IPC144
Introduction to C Programming

Week-1

Computers
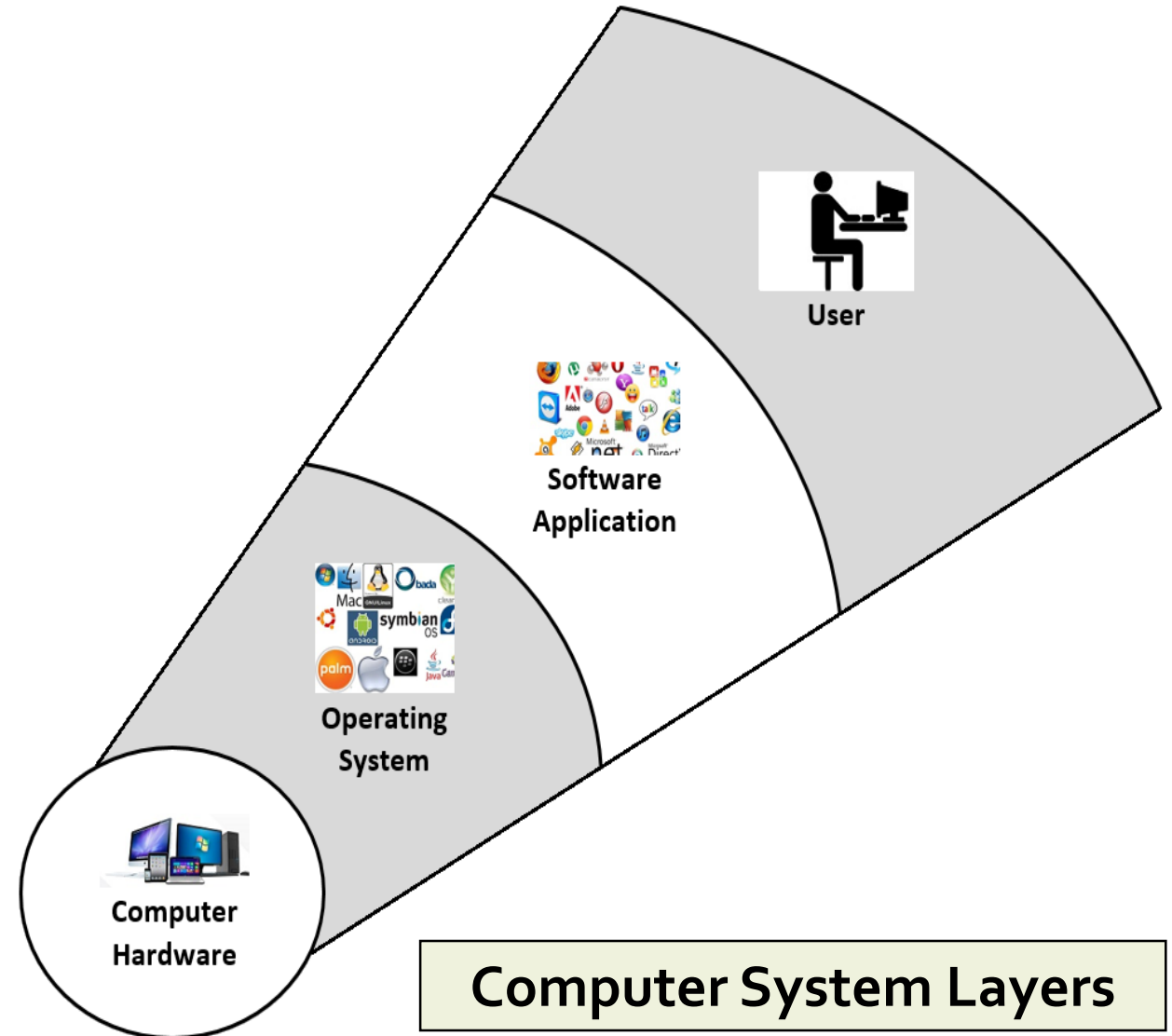**Information**
Compilers

Computers

Hardware

- Phone, Laptop/Notebook, Tablet, PC/Mac, Watches, Cars, Appliances, IoT, etc…

Software

- Clearly defined set of instructions

- "Coded" by a programmer in a computer language

User

Software Application

Operating System

Computer Hardware

Computer System Layers

Computers

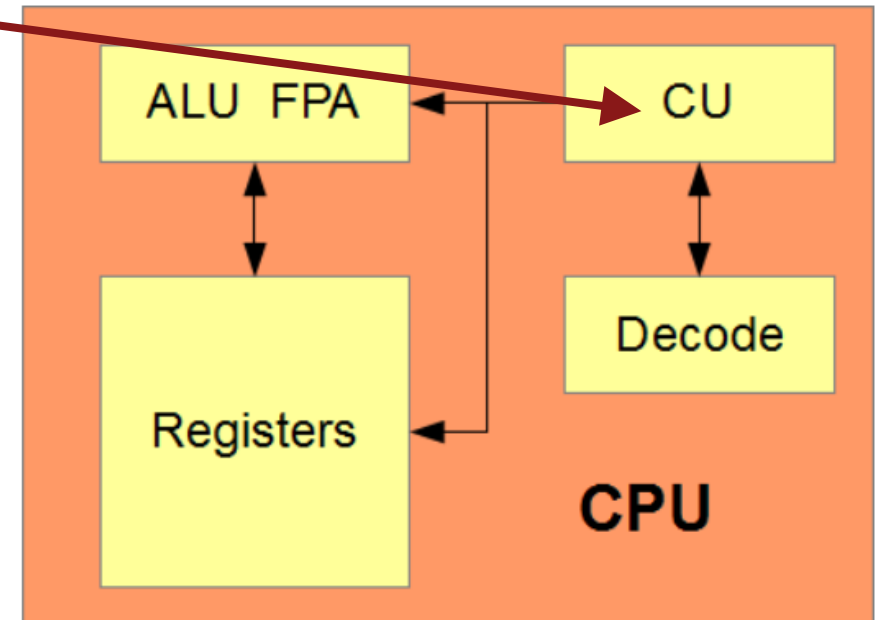## Central Processing Unit (CPU)

