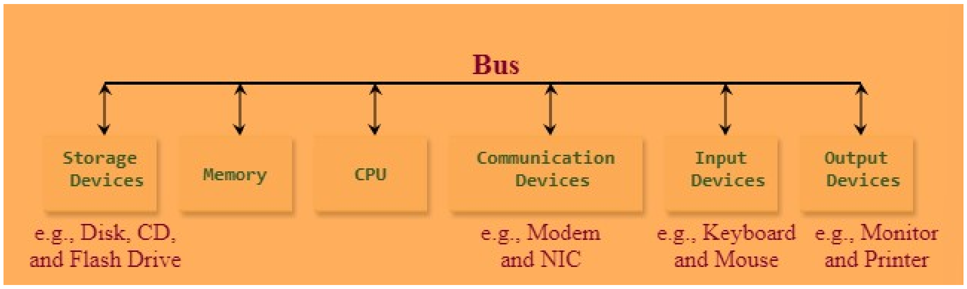
1.2

Wednesday, January 18, 2023 5:59 PM

- Computer? Software + Hardware
  - Hardware = physical, software = invisible that ctrl hardware
  - Major Hardware is
    - CPU (central processing unit)
    - Memory (main mem)
    - Storage devices (CD, etc.)
    - Input Devices (mouse and keyboard)
    - Output Devices (monitor, etc.)
    - Communication devices (modem, etc.)
  -
  - 

  - Parts connected by subsys called bus, it built into motherboard
- CPU
  - Brain, gets instructions from mem and executes, has 2 parts: control unit & arithmetic/logic unit
    - Ctrl unit- ctrls and coord actions of other parts
    - Arithmetic/logic unit- numeric ops and logical ops
  - CPU's now made on small silicon semiconductor chips, many electrical switches (transistors) for processing info
  - Internal clock, now in gigahertz (GHz)
  - Now, many cores, which is part of processor that does reading and doing of instruction, multicore CPU has 2/more independent cores
- Bits and Bytes
  - Data and pgrms stored in comp, switches, on=1, off=0, each 1/0 is bit
  - Min storage of computer is byte, has 8 bits
  - Series/patterns stored as series of bytes
  - Encoding scheme- set of rules, tell how comp translates chars and numbs into data the comp can work with, usually scheme is string of bits, like ASCII
  - Comp storage measured in bytes and multiples,
    - Kilobyte (KB) = 1000 bytes
    - Megabyte (MB) = 1 mil bytes
    - Gigabyte (GB) = 1 bil bytes
    - Terabyte (TB) = 1 tril bytes
- Memory
  - Sequence of bytes, store pgrm and data prgm work w/
  - Mem like comp work area for executing pgrm, pgrm and data must be moved to mem b4 executed by CPU
  - Each byte in mem has unique address, used to find byte for storing & retrieving data, in any

    - order, so random access memory (RAM)
    - More ram = faster operate, mem byte never empty, but initial content can be meaningless
    - Mem built on silicon semiconductor chips, many transistors on surface, less complicated than CPU chips, slower, less expensive
- Storage Devices
    - RAM is volatile for data storage, permanent on storage devices
    - Main types of storage devices are
        - Magnetic disk drives
        - Optical disc drives (CD/DVD)
        - USB flash drives
        - Cloud Storage
    - Drives- devices for operating think, storage thing physically stores data and pgrm instruction, drive reads data from thing & writes data on thing
    - Disks- comp has at least 1 hard disk drive, hard disk used for permanent storing data and pgrms, usually in comp, but can have removable
    - CD/DVD- compact disk, DVD's hold more
    - USB Flask Drives- universal serial bus, lets user attach lots of outside devices to pc, flash drive for storing/transporting data
    - Cloud Storage- internet
- Input/Output Devices
    - User comm w/ pc
    - Keyboard- function keys, modifier key are special keys (shift, alt, ctrl) take norm action of another when both pressed together
    - Mouse
    - Monitor- screen resolution specifies pixels in horizontal & vertical dimensions, higher res = clearer, dot pitch is space btwn pixels, smaller dot pitch = clearer
    - Touchscreen- input is finger/stylus, both input and output
- Communication devices
    - Dial up modem- phone line to dial phone number to internet
    - Digital subscriber line (DSL)- phone line, faster than dial up
    - Cable modem- cable tv line, faster than DSL
    - Network interface card (NIC)- connect pc to Local Area Network (LAN), which used to connect pc in small area
    - WiFi- wireless networking

