Module 3 Hello World

Process of Executing a Program

- Focus on Three Main Components: Secondary Memory, RAM, CPU
  - Both Memories (RAM, Secondary) Contain alot of 0's & 1's
    - Bits of 0's & 1's encode a lot of info

- Bits are collected in 8 = Byte & Stored inside Memory
- Example: 8-Bits or 1 Byte is located in Physical Address 0, another Byte in Address 1 etc.
- All types of Data are saved like this

- In praticular a program or application
- Application: A set of instructions
  - Example: 95 Bytes is a sequence of instructions for a program named "prog.exe" (Typical file extension for an application = ".exe"
    - We can "execute" the application & make the comp perform instructions

- While executing program a few steps take place:
  1. Program is Stored in Secondary Memory
  2. Program is the "copied" to Main Memory (RAM) so the CPU Has faster access to the instructions
  3. These instructions are "executed" one after another by the CPU
     - CPU has "program counter register", tells where or what next set of instructions are to be executed. (Intialized to 100)
  4. Fetch - Decode - Execute Cycle Occurs
     - Each Cycle; the instructions that "program counter" points to is Fetched from Memory in CPU
     - CPU Decodes Instruction & Executes
     - Cycle Repeats now on 101...

Compilation:

- Instructions in Computers to run applications are in 0's & 1's
- Called "Machine Language" understood by CPU
- We will instead right our algorithms using a "high-level" language
- Specifically C++

- Programs written in a high-level language are written in a human language i.e. English
- Computers don't understand "English", so we need to translate to "Machine Language"

- Compilation: Process of translating "High-Level Language" to "Machine Language" or called Build Process
- Compilation is AUTOMATED