**1. Programming Languages**

GOALS

1. Know what Low-Level Machine Language looks like

2. Know what High-Level Programming Language looks like

3. Describe how to Translate High-Level language into Low-Level instruction

4. State the Main Features of C, C++, Java, Python

Programming is:

* Implementing a Representation of a solution for the Computer to Execute
* Taking an Algorithm and Encoding it

Programming Language:

* A medium through which Programmer gives instructions to Computer
* Must support Control Constructs and Data types to implement Algorithms

eg.

Sequential Processing

Decision-Making Selection

Iteration

Numbers

Characters and Strings

**Machine Language**

Machine Language is entirely in 01101001

It is specific to the Processor.

It is Long, Tedious, Error-Prone.

Therefore, we Invented High Level Programming Languages

**Programming Languages**

* Provide Abstraction from the internal operating detail of the computer

(working with ideas)

* Simpler to Develop Algorithms
* Enable Programmers to Focus on Solving Problems

Most programs are coded in High Level Programming Langauges

**Program Translations**

High Level Code 🡪 Low Level Machine Language 🡪 Instruct Computer Operations

There are two approaches for translation:

1. Interpretation

- Reads One high level statement at a time

- Translates and Executes immediately before going to the next line

[Python, R, JavaScript]

2. Compilation

- Reads and Translates the entire source code into Machine Level Instructions

- The Machine Language Instruction can now be executed directly on the computer

[C, C++]

Java uses a combination of Both Approaches.

Interpreters are:

* Portable across Computing Platforms
* Give Immediate Results
* Easy to Debug
* Implement Dynamic, Interactive Features (web pages)

Compiler Code:

* runs FAST after Compiling
* is Smaller
* is used in Large, Sophisticated software when Speed is of MAX Importance
* needs to be Recompiled for Different Platforms

**C**

C is a Compiled Language:

* Designed to help implement Unix OS
* Allows for Direct Access to Computer Hardware
* Fast

Applications that need:

1. Direct Access to Hardware

2. Fast Real-Time Response

[Embedded Systems Development, OS Implementations, Device Drivers]

**C++**

C++ is a Compiled Langauge:

* Extension of C
* Supports OOP (Object Oriented Programming)

Applications that need:

1. High Performance

2. Object Oriented Software Design

[Video Games, Graphic UI applications, Modern OS Design]

**Python**

Python is an Interpreted Language:

* Easy to Learn
* General Purpose Programming
* Cannot Access Hardware

Applications that need:

1. Instant Feedback to the User

[Data Science, Web Dev]

**Java**

Java is a combination of Compiler and Interpreter:

* Highly Portable General Purpose Language
* OO support
* Cannot Access Hardware

Applications that need:

* A Combination of Compiling and Interpreting
* A Balance of Performance

This is done with JIT (just in time compiler)

1. Source Code .java is compiled to Java bytecode .class

2. Java bytecode .class is interpreted by JVM (java virtual machine) during execution

3. JVM may call JIT to compile some bytecode .class at runtime before execution.

This achieves better performance.

[Mobile Apps, Game Dev]