Session 4: Software Part 1

Tech Skills 101: Driver's Ed for the Digital World

David Shamlin
OLLI Fall 2025

Apps (i.e., *software*)

- This week (Session 4)
  - What is software?
  - The role of the operating system (a.k.a. **OS**)
  - User Interface (a.k.a. **UI**): how we interact with apps
  - Settings (a.k.a. “preferences”)
- Next week (Session 5): the ***File System***
    - How to find a file you previously saved
    - “Cleaning” storage
    - Backing up storage

The size of data is measured in *bytes*

byte	• A byte is the smallest unit of data
	• One byte…
	  • can represent 256 values
	  • is made up of eight ***bits***		

kilobyte	KB	1,000 bytes	one thousand bytes
megabyte	MB	1,000,000 bytes	one million bytes
gigabyte	GB	1,000,000,000 bytes	one billion bytes
terabyte	TB	1,000,000,000,000 bytes	one trillion bytes

Examples

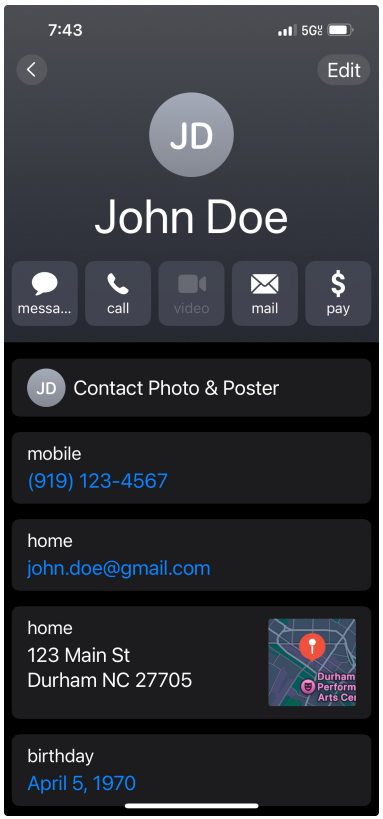
- *War and Peace*: 2MB
- Bible KJV: 2.2 MB
- Harry Potter series: 20MB
- 5 minutes of HD video: 1GB
- 5 minutes of 4K video: 3.4GB
- 5 minutes of 8K video: 6.7GB

Note: You may see the acronyms "Gbps" (*gigabits per second)* and "Mbps" (*megabits per second*) sometimes—often by Internet providers. E.g., "1000 Mbps". These numbers describe the rate data moves between devices/components; the higher the number, the faster data moves.

More on data: basic *data types*

Number of bytes	Number of values
1	256
2	65,536
4	4,294,967,296

- Numbers
  - Integer
  - Decimal
  - Floating point
- Text
  - Individual characters: "***char***"
  - Words / phrases / sentences: "***string***"
- Date & time
- Lists / sets / collections



More on “it’s all math to the processor”

- Images: geometry & algebra
- Audio: trigonometry
- Recommendations: linear/matrix algebra
- Predictive typing: probability
- AI: calculus & statistics
- Also (used broadly)
  - Set theory
  - Graph theory

