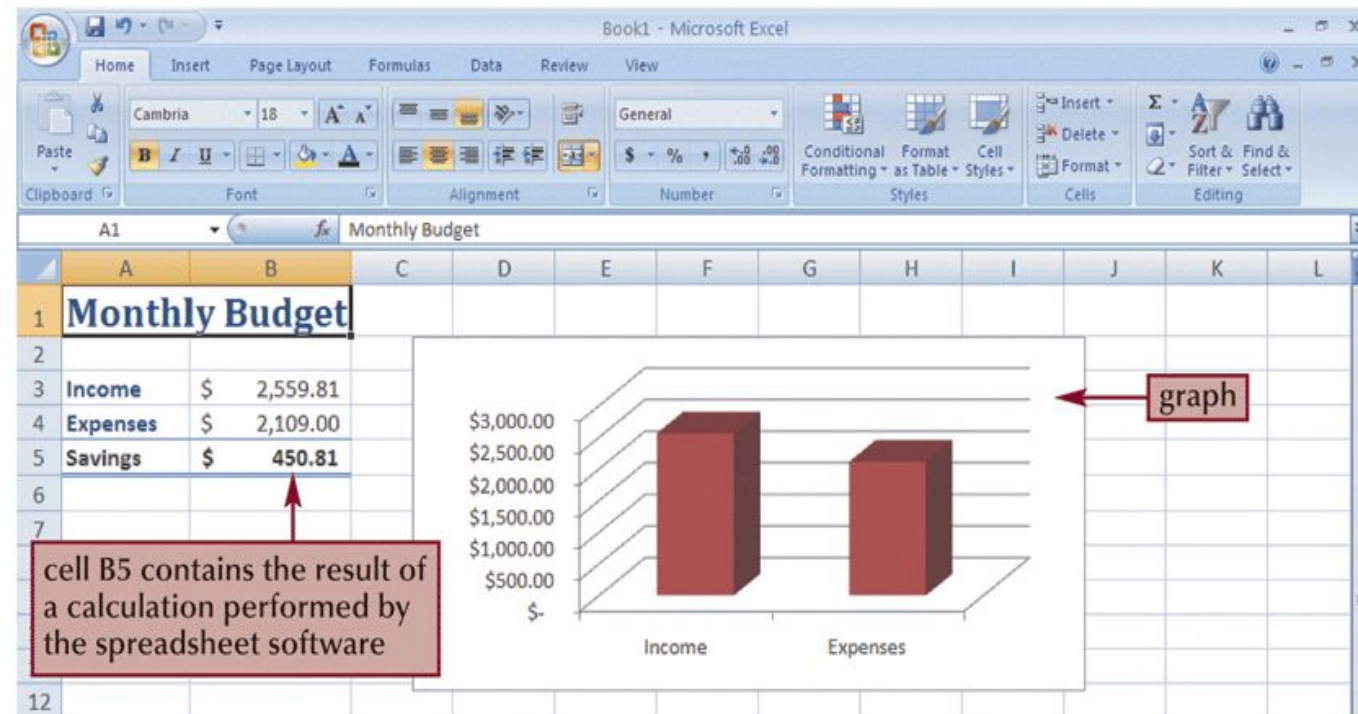


□ Spread sheet software:

- The spread sheet software is used to maintain budget, financial statements, grade sheets, and sales records.
- The purpose of this software is organizing numbers.
- It also allows the users to perform simple or complex calculations on the numbers entered in rows and columns.
- MS-Excel is one of the example of spreadsheet software.

Figure 28

Typical worksheet with numerical data and a graph



Assignment



- Look into the Storage Devices

