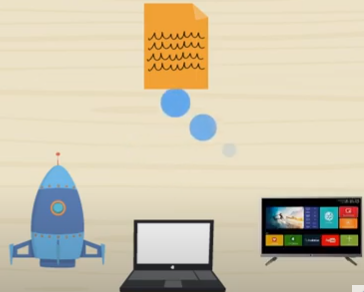
**Programming Paradigms**

* Paradigm: method to solve some problem or do some task.
* Programming paradigm: approach to solve problem using some programming language or also we can say it is a method to solve a problem using tools and techniques.

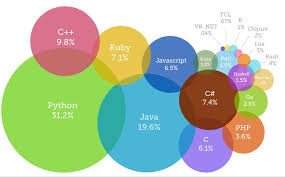
**Programming Languages:**

* Designed to communicate instructions to the machine
* Used to create programs that controls the behavior of the machine(Laptop, Rocket, TV etc)

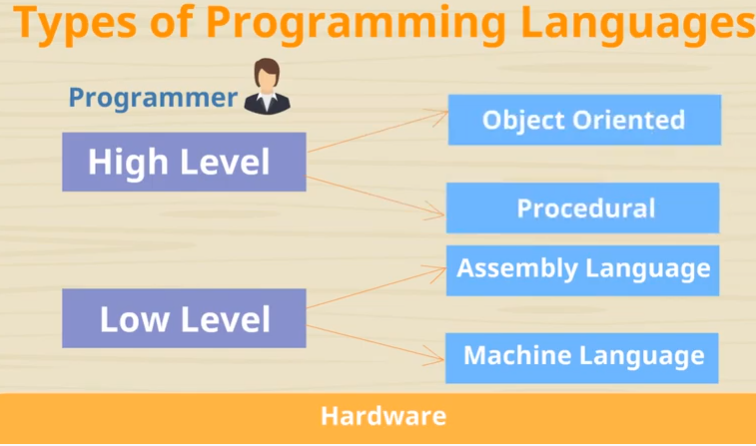


**Program**:

* List of instructions written in programming languages.
* Popular Programming Languages



**Different Types of Programming Languages:**



**Low-level:** Machine understands

**High-Level:** User friendly. Closer to user language

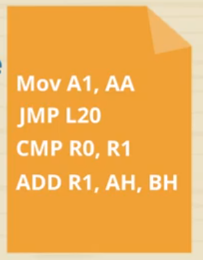
**Machine Language:**

* Directly run on CPU
* Series of bits like 0’s and 1’s
* Tedious and error prone to write code manually
* Not Portable. Specific to particular type of machine
* All languages need to be translated to Machine level – languages.



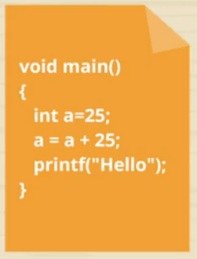
**Assembly Language:**

* Less error prone
* Coding easier than machine level languages
* Replaces 0’s and 1’s with English instructions
* Mnemonic codes for corresponding machine languages



**High- Level Languages:**

* It is portable
* Statements are like English languages
* Amount of abstraction provided defines level of programming language



|  |  |
| --- | --- |
| 1. Procedural language: | 2. Object- Oriented Language: |
| Program is written as sequence of instructions | Program is an interaction of functions between objects |
| Eg. Recipes, Morning Steps |  |
|  |  |
| Top Down Approach | Bottom-Up approach |
| More Focus on Functions | More Focus on data |
| It doesn’t have proper way of hiding data | Helps in wrapping data and functions in a class |
| Procedural programming does not have any proper way for hiding data so it is less secure. | Object oriented programming provides data hiding so it is more secure. |
| Data Not Secure | Helps building secure programs |
| Code is interdependent | Code is modular |
| Reuse Difficult | Can be extended for reuse |

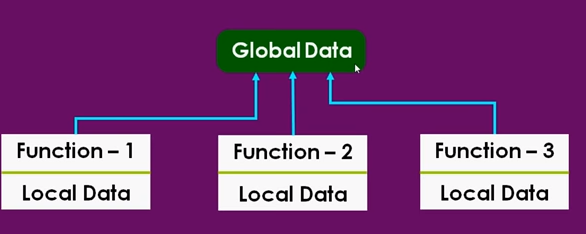
|  |  |
| --- | --- |
| Compiler | Interpreter |
| Translates high-level instructions into machine instructions | Translates high-level instructions into machine instructions |
| Generates an executable file.exe | Each and every line is executed individually |

|  |  |
| --- | --- |
| Top-down Approach | Bottom-up Approach |
| A top-down approach is essentially the breaking down of a program to gain insight into its compositional small program (or module) in a reverse engineering fashion. | A bottom-up approach is the piecing together of module (or small program) to give rise to more complex program, thus making the original modules of the emergent program. |
| Structure / procedure oriented programming languages like C programming language follows top-down approach. | Object oriented programming languages like C++ and JAVA programming language follows bottom-up approach. |
| A top-down approach begins with high level design and ends with low level design or development. | A bottom-up approach begins with low level design or development and ends with high level design. |
| In top-down approach, main function is written first and all sub functions are called from main function thus, sub-functions are written based on the requirement | In bottom-up approach, code is developed from modules and then these modules are integrated with main function |

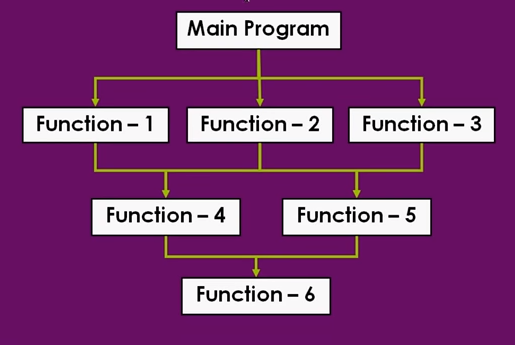
There are lots for programming language that are known but all of them need to follow some strategy when they are implemented and this methodology/strategy is paradigms. Apart from varieties of programming language there are lots of paradigms to fulfil each and every demand. They are,

Procedural Programming :

* + Programs are decomposed into number of subprograms- Functions
  + Each function accomplishes a task
  + Very little attention is given to data
    - Each function have local data
    - Program can have global data
    - Every function access global data
    - Difficult to track the values

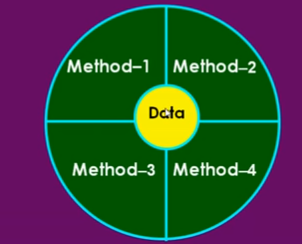


* Top-Down Approach



Object-Oriented Programming: Car

* + Eliminates flaws of procedural language
  + Treat data as critical element
  + Ties data more closely to the function



* + Data is hidden and cannot be accessed external functions
  + Bottom-Up approach
  + Object=Data+Methods
  + Natural way to programming
    - Object-Real world entity. Characteristics are,
      * Identity: Name of an object
      * Behavior: What an object does(Methods)
      * State: Describes object’s data
    - Classification of objects
      * Objects are grouped - Similar characteristics
    - Object composition
      * House object is composed of several bricks object
    - Object interaction
      * Interact to accomplish a task
        + Car example
    - Using Object in programs
      * World consist of Interacting, identifiable and classifiable objects. Programs are structured in this way
    - Represent real-world in,
      * Design
      * Working