Concepts and Material to Study Lectures 1-3:

- What are examples of computers?
    - Laptops, desktops - everyday computers also known as PCs (personal computers)
    - Servers - data processing computers and network computers (web servers, etc.)
    - Mobile devices - smartphones, tablets
    - Embedded systems - computers in cars, tv, appliances, etc.
- What are the components of an Information System (IS)?
    - Technology - Hardware, software, network, etc. The physical and informational (code, data) "things" that make up an Information System.
    - People - Developers, network administrators, analysts, etc. The individuals who utilize the Information System.
    - Method - Business orgs, process of conception to release, etc. The decisions on how to implement technology towards a goal or product.
- History of IS
    - Pre-1940s - mechanical devices for automated calculation
    - 1940-1960s - Mainframe era
    - 1970-1980s - PC era
    - 1990-2000s - Internet era
    - Late 2000-Present - Post-PC era
- Historical Figures
    - Charles Babbage - analytical engine to compute polynomials
    - Ada Lovelace - first computer programmer
    - Herman Hollerith - created punch cards, predecessor to language level code
    - Alan Turing - movie (The Imitation Game, tragic story of a very interesting and brilliant person) about him a few years back
        - Turing test - test to determine if a machine is "intelligent" utilizing a series of questions asked by a human. If human cannot tell if answers were given by a machine or a human then it would be considered "intelligent"
            - Note: I use quotes because intelligence is not a well defined concept
        - The bombe - machine used to decipher encrypted message by Germans in World War 2
        - Contributed probably too many ideas and theories in computer science to list
    - Grace Hopper - created the first compiler
- How computers "understand" the code we write?
    - Computers are a series of switches called transistors
    - These switches can output "high" and "low" voltages we call bits and which we represent as binary (base-2 number system)
    - We give computers a set of instructions (code) in binary which we call machine language

    - Since machine language is all 1's and 0's which is very hard to read, we have created an intermediate form of code called assembly, like 'mov ax, bx' (meaning moving the value stored in the memory location bx to memory location ax)
    - Assembly is translated to machine language by an assembler which takes the example 'mov ax, bx' and outputs something like 1001 1110 … (the binary I wrote is gibberish and not a real translation since I do not know machine language)
    - Since assembly itself is very tedious to work with to write large programs since not many people want to concern themselves with how to move information inside the memory (RAM) of a computer, a major step was taken when a compiler was first constructed (Grace Hopper)
    - Compilers take human readable coding languages (C/C++, etc.) and translate it into machine language
        - Note: if you want to see the steps you can write a basic C++ program like hello world and when you want to compile the C++ code write "g++ -S hello_world.cpp", this will output a hello_world.s file instead of an a.out file (executable file). This file is an assembly language file.
- Binary, Hexadecimal, Octal
    - Definitely review this section is Zybooks
    - Note: the right most digit of an n-base systems is n^0, the
    - Remember that a 3-bit binary conversion to decimal (base-10) is pretty easy:
        - 110 = 1*2^2 + 1*2^1 + 0*2^0 = 1*4 + 1*2 + 0*1 = 4 + 2 + 0 = 6
    - Hexadecimal is base-16 so instead of 0 through 9 in each digit as in decimal, or 0 or 1 in each digit like binary, each digit can be 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f
        - These represent the values of 0 through 15
    - Octal is base-8 so each digit can be 0 through 8
        - So in a 3 place octal number 162:
            - 1*8^2 + 6*8^1 + 2*8^0 = 1*64 + 6*8 + 2*1 = 64 + 42 + 2 = 108
- Number terminology
    - Kilo (k) = thousands, e.g. 5kB = 5,000 bytes
    - Mega (M) = millions, e.g. 5MB = 5,000,000 bytes
    - Giga (G) = billions, e.g. 5GB = 5,000,000,000 bytes
    - Tera (T) = trillions, e.g. 5TB = 5,000,000,000,000 bytes
    - Milli (m) = thousandths, e.g. 5ms = 0.005 seconds
    - Micro (μ) = millionths, e.g. 5μs = 0.000005 seconds
    - Nano (n) = billionths, e.g. 5ns = 0.000000005 seconds
- Memory size divisions
    - A bit is written as b, and byte is B
    - 8 bits = 1 byte
    - We often (particularly in cloud computing) will use a different division than base-10 like the number terminology above since memory and transistors are created by powers of 2
        - Kibibyte (KiB) = 1024^1 bytes (instead of 1000 bytes), Membibyte (MiB) = 1024^2, and so on

- Moore's Law
    - IC mean integrated circuit (basically what we call a computer chip)
    - And IC contains many transistors (gates, or what we can also call switches)
    - Moore's Law states that every 2 years we double the number of transistors on a single chip (exponential growth of transistors on a chip)
    - Moore's Law held steady for decades although now there are limitations as the size of a transistor is reaching physical limitations in how small they can be manufactured
    - However, there are ways to somewhat get around that problem
    - A corollary of Moore's Law has been information and data
        - The amount of data that can be processed has also increased in an exponential fashion
    - Clock frequency (measured in hertz, Hz) on the other has not increased exponentially
        - Higher clock frequency means more electricity being pulsed through a circuit which mean higher temperatures that may damage the chip itself
- Disk, memory, cache
    - Disk, also known as disk drives (HDD), were originally actual disks
        - The disks were spinning magnetic plates that held 1's and 0's (based on orientation like north and south of magnets)
            - A "head" was used to read the 1's and 0's
        - Disks are often now SSD (solid state drive) instead, these are faster, more durable, but also more expensive
    - Memory is also known as RAM
        - Can be standard RAM
        - Can be GPU RAM
        - Usually DRAM (dynamic random access memory)
        - RAM is fast
    - Cache is the "memory" directly attached to the CPU
        - It is very fast
        - It is very expensive
        - It is smaller (kB to MB in size)
        - This is the memory that the CPU directly uses to execute instructions
    - Speed: cache > memory > disk
    - Cost: cache > memory > disk
    - Size: cache < memory < disk
- Operating system
    - Please read **3.8**
    - I do not have much to add as the whole section I believe is fairly important to the understanding of the operating system from surface level view
    - Focus on what this section says, if you want to learn more on multitasking, multithreading, and multiprocessing I can point you to resources online for more in-depth investigations. Just focus on the section and not the details I went into

class as I will only ask you about multitasking and not multithreading or multiprocessing