1.3 Programming Languages

Friday, January 20, 2023 12:11 PM

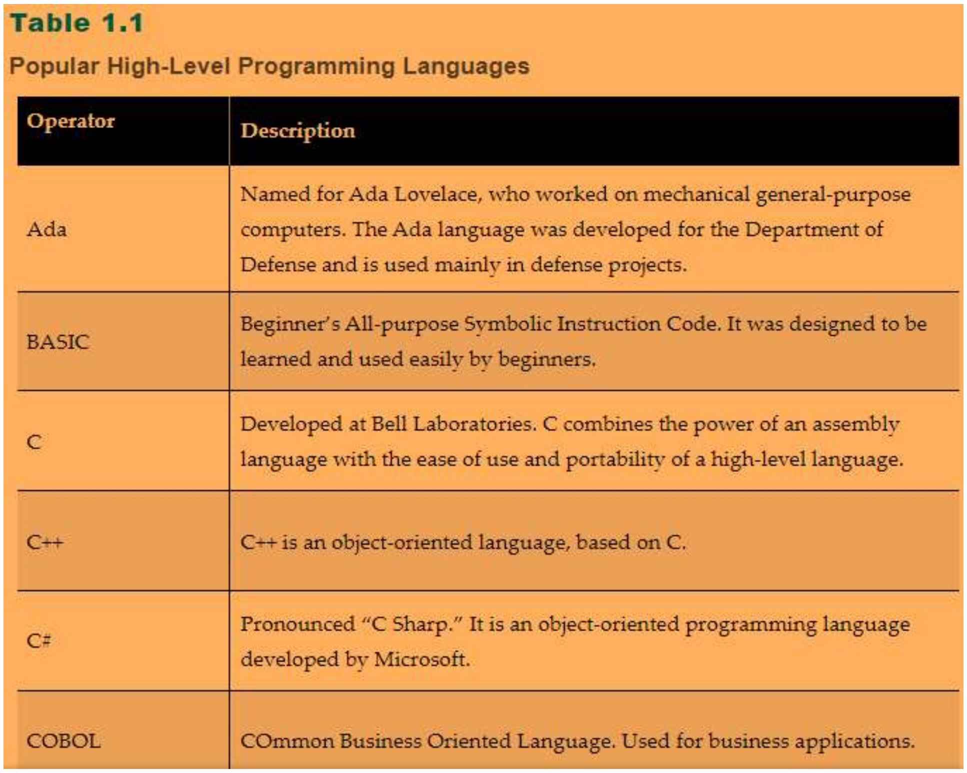
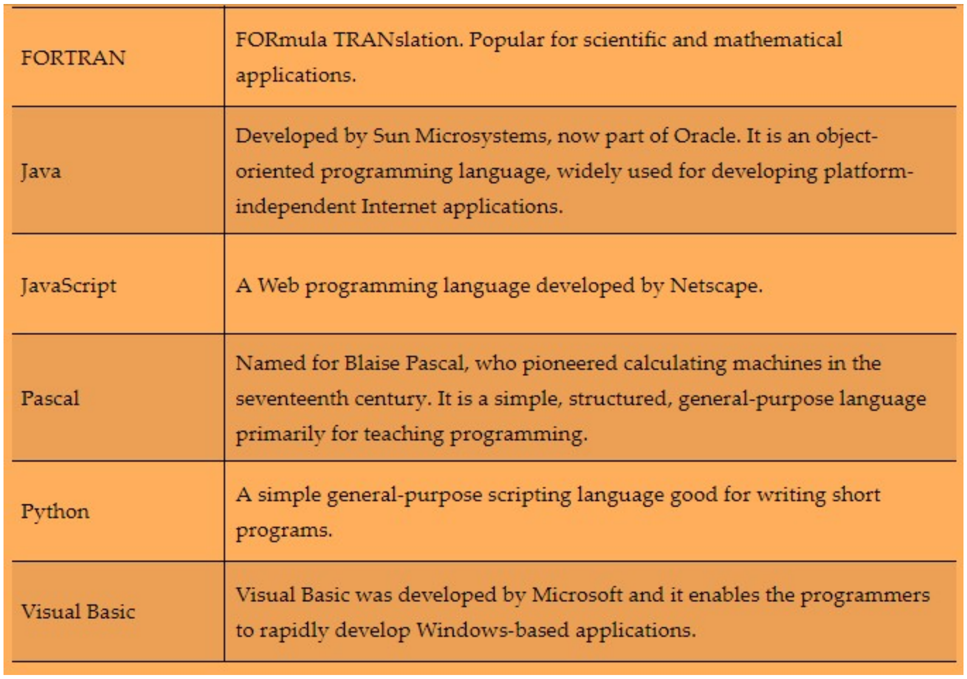
- Machine Language
  - Primitive instructions, binary
- Assembly
  - Alternative to machine, uses mnemonic to rep machine lan instructions
  - Assembler is program to translate assembly lang pgrm into machine code
  - Still tedious, low-level language
- High-Level Lang
  - Platform independent, so can run in diff types of machines
  - Instructions in high-lvl pgrm called statements
  -
  - 

