

# Persistence and Memory

Week 8 – Presentation 1



# Persistence

New Topic :-)

Java persistence refers to programs which have elements that can  
“persist”

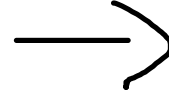
The term is also sometimes used to refer specifically to the practice of storing and retrieving java program elements (e.g. objects and methods) using the Jakarta Persistence API (was renamed from the Java Persistence API to the current name last year)



# Data Persistence



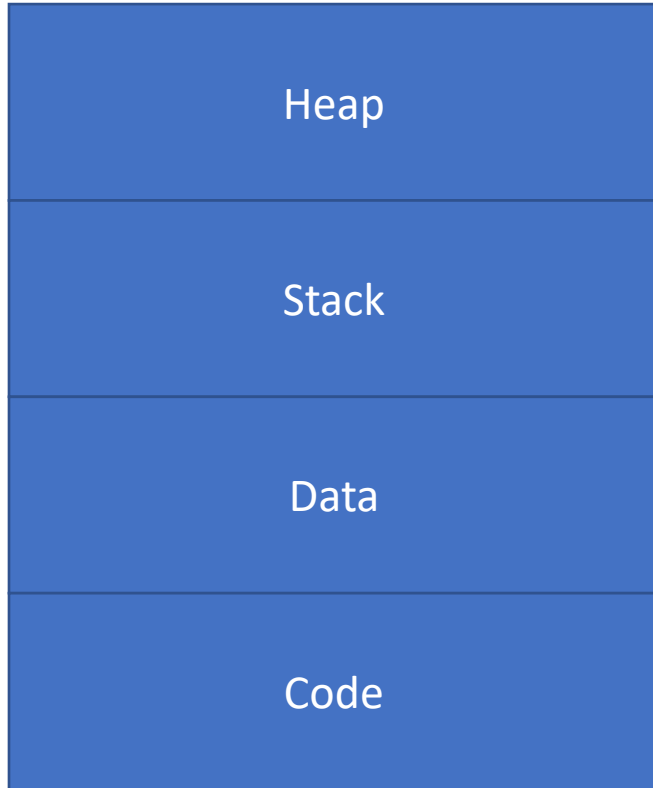
Per - Sistere



Per means thoroughly  
Sistere means to stand still

The word persistence comes from Latin



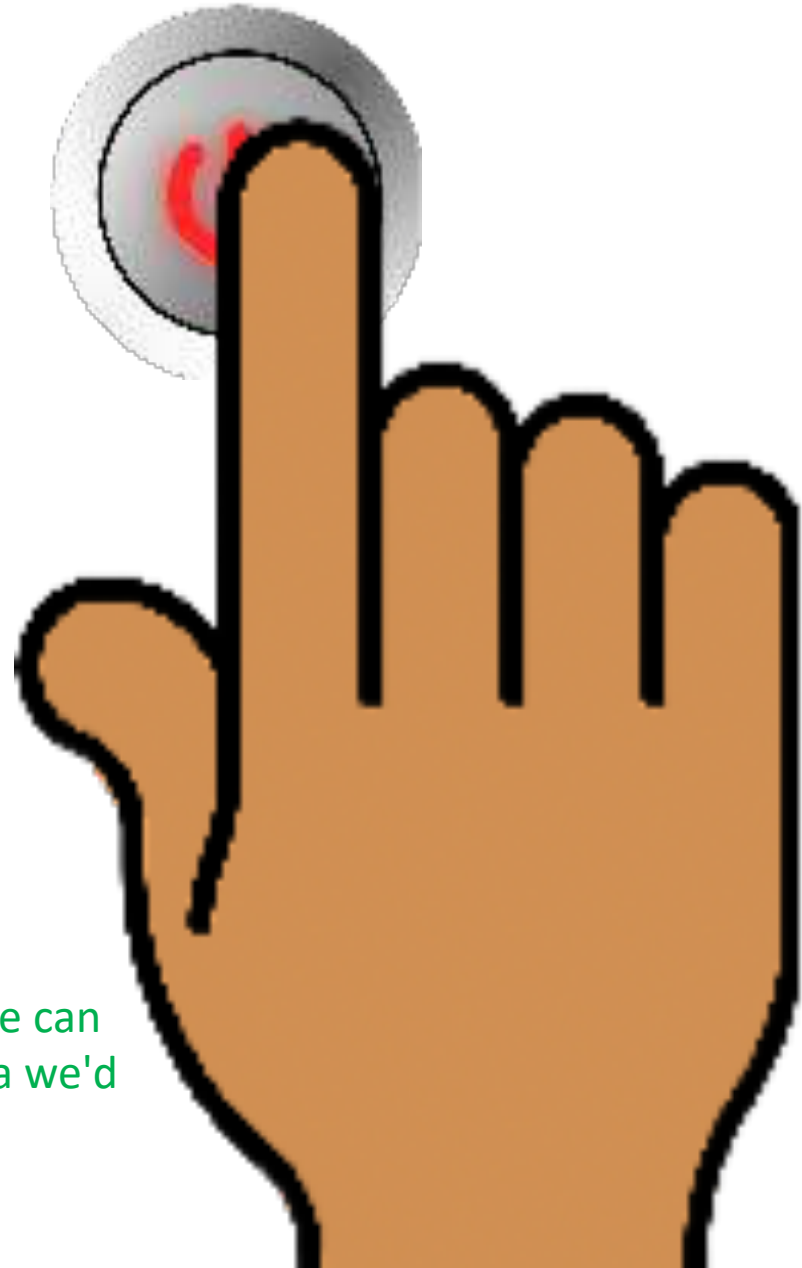
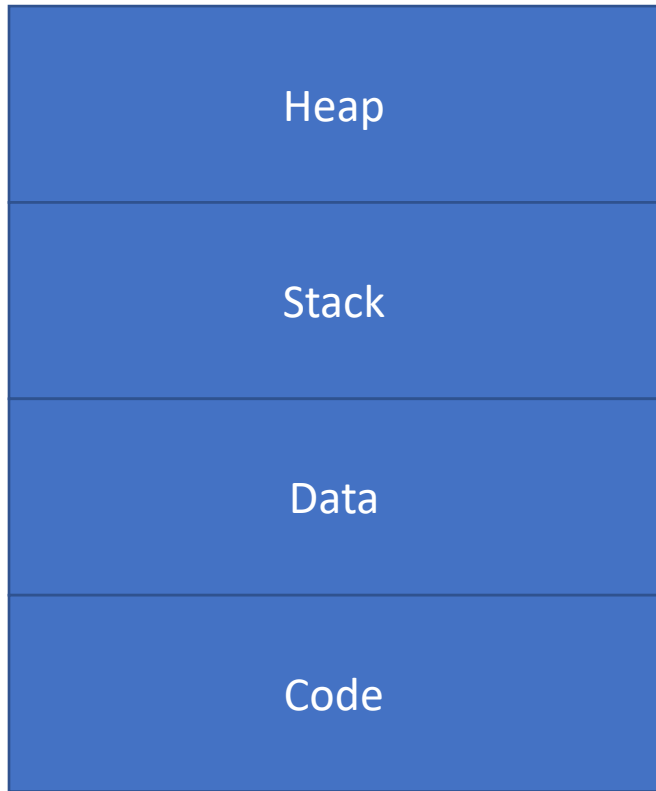


## The von Neumann Architecture



In the von Neuman architecture we have everything in memory.





But what happens when the power is cut? Poof, gone. We can reload programs when we restart, but if they modify data we'd rather save the work done somewhere.



