

- ❖ **APPLICATIONS SOFTWARE** — a program or group of programs designed for end users
 - Divided into two classes: **system software** and **application software**.
 - System software= low-level programs that interact with computers at a basic level
 - Application software= Above system software. Includes applications such as database programs, word processors and spreadsheets. May be grouped along with system software or published alone.
 - Application software may simply be referred to as an application.
- ❖ **ARRAY** — The address of the 0th element of the array.
 - Memory** —Large array of location. Binary Info is stored/fetched, 1 Word at a time.
- ❖ **ASSEMBLER** —
 - Assembler**(Utility Program) translates the **Mnemonic**(Coding of Programmer) into **Machine Language**(Binary), then stores it in a File on Disk. TO EXECUTE, a **Linking Loader**(Another Utility Program) Loads/Links all necessary Machine Language Modules into Computer's Main Memory, and the PC is loaded with the Address of the 1st Instruction of the Main Program. The Last Instruction executed in the program Returns Control back to the OS(by using a System Call that generates a Software Exception).

- ❖ **BINARY ADDITION/ SUBTRACTION** — Two's Compliment for Subtraction. Inverse, +1, Add'em.
- ❖ **BINARY TO DECIMAL CONVERSIONS** —
- ❖ **DECIMAL TO BINARY CONVERSIONS** —

16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
32,768	16,384	8,192	4,096	2,048	1,024	512	256	128	64	32	16	8	4	2	1
$16^4 = 65,536$				$16^3 = 4,096$			$16^2 = 256$			$16^1 = 16$			$16^0 = 1$		

- ❖ **COLLATING SEQUENCE** — (also called a sort sequence) -defines how characters in a character set relate to each other when compared and ordered. Such as ASCII / EBCDIC.
- ❖ **COMPILER** — programs in a high-level language are translated into machine language by a compiler or interpreter.
 - The compiler derives its name from the way it works, looking at the entire piece of source code and collecting and reorganizing the instructions.
- ❖ **CONDITIONAL STATEMENTS** — most important components of Programming -enable a program to act differently each time it is executed, depending on the input. Most programming languages use the word *if* for conditional expressions.
 - (Can also be combined)Example, conditional statements:
 - If-Then-Else & Switch (Case Statements) & While-Do & For-Do
- ❖ **CONVERT "C" CODE TO MIPS** —
 - Use "Pseudocode" to specify and name the processor Registers. Assembly Pseudocode appears to be "c" in terms of Control Structures and Arithmetic expressions.
- ❖ **EDITOR** —
 - Nowadays, the term editor usually refers to **source code editors** that include many special features for writing and editing source code.
 - Sometimes called **text editor**, a program that enables you to create and edit text files. There are many different types of editors, but they all fall into two general categories:
 - line editors**: A primitive form of editor that requires you to specify a specific line of text before you can make changes to it.
 - screen -oriented editors**: Also called full-screen editors,these editors enable you to modify any text that appears on the display screen by moving the cursor to the desired location.
 - The distinction between editors and word processors is not clear-cut, but in general, word processors provide many more formatting features.
- ❖ **FUNCTIONS** —
 - a function is a type of procedure or routine. a distinction between a *function*, which returns a value, and a *procedure*, which performs some operation but does not return a value.

❖ **HEXADECIMAL –**

--Base-16 Number System. 0-9 A-F.

--commonly used by programmers to describe locations in memory because it can represent every byte (i.e., eight bits) as two consecutive hexadecimal digits instead of the eight digits that would be required by binary (i.e., base 2) numbers and the three digits that would be required with decimal numbers.

--Is much easier for humans to read hexadecimal numbers than binary numbers

❖ **HIGH-LEVEL LANGUAGE –**

--closer to human languages and further from machine languages.

--The main advantage of high-level languages over low-level languages is that they are easier to read, write, and maintain.

-- programs written in a high-level language must be translated into machine language by a compiler or interpreter.

-- The first high-level programming languages were designed in the 1950s. Now there are dozens of different languages, including Ada, Algol, BASIC, COBOL, C, C++, FORTRAN, LISP, Pascal, and Prolog.

❖ **INTERPRETER –**

--*Interpreters* analyzes and executes each line of source code in succession, without looking at the entire program.

❖ **JAVA –**

--Used for Pseudocode for Assembly.

-- Java is an object oriented languages (OOPS). All the programs based on class as well as objects. Java is a machine independent language.

--Source code files (files with *.java* extension) are compiled into a format called *bytecode* (files with *.class* extension), then executed by Java interpreter.

❖ **JAVASCRIPT –**

--"**JS**" is a high-level, interpreted programming language, which is also characterized as dynamic, weakly typed, prototype-based and multi-paradigm.

--Key differences between Java and JavaScript:

- Java is an OOP programming language while Java Script is an OOP scripting language.
- Java creates applications that run in a virtual machine or browser while JavaScript code is run on a browser only.
- Java code needs to be compiled while JavaScript code are all in text.
- They require different plug-ins.