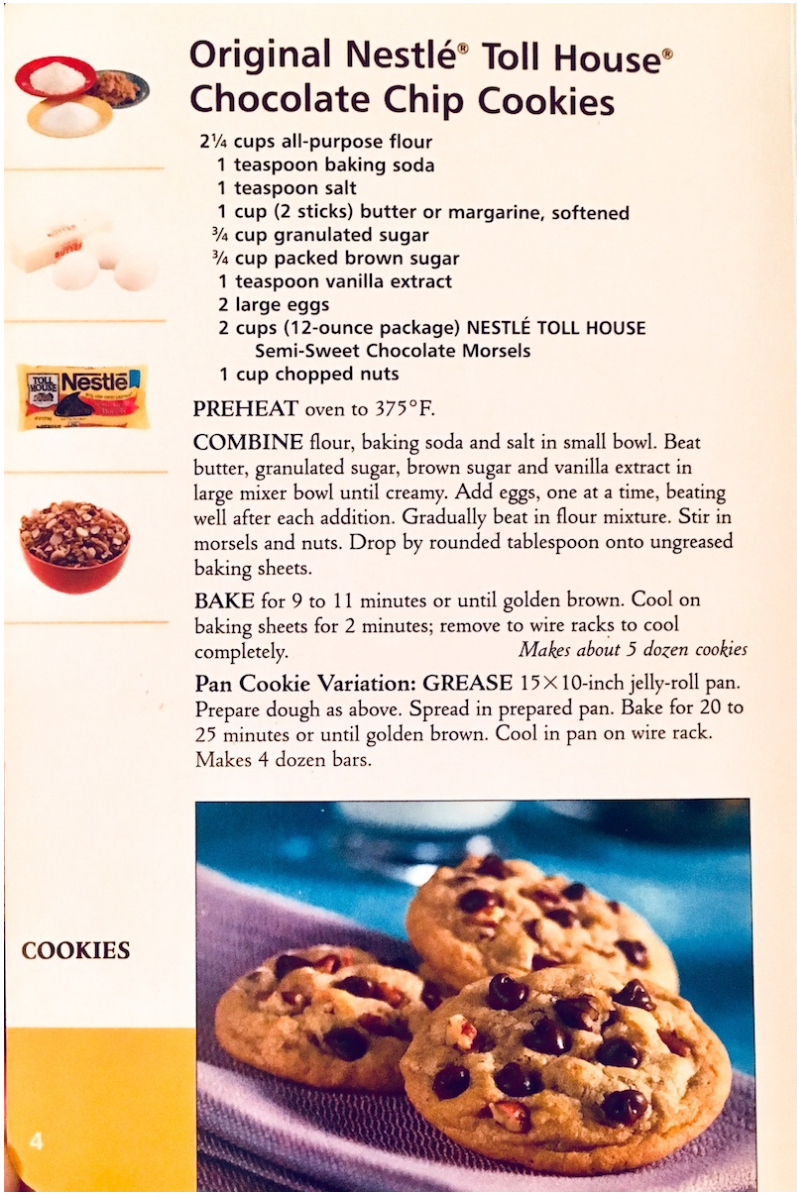
The power of mathematics is its ability to represent the physical world.

Processors can can perform arithmetic and logic.

All other kinds of math (i.e., algebra, geometry, calculus, etc) equations can be reduced/simplified to arithmetic and logic.

Software = Code + Data

Apps that run on…	…are called…
Desktop & Laptop	“Desktop”, “fat”, or “rich” apps
phones & tables	“Mobile apps”
In browsers (i.e., Safari, Chrome, Firefox, Edge, etc)	“Web apps”



Original Nestlé® Toll House® Chocolate Chip Cookies

2¼ cups all-purpose flour
1 teaspoon baking soda
1 teaspoon salt
1 cup (2 sticks) butter or margarine, softened
¾ cup granulated sugar
¾ cup packed brown sugar
1 teaspoon vanilla extract
2 large eggs
2 cups (12-ounce package) NESTLÉ TOLL HOUSE Semi-Sweet Chocolate Morsels
1 cup chopped nuts

PREHEAT oven to 375°F.

COMBINE flour, baking soda and salt in small bowl. Beat butter, granulated sugar, brown sugar and vanilla extract in large mixer bowl until creamy. Add eggs, one at a time, beating well after each addition. Gradually beat in flour mixture. Stir in morsels and nuts. Drop by rounded tablespoon onto ungreased baking sheets.