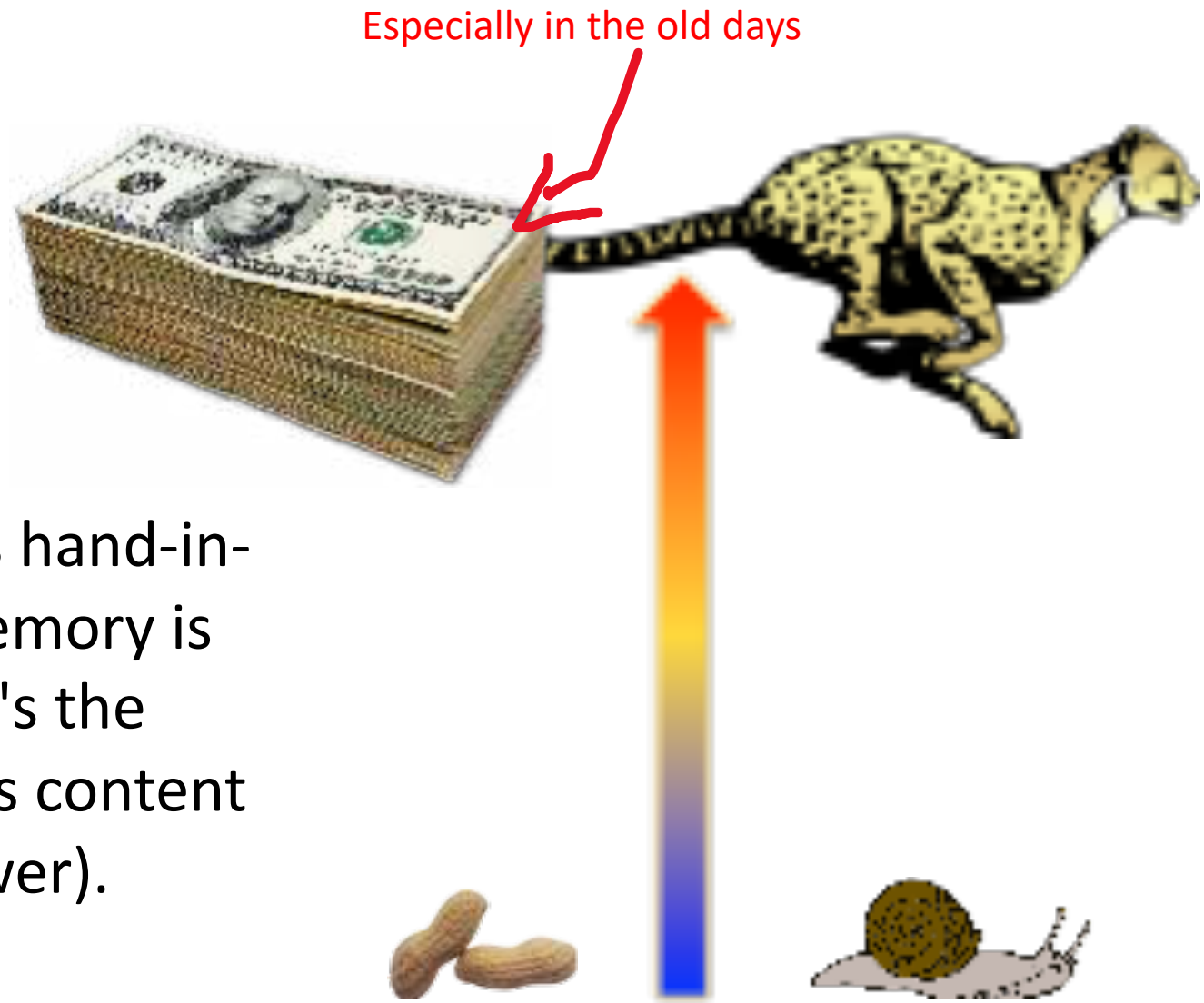
“Save”, “Keep”  
Need to **persist**  
program results  
somewhere

That's all the idea of persistence. "Somewhere" may take multiple shapes (including a remote computer)



# Memory

Memory is like cars: fast goes hand-in-hand with expensive. Fast memory is "volatile", which means (that's the meaning of the word) that its content can "fly away" (if you cut power). Slow memory stays, but the problem is that when you access it your program works at its slow speed.



So we have to work as much as we can in fast memory (RAM), and only in memory, for speed ...

# Memory

*Mostly work in memory  
for speed.*

... while we still need a safety net.

*But write to files for  
safety.*





# Non-Volatile Memory Technology

## Magnetic



There are several non-volatile technologies available. The oldest one, still alive and kicking, is the magnetic disk, which has done tremendous progress over the years in data density and transfer rates (not so much directly accessing data somewhere on the disk, called "random access")



# Non-Volatile Memory Technology

## Optical

Optical technologies are no longer so hot as they used to be, but they are very good for archiving and, because readers/writers are mass produced and cheap. Data updates? Better to look elsewhere.





# Non-Volatile Memory Technology

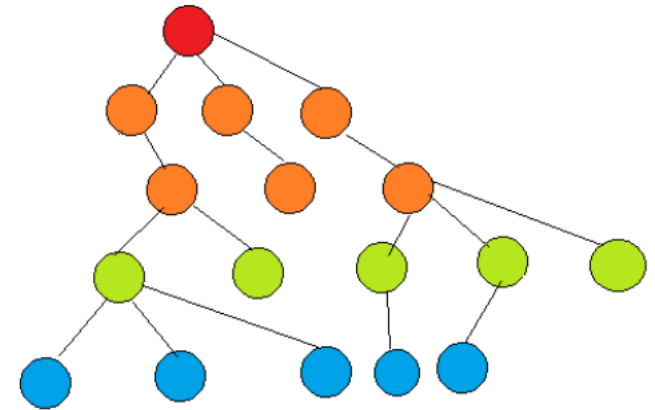
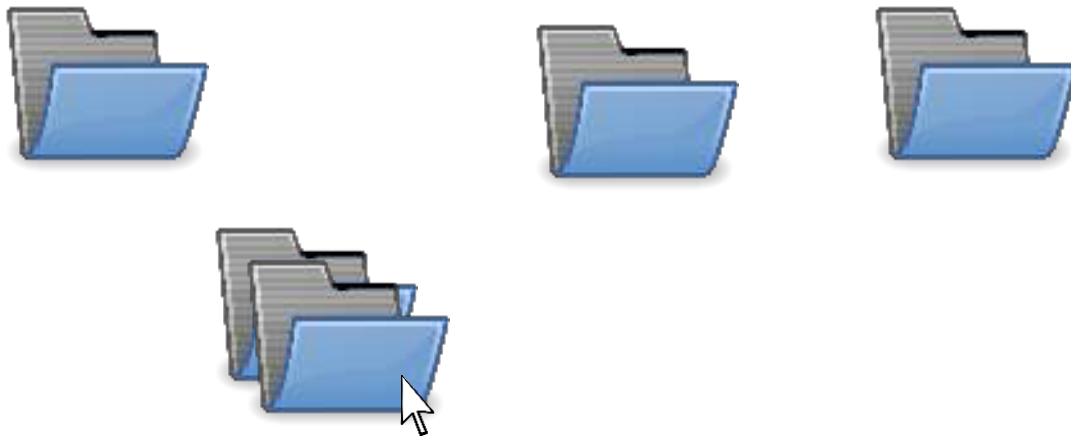
Solid-state technologies are very good for reading, much less so for writing.