- Heart of the computer – executes all instructions one at a time as they are queued

Control Unit

- Manages the data

- Moves data to and from:
  - Registers
  - ALU & FPA
  - Decode Unit
  - RAM

http://scs.senecacollege.ca/~ipc144/pages/content/probl.html

Computers

Decode Unit

- Translates instructions into something the control unit (CU) understands

Arithmetic Logic Unit

- Calculations and comparisons for integral (whole) numbers as instructed by the CU
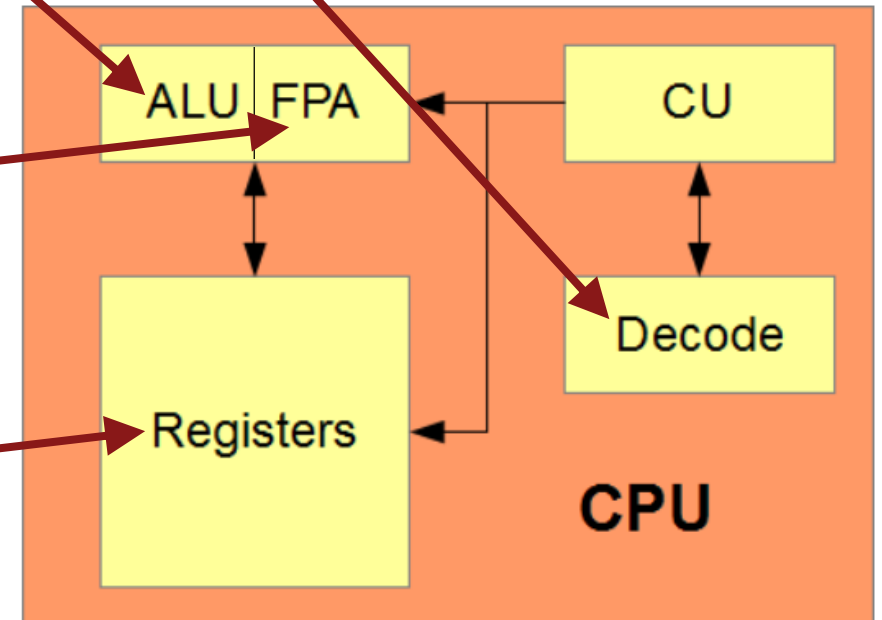
Floating Point Accelerator