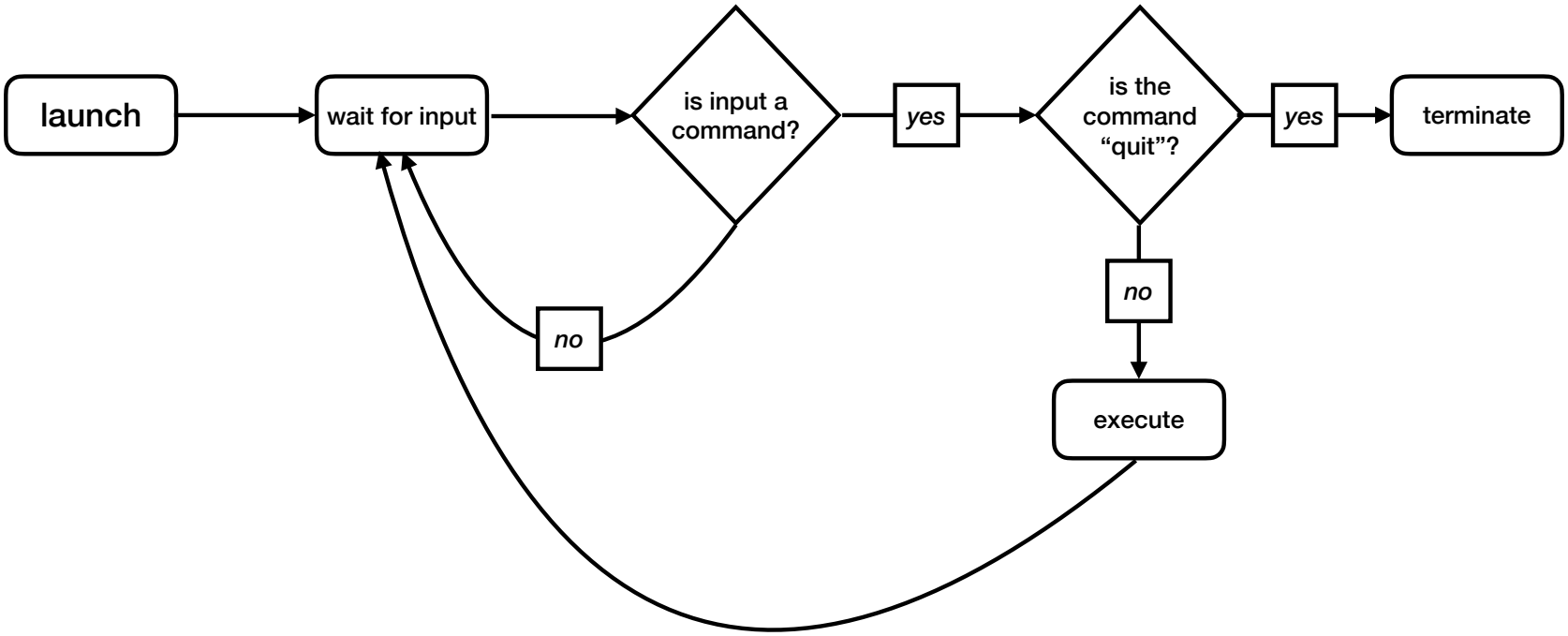
BAKE for 9 to 11 minutes or until golden brown. Cool on baking sheets for 2 minutes; remove to wire racks to cool completely. *Makes about 5 dozen cookies*

Pan Cookie Variation: GREASE 15×10-inch jelly-roll pan. Prepare dough as above. Spread in prepared pan. Bake for 20 to 25 minutes or until golden brown. Cool in pan on wire rack. Makes 4 dozen bars.