    Table 1.1

    Popular High-Level Programming Languages

Operator	Description
Ada	Named for Ada Lovelace, who worked on mechanical general-purpose computers. The Ada language was developed for the Department of Defense and is used mainly in defense projects.
BASIC	Beginner's All-purpose Symbolic Instruction Code. It was designed to be learned and used easily by beginners.
C	Developed at Bell Laboratories. C combines the power of an assembly language with the ease of use and portability of a high-level language.
C++	C++ is an object-oriented language, based on C.
C#	Pronounced "C Sharp." It is an object-oriented programming language developed by Microsoft.
COBOL	COmmon Business Oriented Language. Used for business applications.
  -



FORTRAN	FORmula TRANslation. Popular for scientific and mathematical applications.
Java	Developed by Sun Microsystems, now part of Oracle. It is an object-oriented programming language, widely used for developing platform-independent Internet applications.
JavaScript	A Web programming language developed by Netscape.
Pascal	Named for Blaise Pascal, who pioneered calculating machines in the seventeenth century. It is a simple, structured, general-purpose language primarily for teaching programming.
Python	A simple general-purpose scripting language good for writing short programs.
Visual Basic	Visual Basic was developed by Microsoft and it enables the programmers to rapidly develop Windows-based applications.

- Pgrm written in high lvl lang called src pgrm/ src code
- Translated into machine code for execution thru compiler, which translates whole src code to machine-code file, then executed

1.4 Operating Systems

Friday, January 20, 2023 12:43 PM

Major job is controlling & monitoring sys activities, allocating & assigning sys resources, and Scheduling ops

- Controlling & Monitoring System Activities
  - OS does basic things and ensures diff pgrms don’t interfere w/ each other, also security
- Allocating & Assigning Sys Resources
  - OS resposible for what pc resources pgrm needs (like CPU time, mem space, disks, I/O devices) and giving/assigning them to run the pgrm
- Scheduling Operations
  - OS do scheduling pgrm's activities, efficiency of sys resources, techniques like Multiprogramming, Multithreading, and Multiprocessing
  - Multiprogramming- multiple pgrms run @ same time, share CPU, CPU is fast, so idle a lot, use to advantage
  - Multithreading- 1 pgrm do many tasks @ same time
  - Multiprocessing- like multithreading, but run multiple prgms @ same time w/ many processors