## Solid State



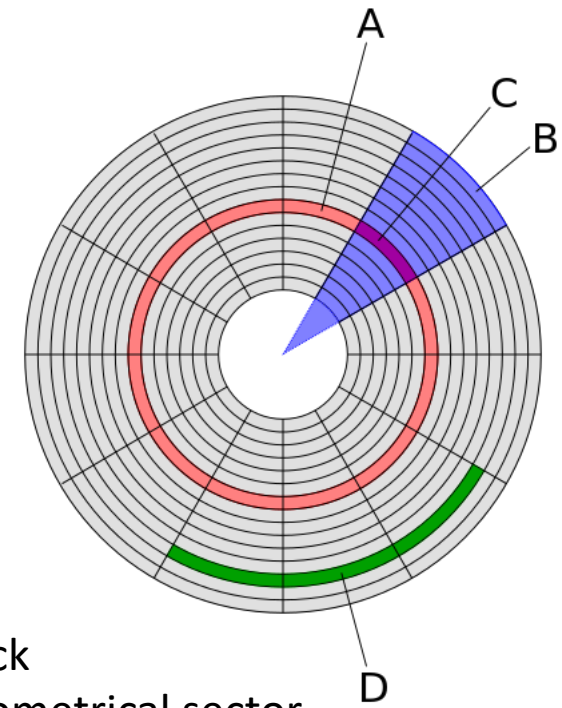
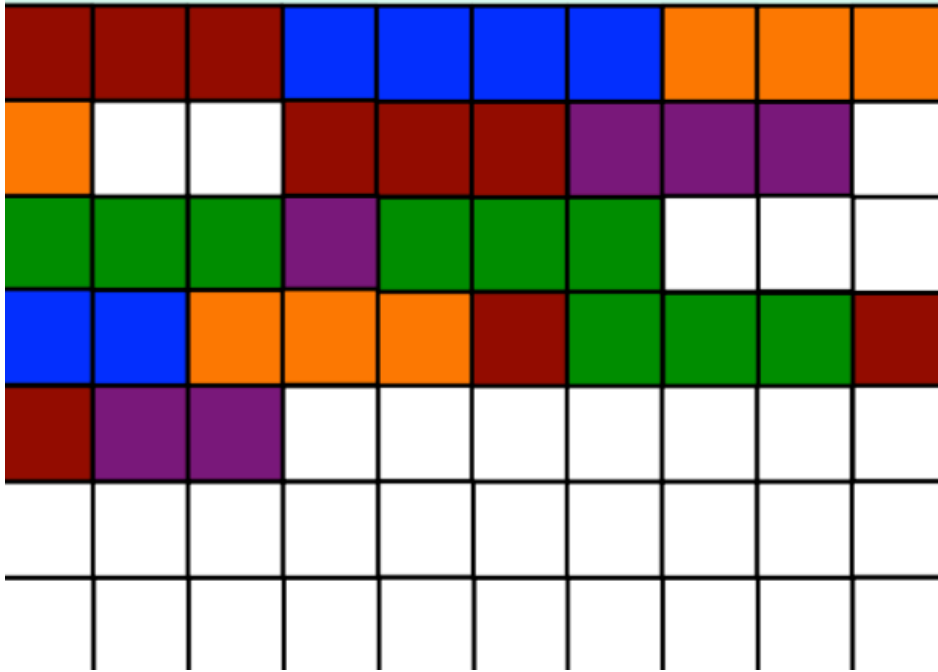
# File System

## Directories/folders



Your operating system will create on a disk a "file system" giving you a hierarchical view of directories and files.





- A – track
- B – Geometrical sector
- C – Track sector
- D – Cluster

But in reality the system will reserve blocks, all multiples of a same unit, that it will associate with one file. Files will grow, deletions will create gaps soon filled by blocks from other files, and the system will try to do all this as efficiently as possible. There is a strong disconnect between the view we have and physical reality.



Next: using file systems to store data

