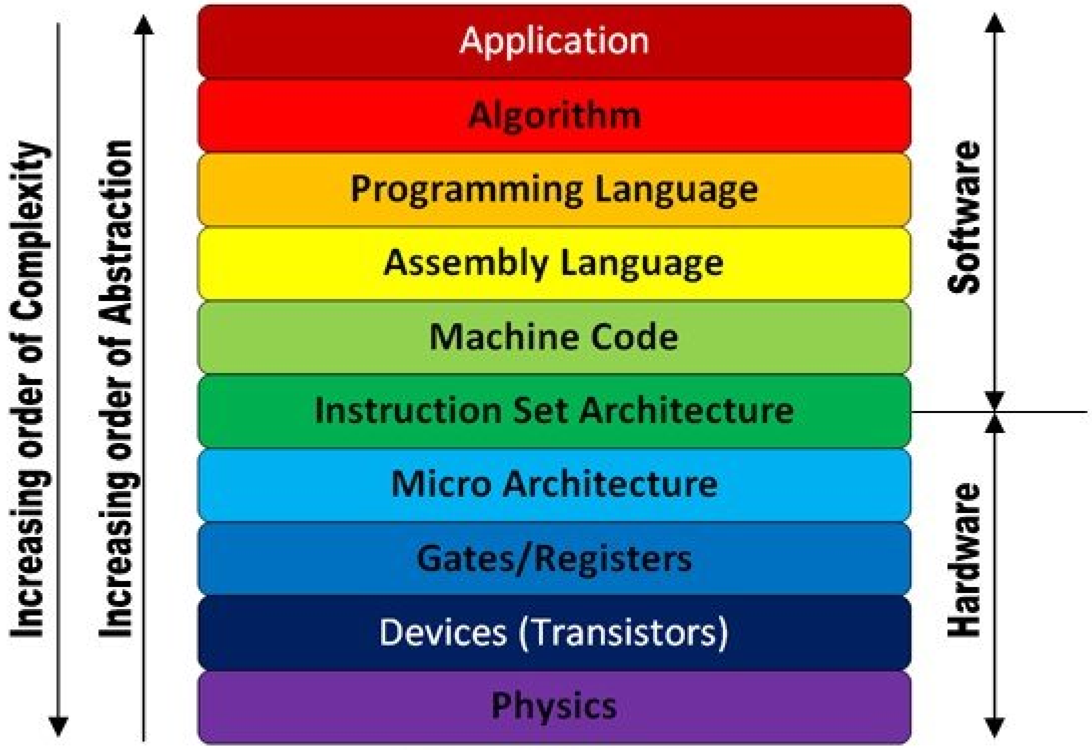
Tutorial 1
CSE 112 - Computer Organization

Q. Why do we need abstractions in systems design? Explain their advantages and disadvantages. Can you point out some systems which use this abstraction based design.

Advantage / Need: To enable parallel development across the technology stack.
Disadvantage: Special co-design strategies are needed for across-the-stack changes. Some changes might require changes to the abstraction layers, which might be difficult to do. In a monolithic architecture without layers of abstractions, changes might be easier to implement as there is no construct to maintain the abstraction interface.
Some examples: Computer Architecture, Compilers (Often we have multiple Intermediate Languages (IR)), Computer Networks (OSI model) etc.



Layers of Abstraction

Q. How does a C program run on your laptop? List as many steps as possible.

C program -> compiler -> Assembly language -> Assembler -> Binary (aka Executable) -> Loader -> Running Process

NOTE to TAs: Focus on the point of abstractions from High level C to low level assembly and Binary. The links provided are exhaustive and you can refer to other resources as well. Since this question is a bit long therefore explain each layer *briefly and concisely*. And it is not required to explain everything provided in these links. The motive of this question is to get students acquainted with a general flow of how the high level codes are executed on the machine.

1. https://medium.datadriveninvestor.com/compilation-process-db17c3b58e62
2. https://en.wikipedia.org/wiki/Loader_(computing)

Q. What are the main components in Modern desktop systems? Classify them as storage, computation, interconnect or power. List some of the key defining features of these components.

1. CPU, computation, core count, SMT support, clock frequency, IPC, etc.
2. GPU, computation, clock frequency, amount of GDDR memory, etc.
3. RAM, volatile storage, size, latency, etc.
4. HDD, non-volatile storage, size, latency, robustness (how long before it likely fails), etc.
5. SSD, non-volatile storage, size, latency, robustness (how long before it likely fails, often denoted in terms of read/write cycles), type (NVME/SATA) etc.
6. Motherboard, Interconnect, chipset, IO ports, form-factor, etc.
7. Power Supply, power, wattage, efficiency, etc.

NOTE to TAs: The motive of this question is to get students acquainted with the common terminologies used while working with the system. No need to focus on all abbreviations. Make sure basic terminologies are covered. But TAs should be familiar with all these concepts in case any students ask questions beyond the mentioned same concepts.

Q. Label all the components that you see in this picture.

1. CPU (Under the heatsink+pump of cooling system) (Top Left)
2. GPU (Under the CPU radiator)
3. RAM (Right of the CPU)
4. SSD (green color pcb under the chipset heatsink) (M.2 type (type of ssd does not matter much))
5. Power supply (Bottom Left)
6. Motherboard (The main PCB holding everything together)
7. Cooling system (Water cooling in this case) (Top Left)
8. HDD (might also be SATA SSDs, unclear from the picture) (Bottom right)
9. CD/DVD (Top right corner)