1.5 History of C++

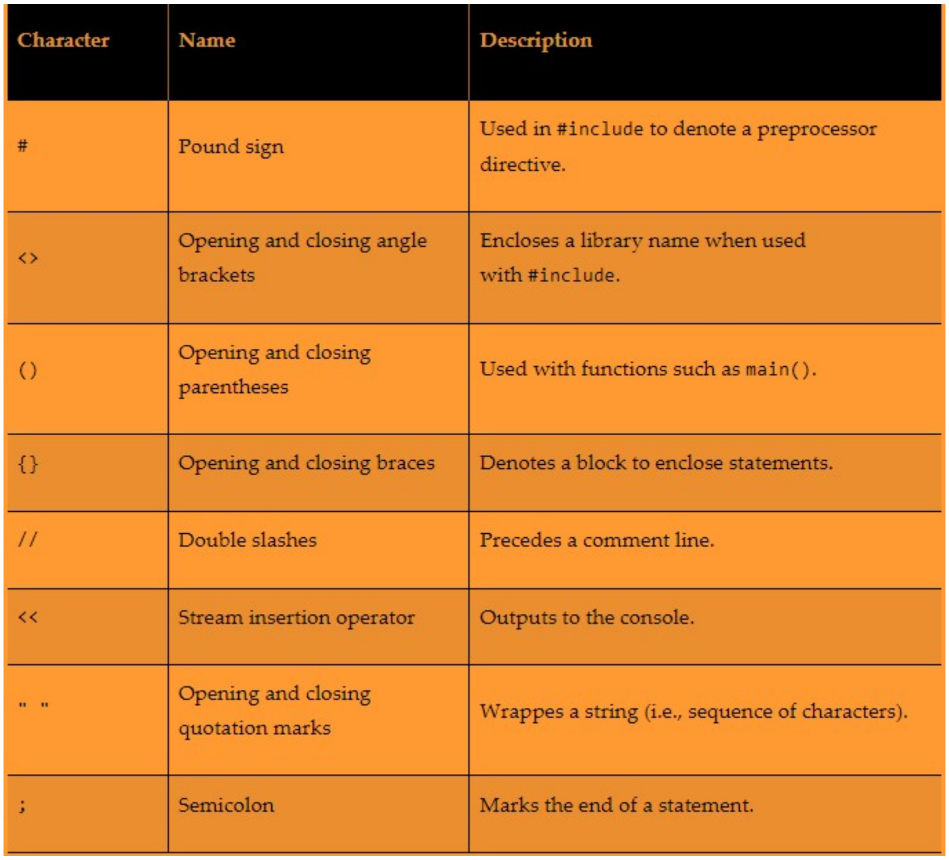
Friday, January 20, 2023 1:00 PM

- C++ is gen-purpose, oop lang
- C from B, which from BCPL (Basic Combined Programming Language), also used to make UNIX (@ Bell Labs, like the laser place)
- Java modeled after C++
- C# subset of C++
- C++ from C (Bell Labs), made it oop
- ISO standard makes C++ standardized

1.6 Simple C++ Program

Friday, January 20, 2023 1:08 PM

- Code is supposed to print "Welcome to C++!" on console
- Line 1 has #include, along with <library_name> so that the library can be used in the code, this actually called preprocessor directive
    - Ex. library is iostream, which lets input/output from keyboard, it looks like:
    - #include <iostream>
    - Library like iostream called header file
- Next line says using ___ ___ where first spaces are from the library and the second spaces are what it will be called in the code when referred to
    - Ex. Is namespace, it looks like
        - using namespace std;
- Then type main (type can my double, int, etc.), with curly braces
- // Denotes a comment line
- cout stands for console output
- << operator means stream insertion operator, which sends a string to the console, the string must be kept in " ", then the output should end with endl, stands for end line
- return 0; is always at end of main function to exit the prgm
- // is line comment,
- /* */ is a block/paragraph comment
- Don't put ; at the end of preprocessor directives, will cause error
- Don’t space out iostream and <>,
- C++ is case sensitive, it's main, not Main
-
-



