**Chapter 1 – COP 1000**

1. Introduction
   1. Computers can be used for anything
      1. Software/programs dictates what a computer does
         1. Made by programmers/software developers
   2. Python is an easy and common language
      1. Used by Google, YouTube, NASA, government, game companies
   3. Five main components of a computer
      1. CPU
      2. Main memory (RAM)
      3. Secondary storage (HDD/SSD)
      4. Input devices
      5. Output devices
2. Hardware and Software
   1. Hardware
      1. CPU (processor/microprocessor)
         1. Part of the computer that runs the program
         2. Most important part, could not run software without it
         3. ENIAC
            1. World’s first programmable computer
            2. Built in 1945 to calculate ballistic table for the US army
            3. One big CPU, 8ftx100ft, weighed 30 tons
         4. Small chips known as microprocessors
   2. Main Memory
      1. Computer’s work area
      2. Place where computer stores the program and data that is currently being used
         1. Ex: MS Word currently being stored on main memory while being used
      3. Commonly known as random-access memory, RAM
      4. CPU can quickly access data stored at any random location in RAM
      5. RAM is considered volatile memory and only used as temporary storage
         1. Once power is lost, all data on RAM is deleted
      6. RAM is stored in chips
   3. Secondary storage devices
      1. Type of memory that can hold data for long periods of time
         1. Even when there is no power to the computer
      2. Programs normally stored there until it is needed, then loaded onto RAM
      3. Important data is also stored on secondary storage
         1. Word docs, payroll data, inventory records, etc…
      4. Most common type are disk drives
         1. HDD – Hard disk drive
            1. Stores data on a magnetic spinning disk
            2. Slower but more reliable than SSD
         2. SSD – Sold-State drive
            1. Data is stored in solid-state memory
            2. No moving parts and operates faster than HDD
      5. External drives also are an option to store additional data or a back up of the main secondary storage
      6. USB drives
         1. Another type of secondary storage
         2. Uses flash memory
      7. Optical drives
         1. Another type of secondary storage
         2. Data not recorded on magnetic disks
            1. Recorded as a series of pits on the disc surface
            2. Laser is used to record and read data
         3. Hold a large amount of data
            1. Commonly used for creating backup copies of data
   4. Input devices
      1. Input is any data the computer collects from outside sources
      2. Keyboard, mouse, touchscreen, scanner, microphone, camera, etc…
      3. Disk drives and optical drives can also be considered input devices
         1. Since they data is retrieved from them and loaded into RAM
   5. Output devices
      1. Any data the computer produces for people or for other devices is considered output
      2. Any device the computer sends output to
         1. Monitors, printers
         2. CDs can also be considered output devices because the system sends data to them to be saved
   6. Software
      1. Dictates everything the computer does
      2. System software and application software
         1. System software
            1. Control and manage basic operations of a computer
            2. Operating system

Most fundamental set of programs on a computer

Controls the internal operations of the computer’s hardware

Manages all of the devices connected to the computer

Allows data to be saved o and retrieved from storage devices

Includes Windows, macOS, Linux, Android, iOS

* + - * 1. Utility programs

Performs specialized tasks that enhances the computer’s operations or safeguards data

Virus scanners, file compression programs (ZIP), data backup programs

* + - * 1. Software development tools

Programs that programmers use to create, modify, and test software

Assemblers, compilers, interpreters fall not this category

* + - 1. Application software
         1. Programs that make a computer useful for everyday tasks
         2. Programs that people normally spend most of their time running
         3. MS word, MS PowerPoint, MS Excel, email programs, web browsers, game programs

1. How Computers Store Data
   1. All data is stored as 1s and 0s
   2. Storage is divided into tiny storage locations known as bytes
      1. One byte can store a single letter or small number
      2. Each byte is divided into eight bits
         1. Bit = binary digit
         2. Bits can be thought of as switches
            1. Positive charge on a bit is on, negative is off
            2. Figure 1-6 pg. 8
   3. Storing numbers
      1. Bit can be used in very limited ways to represent numbers
      2. Bit on = 1, Bit off = 0
         1. Corresponds to binary perfectly
      3. Position of each bit has an assigned value
         1. Figure 1-8 pg. 9
      4. To determine value of binary number, add up the position of all values of 1s
         1. Figure 1-9, 10 pg. 9
      5. Smallest number storable is 0
         1. All bits 0
      6. Largest is 255
         1. All bits 1
      7. Bytes can be combined to get more bits to store bigger numbers
   4. Storing characters
      1. Letters and punctuation are first converted to numeric code then stored as a binary number
      2. ASCII
         1. American Standard Code for Information Exchange
         2. Set of 128 numeric codes that represent the English letters, punctuation, and other characters
         3. ASCII code for “A” is 65
            1. Figure 1-13 pg. 11
         4. Appendix C shows all ASCII codes and characters
         5. Developed in the early 1960s
         6. Limited as it only defines code for 128 characters
            1. Developed Unicode to remedy limitations of ASCII

Extensive encoding scheme that is compatible with ASCII

Can represent characters for many languages of the world

Quickly becoming the standard character set used in the computer industry

* 1. Advanced Number Storage
     1. Binary cannot be used to represent negative, and real numbers
        1. Only integers
     2. Negative numbers encoded using “two’s complement”
     3. Real numbers encoded in “floating-point notation”
     4. TC and FPN are used to convert negative and real numbers to binary
  2. Other types of data
     1. “digital” can be used to describe anything that uses binary numbers
     2. Digital data is data that is stored in binary format
     3. “Digital device” is anything device that works with binary data
     4. Pictures are composed of pixels
        1. Pixels are converted to a numeric code that represents the color
        2. Code is stored as binary
     5. Music is broken down into small pieces called samples
        1. Samples are converted to binary and stored
        2. The more samples the recording is divided into the more it sounds like the original music when played back
           1. CDs are more than 44,000 samples per second