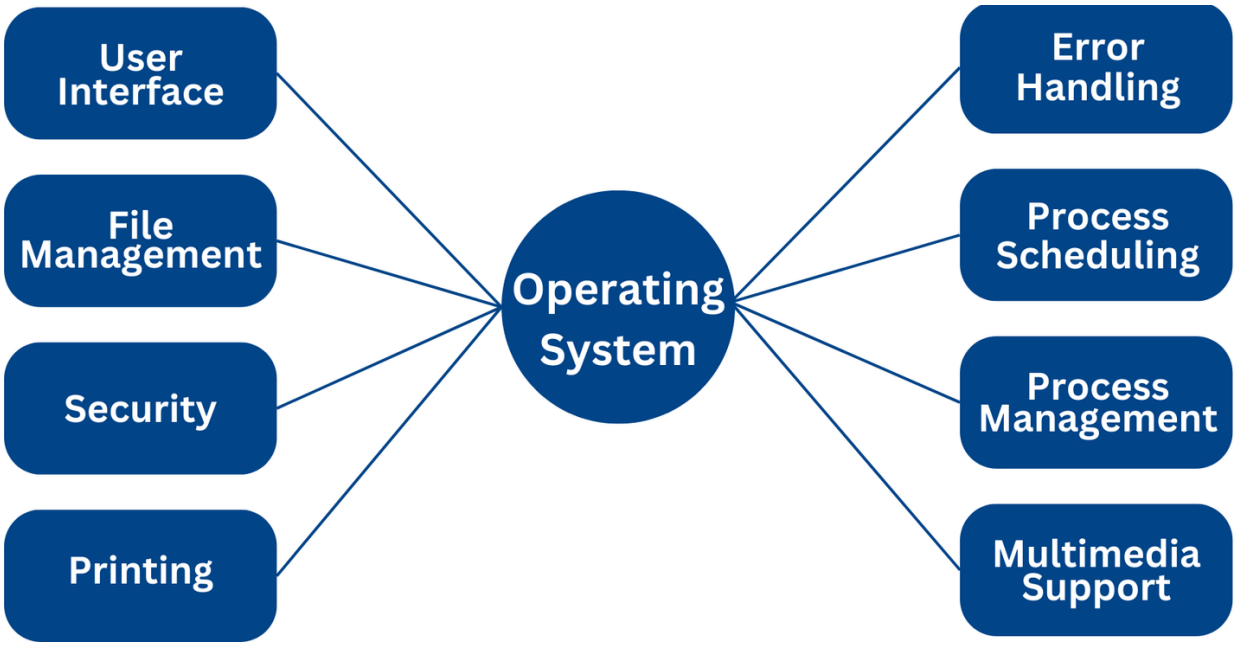
COOKIES

4

General “workflow” of an app



The role of the *operating system* (a.k.a. *OS*)



Kinds of Operating Systems

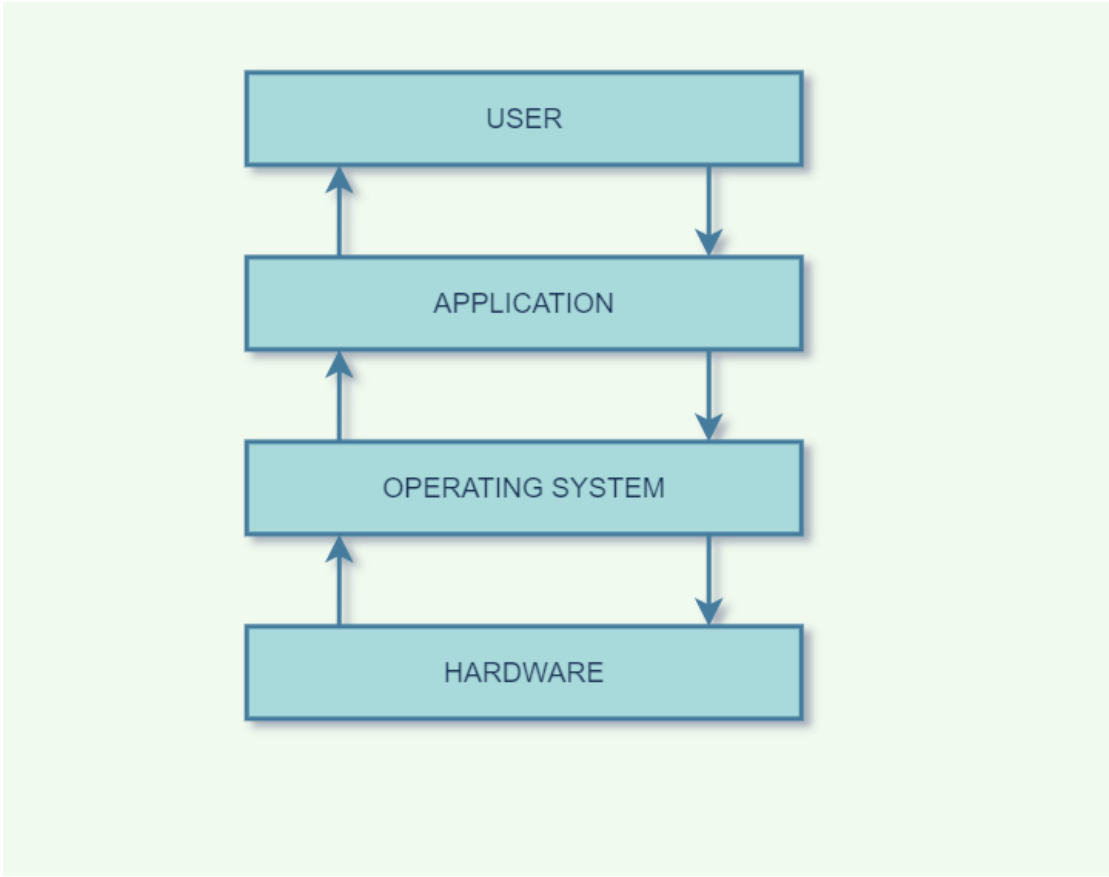
	Desktops & Laptops	Phones	Tablets
Apple	macOS	iOS	iOS
Microsoft	Windows	*none*	Windows
Google	ChromeOS	Android	Android