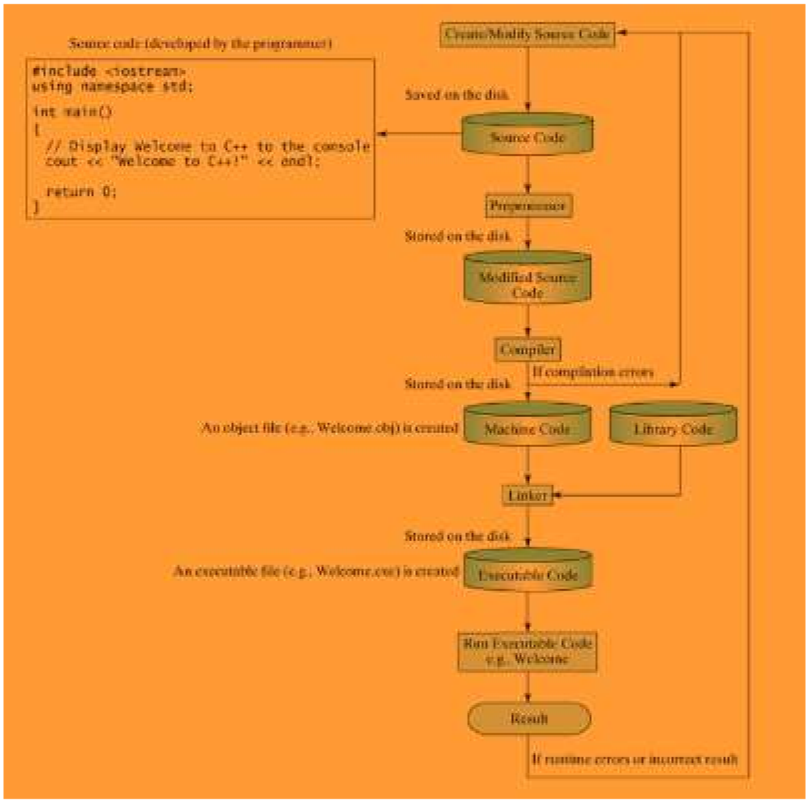
Character	Name	Description
#	Pound sign	Used in #include to denote a preprocessor directive.
<>	Opening and closing angle brackets	Encloses a library name when used with #include.
()	Opening and closing parentheses	Used with functions such as main().
{}	Opening and closing braces	Denotes a block to enclose statements.
//	Double slashes	Precedes a comment line.
<<	Stream insertion operator	Outputs to the console.
" "	Opening and closing quotation marks	Wrappes a string (i.e., sequence of characters).
;	Semicolon	Marks the end of a statement.

- Syntax is important, all statements end with ;
- Can do math, just type it in with cout << ____ << endl;

1.7 C++ Program Dev Cycle

Saturday, January 21, 2023 9:54 PM

- Create & compile pgrm b4 execute, fix compile errors and runtime errors
- C++ does in sequence: preprocessing, compiling, linking
- Uses 3 sep pgrms: preprocessor, compiler, and linker
    - Preprocessor- process src file b4 compiler, process directives (start w/ # sign)
    - Compiler- translate intermediate file to machine code file (objective file)
    - Linker- links machine code file w/ supporting library files, make executable file (on windows, machine code stored as .obj extension, executable stored as .exe, on unix, machine-code file stored as .o extesnion
    -
    -



```
#include <iostream>
using namespace std;

int main()
{
  // Display Welcome to C++ to the console
  cout << "Welcome to C++!" << endl;

  return 0;
}
```

Source code (developed by the programmer)
Create/Modify Source Code
Saved on the disk
Source Code
Preprocessor
Stored on the disk
Modified Source Code
Compiler
If compilation errors
Stored on the disk
An object file (e.g., Welcome.obj) is created
Machine Code
Library Code
Linker
Stored on the disk
An executable file (e.g., Welcome.exe) is created
Executable Code
Run Executable Code e.g., Welcome
Result
If runtime errors or incorrect result

    - C++ src files usually stored w/ .cpp extension
- Use ide (integrated development environment), visual studios

1.8 Programming Style & Documentation

Saturday, January 21, 2023 11:04 PM

- Documentation is important, explain code
- Use comments at beginning
- Use right indentation, space out operators
- Use brackets at end line style{
  Or at next line style
  {
  }
  }
- Make them vertically aligned as well

