Software

OS, Applications and Programming

What is Software

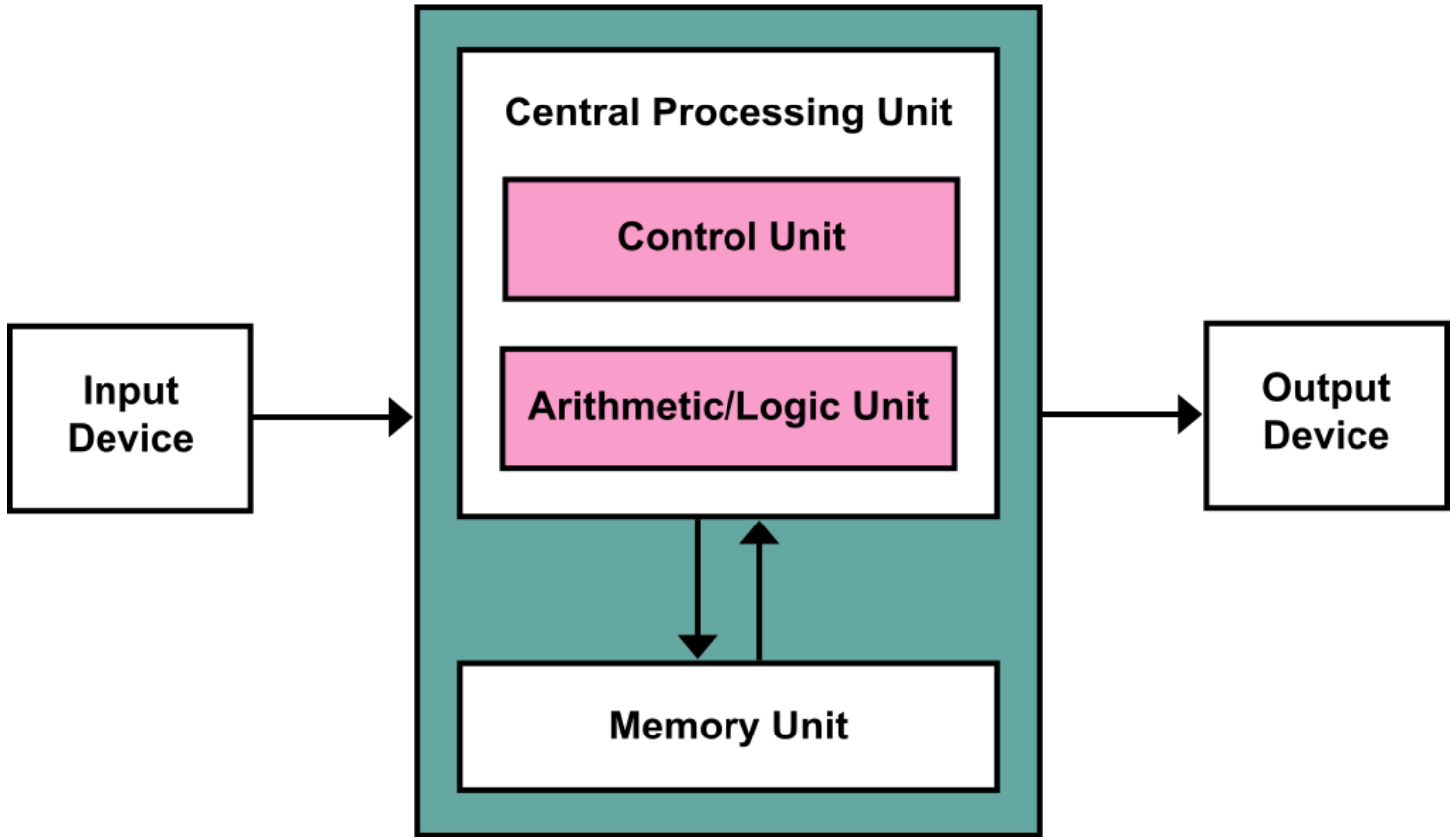
- Software refers to a set of instructions or programs that tell a computer what to do.
- Types
  - System Software: System software includes operating systems, device drivers, utilities, and other essential programs that facilitate the operation of computer hardware and provide a platform for running application software.
  - Application Software: Application software comprises programs designed to perform specific tasks or provide services to users. This category encompasses a vast array of software, including word processors, web browsers, media players, games, and business applications.