Note: You may encounter the names of other operating systems:

- **Unix** and **Linux**
  - Very commonly used by businesses
  - macOS and iOS are based on a "flavor" of Unix called BSD (a.k.a., "Berkley Software Distribution"); Apple renamed their "flavor" of BSD "Darwin".
  - ChromeOS is based on a "flavor" of Linux called ChromiumOS
  - Android uses some core parts of Linux

- **DOS**
  - Predecessor of Windows OS
  - DOS features are still included on Windows machines for compatibility reasons but DOS is not longer considered part of the Windows OS

The relationship between apps, OS, & Hardware



Examples of code

```
mylist = [7, 3, 9, 12, 11]

n = len(mylist)
for i in range(n-1):
  swapped = False
  for j in range(n-i-1):
    if mylist[j] > mylist[j+1]:
      mylist[j], mylist[j+1] = mylist[j+1], mylist[j]
      swapped = True
  if not swapped:
    break

print(mylist)
```

```
MYDATA SEGMENT
   ARRAY DB 5, 3, 8, 6, 2  ; Array of integers to be sorted
   LEN DW 5              ; Length of the array
   MSG1 DB 'Sorted array: $' ; Message to display before printing array
MYDATA ENDS

MYCODE SEGMENT
ASSUME CS:MYCODE, DS:MYDATA

START:
   ; Initialize data segment
   MOV AX, MYDATA
   MOV DS, AX

   ; Bubble Sort Algorithm
   MOV CX, LEN          ; Outer loop counter (number of passes)
OUTER_LOOP:
   DEC CX               ; Decrement counter
   MOV SI, 0            ; Initialize index for inner loop
   MOV DI, LEN          ; Load length of array for inner loop

INNER_LOOP:
   MOV AL, ARRAY[SI]      ; Load current element
   MOV BL, ARRAY[SI + 1]  ; Load next element
   CMP AL, BL             ; Compare elements
   JBE NO_SWAP            ; If AL <= BL, no swap needed

   ; Swap elements
   MOV ARRAY[SI], BL      ; Move larger element to current position
   MOV ARRAY[SI + 1], AL  ; Move smaller element to next position

NO_SWAP:
   INC SI               ; Move to the next index
   DEC DI               ; Decrement inner loop counter
   JNZ INNER_LOOP         ; Repeat inner loop if not done

   INC CX               ; Increment outer loop counter
   CMP CX, LEN          ; Check if we need to repeat outer loop
   JNZ OUTER_LOOP         ; Repeat if not done

   ; Print message before printing array
   MOV AH, 09h            ; Function to print string
   MOV DX, OFFSET MSG1    ; Address of message
   INT 21h              ; Print message

   ; Print sorted array
   MOV CX, LEN          ; Counter for printing array
   MOV SI, OFFSET ARRAY    ; Address of array
PRINT_LOOP:
   MOV DL, [SI]           ; Load element to print
   ADD DL, 30h            ; Convert to ASCII
   MOV AH, 02h            ; Function to print character
   INT 21h              ; Print character
   INC SI               ; Move to next element
   LOOP PRINT_LOOP        ; Repeat loop until all elements printed

   ; Newline
   MOV DL, 0Dh            ; Carriage return
   MOV AH, 02h            ; Function to print character
   INT 21h              ; Print character
   MOV DL, 0Ah            ; Line feed
   MOV AH, 02h            ; Function to print character
   INT 21h              ; Print character

   ; Program finished, halt
   MOV AX, 4C00h          ; Terminate program
   INT 21h              ; DOS interrupt

MYCODE ENDS
END START
```

Icons are frequently used to represent *actions*