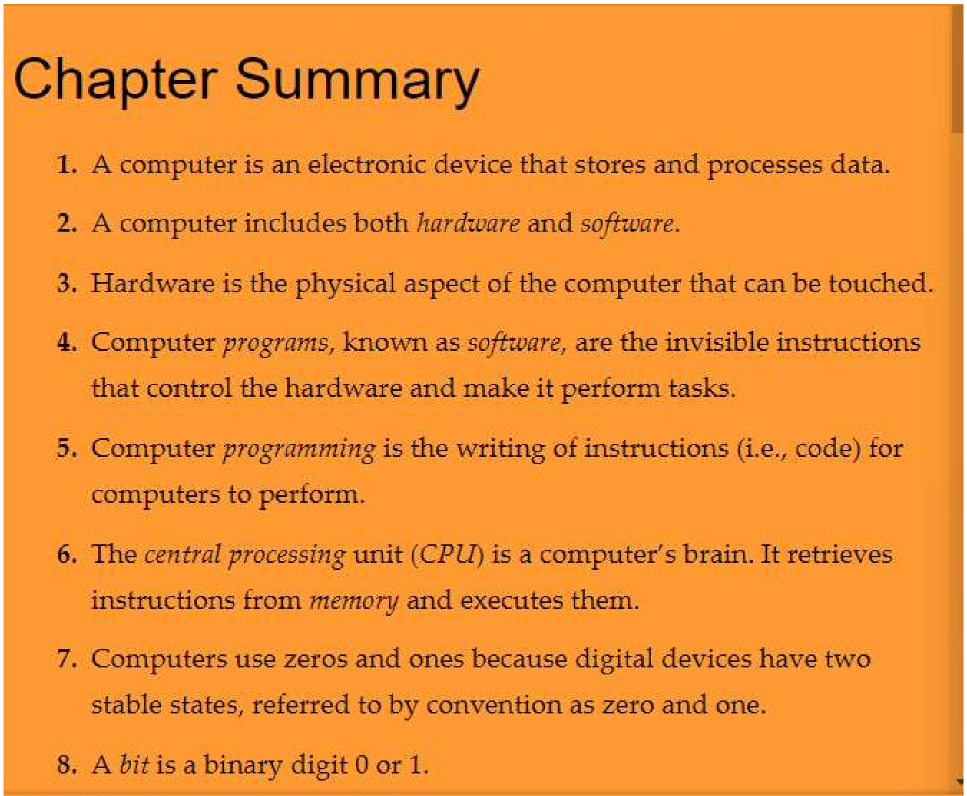
1.9 Programming Errors

Saturday, January 21, 2023 11:08 PM

- 3 types: Syntax, runtime, and logic
- Syntax Errors
    - Detected by compiler
    - Problems with code construction, usually easy to detect and fix, compiler usually says what and where
- Runtime errors
    - When application running, operation impossible to do
    - Input mistakes usually the cause, also /0
- Logic Errors
    - Pgrm doesn't do what intended
- Common Errors
    - Braces
    - Colons, semicolons
    - Quotations
    - Misspelled