1. How a Program Works
   1. A computer’s CPU can only understand instructions that are written in machine language. Because people find it very difficult to write entire programs in machine language, other programming languages have been invented.
   2. CPU is an electronic device that is designed to do specific things
      1. Reading a piece of data from RAM
      2. Adding two numbers
      3. Subtracting one number from another number
      4. Multiplying two numbers
      5. Dividing one number by another number
      6. Moving a piece of data from one memory location to another
      7. Determining whether one value is equal to another value
   3. CPU has to be told what to do
      1. Purpose of a program
         1. Program is a list of instructions that cause the CPU to perform operations
      2. Each instruction is a command that tells the CPU exactly what to do
         1. Ex: 10110000
         2. CPU understands that instruction and carries out the task
      3. CPUs only understand machine language
         1. Binary code of instructions in programs for the CPU
   4. Instruction set
      1. Entire set of instructions that a CPU can execute
      2. Not unusual for a program to contain thousands or even millions of machine language instructions
   5. Program installed on secondary storage but must be copied to RAM each time the CPU executes
      1. Double clicking a shortcut begins the process
         1. CPU moves program from storage to RAM and begins to run the program
         2. Figure 1-15 pg. 14
   6. Fetch-decode-execute Cycle
      1. Fetch
         1. Read next instruction from RAM to CPU
      2. Decode
         1. CPU decodes binary commands to determine which operation to perform
      3. Execute
         1. Perform the operation
      4. Figure 1-16 pg. 15
   7. Machine Language to Assembly language
      1. Computers can only execute programs that are in machine language
         1. Slow, tedious, and difficult to code in machine language
      2. Assembly language was created as an alternative to machine language
         1. Instead of binary, assembly language uses short words (mnemonics)
            1. add = addition
            2. mul = multiply
            3. mov = moving a value to a location in memory
         2. Many different versions of machine language
            1. Each brand of CPU has its own machine language instruction set
            2. Each brand of CPU also has its own assembly language
      3. Assembly language cannot be executed by the CPU
         1. An assembler is used to translate assembly language to machine language
         2. Figure 1-17 pg. 16
   8. High-Level Languages
      1. COBOL
         1. Created in the 1950s
         2. One of the early high-level languages
         3. DISPLAY “Hello world”
      2. Python
         1. Modern high-level programming language
            1. print(“Hello world”)
      3. Allows programmers to concentrate on the tasks they want to perform rather than the details of how the CPU will execute
      4. Thousands of high-level languages have been developed since the 1950s
         1. Table 1-1, pg. 17 shows some of the more well-known ones
   9. Key Words, Operators, and Syntax: An overview
      1. Each high-level language has its own set of predefined words that programmers must use to write a program (key words/reserved words)
         1. Each key word has a specific meaning and cannot be used for any other purpose
         2. Table 1-2, pg. 17 shows all Python key words
      2. Programming languages have operators as well as key words that perform various operations to data
         1. All languages have operators that perform arithmetic
            1. Ex: 12 + 75

“+” is the operator that tells the CPU to add

* + 1. Each language has its own syntax
       1. Set of rules that must be strictly followed when writing a program
       2. Dictate how key words, operators, and various punctuation characters must be used in a program
       3. Must learn new syntax rules for each particular language
    2. Statements are individual instructions used to write a program in high-level language
       1. Consists of key words, operators, punctuation, and other allowable programming elements
          1. Must be arranged in the proper sequence to perform an operation
  1. Compilers and Interpreters
     1. Depending on the high-level language used, programmers will use a compiler or interpreter to translate to machine language
     2. Compiler
        1. Translates high-level language program into a separate machine language program
           1. Machine language program can execute any time as necessary
           2. Figure 1-18, pg. 18
     3. Python uses an interpreter
        1. Both translates and executes the instructions in a high-level language program
        2. Reads each individual instruction in the program, converts to machine language and immediately executes
           1. Repeats for every instruction in the program
        3. Typically do not create separate machine language programs
     4. Statements that a programmer writes in a high-level language are called source code (code)
        1. Programmer types code into a text editor then saves
        2. Uses a compiler to translate the code or an interpreter to translate and execute
        3. Cannot be executed if the code has a syntax error
     5. Syntax error is a mistake in the code
        1. Misspelled key word, a missing punctuation character, or incorrect se of an operator
        2. Compiler or interpreter will display an error message if error is found
        3. Must look for and correct error and run through compiler or interpreter again

1. Using Python
   1. The Python interpreter can run Python programs that are saved in files or interactively execute Python statements that are typed at the keyboard. Python comes with a program named IDLE that simplifies the process of writing, executing, and testing programs.
   2. Installing Python
      1. Follow Appendix A for instructions
   3. The Python interpreter
      1. A program that can read Python statements and execute them (the interpreter)
      2. Interactive mode
         1. Waits for you to type Python statements on the keyboard
         2. Executes statement ten waits for next statement to be typed
      3. Script mode
         1. Interpreter reads contents of a file that contains python statements
            1. File known as a Python program or Python Script
         2. Interpreter executes each statement in the Python program as it reads it
2. Writing Python Programs and Running them in Script Mode
   1. Interactive mode is useful for testing code
      1. Statements entered are not saved as a program
      2. Simply executed and results are displayed on the screen
   2. Use Notepad to write a series statements and load it into the interpreter
3. The IDLE Programming Environment
   1. IDLE can be used as an alternative to interactive mode and script mode
   2. Automatically installed when Python is installed