ROBOT PROGRAMMING LANGUAGES

Robot programming languages play a crucial role in enabling effective communication between humans and robotic systems. These languages provide a structured way to instruct robots on how to move, sense their environment, interact with objects, and perform complex tasks with precision. Unlike general-purpose programming languages, robot programming languages are designed with built-in commands for motion control, sensor integration, end-effector operations, and task sequencing.

In industrial robotics, programming languages such as **VAL, RAPID, KRL, and ROS-based frameworks** allow engineers to develop programs that define trajectories, speeds, force limits, and coordinated movements. These languages support different programming paradigms, including **teach pendant programming, lead-through programming, and offline programming**, giving flexibility based on the application.

Robot programming languages are essential in modern manufacturing because they help automate tasks in **inspection, assembly, welding, material handling, medical robotics, underwater operations, and space missions**. By providing high-level abstractions for robot actions, they simplify the process of controlling complex robotic mechanisms while ensuring accuracy, repeatability, and safety.

VARIABLE ASSEMBLY LANGUAGE (VAL):

Variable Assembly Language (VAL) is one of the earliest and most influential robot programming languages developed for industrial robots. It was designed by Adept Technology to control assembly-line robots. VAL allows programmers to define robot actions such as movement, gripping, sensing, and task sequencing using high-level English-like commands, making it easier than low-level assembly language.

Although called *Variable Assembly Language*, it is NOT a low-level assembly language.
It is a robot-specific high-level language used mainly for:

- Assembly operations

- Pick-and-place tasks
- Industrial automation
- Handling materials in manufacturing units

SAMPLE VAL PROGRAM

Program Name: ASSEMBLYTASK

```
SPEED 30          ; Set speed

; Move to component pick-up location
MOVE PICK         ; Go to PICK position
CLOSE             ; Grip the component
APPRO PICK        ; Lift slightly for safety

; Move to assembly position
MOVE ASSEMBLE     ; Go to assembly position
OPEN              ; Release component
DEPART ASSEMBLE   ; Move away

END
```