- Calculations for floating-point (partial numbers) as instructed by the CU

Registers

- The CPU internal/onboard memory
 - The fastest of the data storage tiers

ALU FPA | CU
Decode
Registers
CPU

http://scs.senecacollege.ca/~ipc144/pages/content/probl.html

Computers

Primary Storage (Memory)

- This is memory that is directly accessed by the CPU

- 2nd fastest memory access tier

ROM

- Read only memory

- Manufactured with predefined instructions
  - Can be updated/changed such as firmware architecture (start-up instructions for a computer/device)
  - Some can't be updated or changed though (appliances)

- Data is "persistent" and is not lost when power is turned off
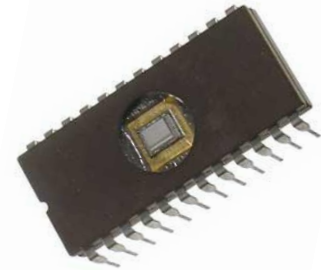
RAM

- Random Access Memory

- **Holds program (application) instructions and program data**

- Considered "volatile" (temporary) as this information is lost when the power is turned off

Computers

Secondary Storage (Memory)

- Slower than primary storage/memory but more affordable
  - Hard Disk Drives (HDD)
  - Solid State Drives (SSD)
  - CD/DVD/Blu-Ray Disks
  - USB "Keys"/Flash Drives

Other Devices (Peripherals)

Input:
  - Keyboard, mouse, touch screen, barcode scanner, voice/microphone, digital scanner

Output:
  - Monitor, Printer, Sound

Memory Speed Comparison

- Registers ~10 nanoseconds
- ROM and RAM ~60 nanoseconds
- Hard disk ~12,000,000 nanoseconds

A nanosecond is 10^{-9} seconds. To appreciate the differences, consider the following analogy:

- *The ratio of the time that the CPU takes to transfer data between <u>registers</u> to the time that a <u>hard disk</u> takes to transfer that same information is the ratio of the width of an **average-sized room** to the **distance once around the earth along the equator.***

https://scs.senecac.on.ca/~ipc144/pages/content/probl.html

Computers

Software: Controls a computer as long as the power is on

Operating System

- Primary software application responsible for managing the interaction between devices, other software applications and the user
- The O/S is loaded into RAM

 Application/Program

- Is given control from the operating system for execution
- The application is loaded into RAM
- Applications process data input, transforms the data in some manor and then generates some form of output (monitor, storage device, printer etc…)
- Application is developed ("Coded") using a programming language:
  - Processes input
  - Transforms input data and/or stored data
  - Generates output data
  - Also defines how data is stored in RAM
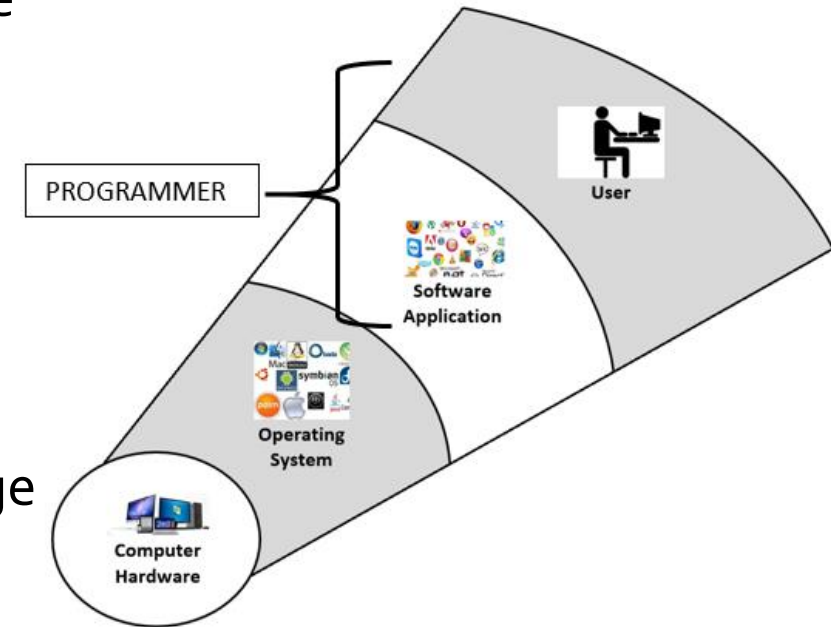
Information & Data Storage