❖ **LOW LEVEL LANGUAGE –**

--A **Machine Language**(Binary) or an assembly language. Low-level languages are closer to the hardware than are high-level programming languages, which are closer to human languages.

❖ **METHOD –**

--A method is really the same as a *procedure*, *function*, or *routine in procedural programming languages*. The only difference is that in object-oriented programming, a method is always associated with a class.

❖ NOTEPAD++ --

--Notepad++: a free source code editor which supports several programming languages running under the MS Windows environment.

❖ OBJECT MODULE –

--Also called **object file**, an object module is a section of code used in object-oriented programming that can be used by other object modules or the program being created. The object modules contain a specific function such as enabling a device on a computer.

❖ OBJECT-ORIENTED –

--A type of programming in which programmers define not only the data type of a data structure, but also the types of operations (functions) that can be applied to the data structure. In this way, the data structure becomes an object that **includes both data and functions**. In addition, programmers can create relationships between one object and another. For example, objects can **inherit** characteristics from other objects.

--One of the principal advantages of object-oriented programming techniques over procedural programming techniques is that they enable programmers to create modules that do not need to be changed when a new type of object is added. A programmer can simply create a new object that inherits many of its features from existing objects. Object-oriented programs are easier to modify. --In object-oriented programming, a procedure that is executed when an object receives a message.

❖ PROGRAMMING –

--writing code with different parts of a form and turning it into an application.

--Also called link editor and binder, a linker is a program that combines object modules to form an executable program. Many programming languages allow you to write different pieces of code, called modules, separately. This simplifies the programming task because you can break a large program into small, more manageable pieces. Eventually, though, you need to put all the modules together. This is the job of the linker.

❖ OCTAL – --Base-8 Number System. Each octal digit is exactly equivalent to 3 bits.

--mainly used as a short-hand for binary.

--B:10101010 = O:252 = B:(0)10-101-010 = O:2-5-2

❖ PYTHON –

--an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, and a syntax that allows programmers to express concepts in fewer lines of code,[25][26]notably using significant whitespace.

--It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

❖ REGISTERS –

-- Storage Component of electronics. (Most hold a 32-bit Binary Number)

--**Register File** – Contains 32 Registers. Range of Register Values =decimal values represented by 32 bits = - 2,147,483,648(-2³¹) TO 2,147,483,647(2³¹ -1)

-- **Instruction Register (IR)** – 32-bit Register. Holds a copy of most Recent fetched Instructions.

❖ REGISTER FORMATS –

Instruction Formats (3)- Register Format / Immediate Format / Jump Format

Immediate Format – Perform Arithmetic/Logical operations between **Variable** (Stored in Register File)& **Constant** (Stored in Instruction). Branch Instructions / Load and Store Instructions = this format.

Jump Format – Equivalent of “Go To” Instruction (High-Level Languages)

-----4 STEPS of OPERATIONS to FETCH/EXECUTE any REGISTER FORMAT INSTRUCTIONS.

1. **Instructions Fetch Phase** – Instruction, Fetched from Memory, Location specified by PC. Is Loaded into IR. PC is incremented by 4.
2. **Operand Fetch Phase** – 2x 5-bit codes, Rs and Rt, within Instructions specify which Register File Locations are used to get the 2x 32-bit Source Operands. Decode the Op-Code.
3. **Execute Phase** – 2x 32-bit Source Operands are Routed to ALU Inputs(possible Operations performed there, depending on Op-Code in Instructions)
4. **Write Back Phase** – Result of Operation, placed into specified(by 5-bit Rd code in IR) Register File. Go to Step 1.

❖ **REGISTER ZERO –**

--Special Register. Nothing is stored. It is the Source of the constant value of 0. Cannot be modified.

❖ **SYSTEMS SOFTWARE –**

Software designed to operate the hardware & provide a platform for running applications software.

❖ **TWO'S COMPLEMENT –**

--Invert Numbers, then add +1. EX: 5= 0101. Two's compliment = $1010+1= 1011 = -5$

-- $1011 - 1 = 1010$, Invert = 0101 = 5

--Used to represent Negative (Signed) Numbers. Such as subtraction. ($2 - 3 = 2 + -3$)

--Most Significant Digit (MSD) = Most Far Left bit. If it's "1", it's Negative.

❖ **VISUAL BASIC –**

--an Event-Driven Programming Language and integrated development environment(IDE) ,

from [Microsoft](#), for its **Component Object Model (COM)** programming model first released in 1991,

--Software development environment created by Microsoft to make it easy for developers to create client applications on Windows. It featured an way to easily create graphical user interfaces by drawing windows and controls on the screen, and an event-driven paradigm that made it easy to attach code to meaningful events in the lifetime of an application (such as a button click).

--Derived from **BASIC**