Machine Language

- At the lowest level of abstraction, computers understand instructions in the form of binary digits, known as machine code or machine language.
- Machine code is composed of sequences of 0s and 1s, representing basic operations such as arithmetic, logic, and data movement.
- While efficient for computers, machine code is complex and cumbersome for humans to work with directly.

Von Neumann Mache Code

- loading data from RAM to CPU register
- storing data from CPU register to RAM
- jumping to a different instruction
- performing an arithmetic or logic operation



Machine code	Assembly code	Description
001 1 000010	LOAD #2	Load the value 2 into the Accumulator
010 0 001101	STORE 13	Store the value of the Accumulator in memory location 13
001 1 000101	LOAD #5	Load the value 5 into the Accumulator
010 0 001110	STORE 14	Store the value of the Accumulator in memory location 14
001 0 001101	LOAD 13	Load the value of memory location 13 into the Accumulator
011 0 001110	ADD 14	Add the value of memory location 14 to the Accumulator
010 0 001111	STORE 15	Store the value of the Accumulator in memory location 15
111 0 000000	HALT	Stop execution

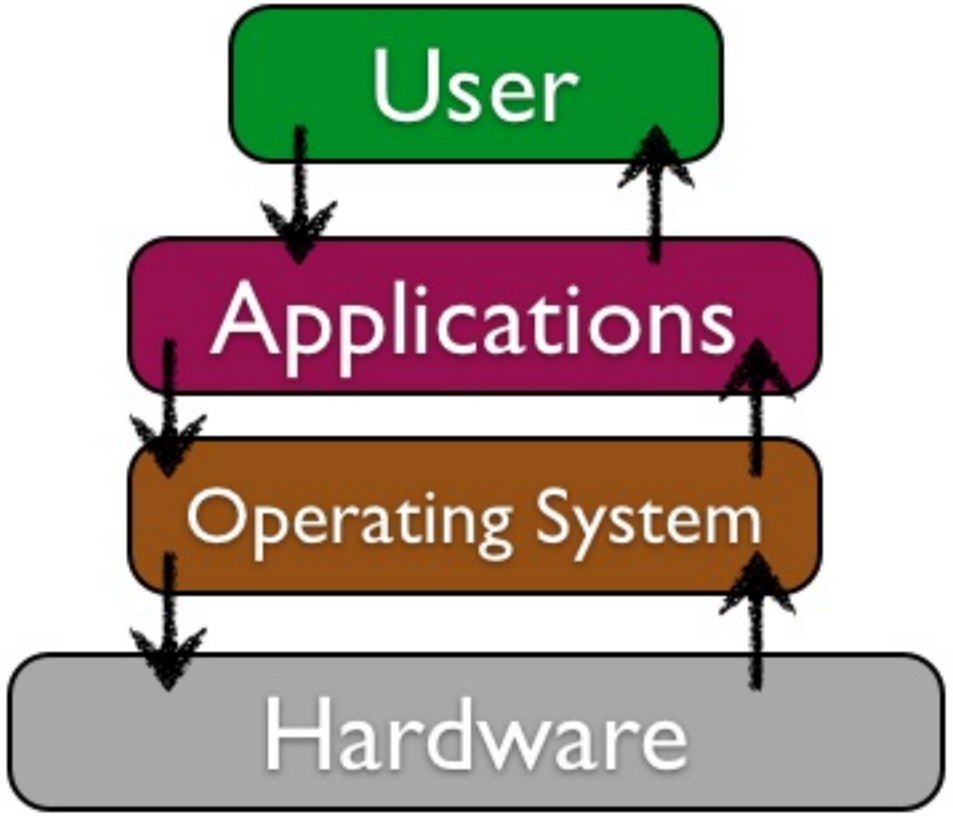
Machine Code

Instruction Set Architecture (ISA)

- An ISA is an abstract model of a computer that defines a set of instructions for a specific CPU.
- Different CPUs have different ISAs.
- Commonly used ISAs include
  - Intel x86 and x86-64: used by most desktop and server computers. AMD64 is x86 compatible.
  - ARM: used by most mobile processors
  - RISC-V: relatively new, open source

Operating System (OS)

- An operating system (OS) is the software that manages and controls a computer's hardware and software resources, providing a platform for running applications and performing various tasks.
- The OS acts as an intermediary between the user and the computer hardware, making it possible for users to interact with the computer in a more intuitive and efficient way.