Fundamental Units

Bit 💡

- Smallest unit of data represented at the binary level ("binary digit" or "bit")

- Is either a value of 0 or 1 (on/off)

Smallest Unit

Nibble 💡

- A larger unit (conceptually referred to) which represents 4 bits (half a byte)

4-Bits in a Nibble

Byte 💡

- The fundamental addressable unit of RAM (denominations of Byte sizes)

- Made up of 2-Nibbles or 8-Bits

8-Bits in a Byte
2-Nibbles in a Byte

Programming Languages

Generations

- There are hundreds of programming languages (over 2500) and can be classified into groups known as generation numbers:

-- Low Level --

1. <u>Machine language</u>: Native to CPU processes (specific by manufacturer)
2. <u>Assembly language</u>: More human readable than machine language but still considered a "low-level" language and is specific by manufacturer of the CPU

----- High-Level -----

3. **Third-Generation languages**: "Procedural" more human readable and are not tied to a particular machine manufacturer (portable)
   - C, C++, Java
4. **Fourth-Generation languages**: More human readable and describe WHAT is to be done without detailing HOW it is to be done (again, not tied to a particular machine and is portable)
   - SQL, Prolog, Matlab
5. <u>Fifth-Generation languages</u>: Among the newer and are used in artificial intelligence (AI), fuzzy sets (fuzzy logic) and neural networks (mocking the human brain)

- The higher the generation the more verbose (human readable) the language is. From the 3rd generation language onward there is no direct connection to manufacturer or machine language.
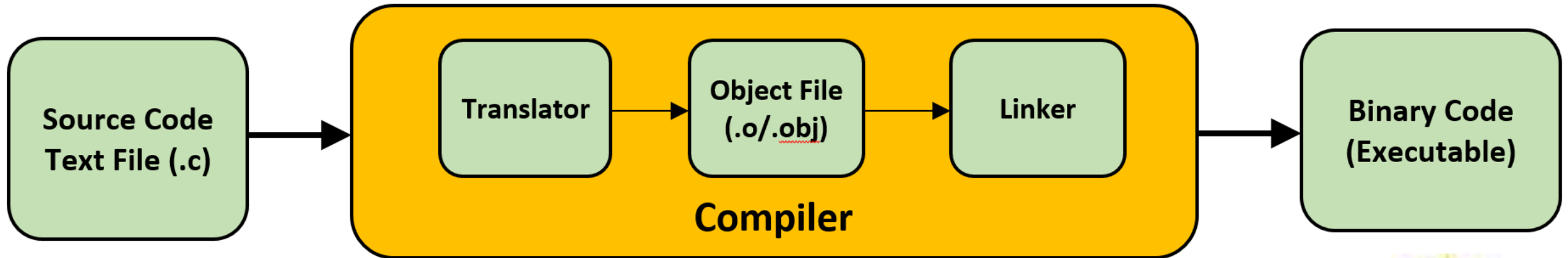
Features of C

- One of the more popular 3rd generation languages

- **Structured procedural language utilizing functions** 💡

- English-like (verbose/readable)

- Lowest in level of the high-level languages

- Very performant (**among the fastest and more powerful of the higher level languages**) 💡

- Widely used in high-performance computing (**Gaming: DirectX and OpenGL mixed C/C++**) 💡

- **UNIX, Linux and Windows** are written in C and C++ 💡

Compilers

Compiler

- Is operating system specific

- **Its purpose is to convert text (C programming language) into binary code (machine language)**


```
Source Code        ┌─────────── Compiler ───────────┐        Binary Code
Text File (.c)  →   Translator → Object File → Linker   →    (Executable)
                                  (.o/.obj)
```

General Steps of the Compiler

- *Reads a text file* with the extension .c containing the source code (C language)

- *Translates* and prepares each text file *into an intermediate machine language* file (object files)

- Each object file is *linked to create a single binary code file* (library or executable)

Writing Code

Text Editor

- Any text editor can be used (ex: Notepad)

IDE (Integrated Development Environment with compilers)

- Windows: Visual Studio 2019 – Community Edition
  - The ultimate in features (awesome debugging and management of large projects, inteli-sense, code navigation, multiple language support)
  - YOU MUST DO YOUR CODING IN VS2019

- Others/Cross Platform: Code blocks, Eclipse