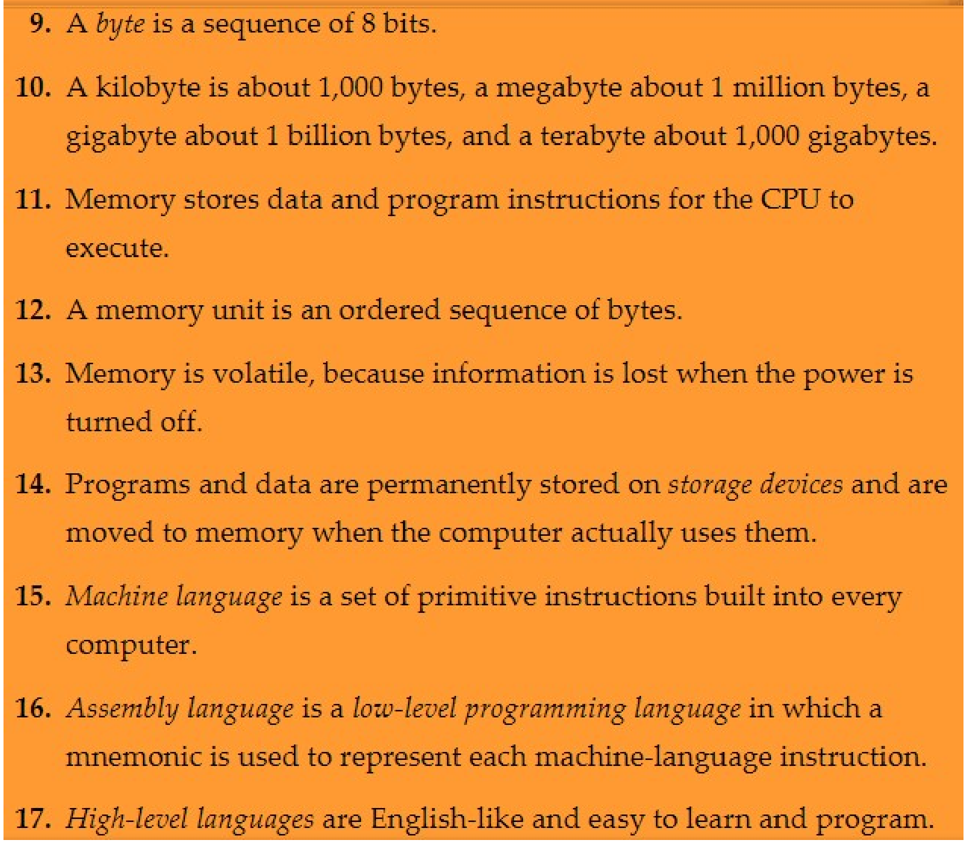
Chapter Summary + Terms

Saturday, January 21, 2023 11:13 PM

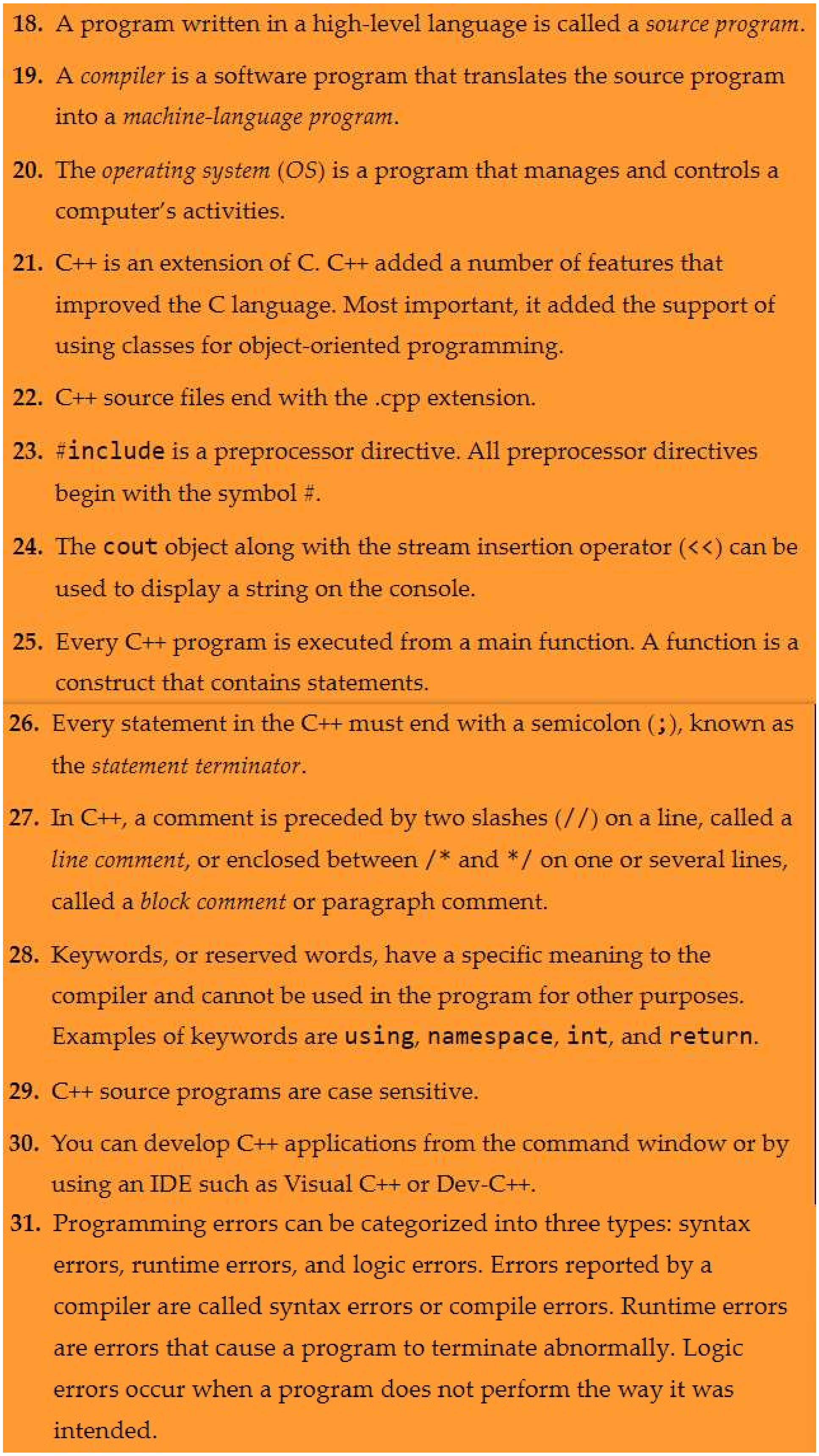


Chapter Summary

1. A computer is an electronic device that stores and processes data.
2. A computer includes both *hardware* and *software*.
3. Hardware is the physical aspect of the computer that can be touched.
4. Computer *programs*, known as *software*, are the invisible instructions that control the hardware and make it perform tasks.
5. Computer *programming* is the writing of instructions (i.e., code) for computers to perform.
6. The *central processing* unit (*CPU*) is a computer's brain. It retrieves instructions from *memory* and executes them.
7. Computers use zeros and ones because digital devices have two stable states, referred to by convention as zero and one.
8. A *bit* is a binary digit 0 or 1.



9. A *byte* is a sequence of 8 bits.
10. A kilobyte is about 1,000 bytes, a megabyte about 1 million bytes, a gigabyte about 1 billion bytes, and a terabyte about 1,000 gigabytes.
11. Memory stores data and program instructions for the CPU to execute.
12. A memory unit is an ordered sequence of bytes.
13. Memory is volatile, because information is lost when the power is turned off.
14. Programs and data are permanently stored on *storage devices* and are moved to memory when the computer actually uses them.
15. *Machine language* is a set of primitive instructions built into every computer.
16. *Assembly language* is a *low-level programming language* in which a mnemonic is used to represent each machine-language instruction.
17. *High-level languages* are English-like and easy to learn and program.



18. A program written in a high-level language is called a *source program*.

