Where we’ve come from; a bit of history
Becoming digital from analog – sampling
Representing data – bits and bytes
How many bits/bytes to represent ....
Converting to binary/hex from decimal and back
Adding binary/hex numbers
Prefixes for large numbers – kilo, mega, giga, tera, peta and friends
EBCIDIC (punched cards) to ASCII (how are things represented) to UNICODE (16 bits wide)
Representing color (RGB) and sound; lossy vs. lossless
Major pieces of a computer and the bus that connects them
The architecture of the CPU – instruction sets, ALU, accumulator, caches
Memory – RAM vs. Disk vs. CPU cache
Fetch/execute cycle
Assembly language – the Toy programming language – simple instructions, testing and branching
Flowcharts and programs
Von Neumann architecture
History of computing devices
Computer circuitry – AND, OR, NOT, then NAND, NOR, XOR,
Truth tables and their realization as circuits
Building circuits from transistors; building transistors from silicon,
Moore’s Law
State machines – what they can and cannot do
Turing machines
Algorithms and their running times – logarithmic, linear, n log n, quadratic, ... exponential
Algorithms for finding the maximum, for sorting, for searching.
NP-complete problems
The difference between programs and algorithms
Constructs in a high level language
The rule of 72
QR problems of various types