19. A *compiler* is a software program that translates the source program into a *machine-language program*.

20. The *operating system* (*OS*) is a program that manages and controls a computer's activities.

21. C++ is an extension of C. C++ added a number of features that improved the C language. Most important, it added the support of using classes for object-oriented programming.

22. C++ source files end with the .cpp extension.

23. `#include` is a preprocessor directive. All preprocessor directives begin with the symbol #.

24. The `cout` object along with the stream insertion operator (`<<`) can be used to display a string on the console.

25. Every C++ program is executed from a main function. A function is a construct that contains statements.

26. Every statement in the C++ must end with a semicolon (`;`), known as the *statement terminator*.

27. In C++, a comment is preceded by two slashes (`//`) on a line, called a *line comment*, or enclosed between `/*` and `*/` on one or several lines, called a *block comment* or paragraph comment.

28. Keywords, or reserved words, have a specific meaning to the compiler and cannot be used in the program for other purposes. Examples of keywords are `using`, `namespace`, `int`, and `return`.

29. C++ source programs are case sensitive.

30. You can develop C++ applications from the command window or by using an IDE such as Visual C++ or Dev-C++.

31. Programming errors can be categorized into three types: syntax errors, runtime errors, and logic errors. Errors reported by a compiler are called syntax errors or compile errors. Runtime errors are errors that cause a program to terminate abnormally. Logic errors occur when a program does not perform the way it was intended.