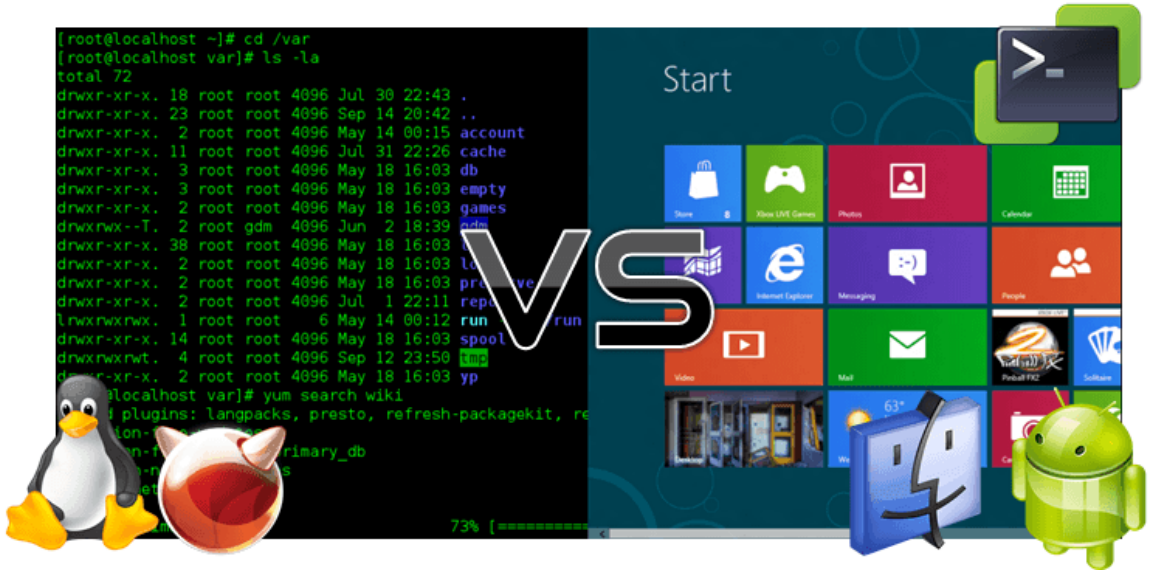


OS Functions

- Process Management: The OS manages the creation, execution, and termination of processes (programs) running on the computer.
- Memory Management: The OS allocates and deallocates memory for programs, ensuring efficient use of system resources.
- File System Management: The OS provides a file system, allowing users to create, delete, and manage files and directories.
- Input/Output Management: The OS handles input/output operations between devices and programs.
- Security: The OS provides mechanisms for user authentication, access control, and protection against malware and viruses.

User Interface (UI)

- Graphical User Interface (GUI): A visual interface using icons, windows, and menus, allowing users to interact with the computer using a mouse and keyboard. Examples: Windows, macOS, Android.
- Command-Line Interface (CLI): A text-based interface where users enter commands using a keyboard, ideal for advanced users and automation.



CLI and Shell

- CLI requires users to enter specific commands using the keyboard to interact with the system.
- This interface is ideal for advanced users who prefer the efficiency and speed that comes with typing commands.
- The command-line interface (CLI) in Linux is called ***shell*** that
  - enables users to interact with the operating system and execute commands.
  - acts as a layer between the user and the kernel, providing a way to communicate with the system and access its services.
  - sets and manages **environment variables**, which are used to store information about the system and user preferences.

Linux Shell Demo

- The *cd* command changes the current directory
- The *ls* command lists files and directories
- The *mkdir* command creates a new directory
- The *rm* command removes files or directories.
- The *cp* command copies files
- The *mv* command moves or renames files.
- The *echo* command prints text to the screen
- The *man* command displays manual pages for commands and functions.

Shell Programming

- The shell supports scripting languages such as Bash, Python, and Perl, enabling users to write complex scripts to automate repetitive tasks, streamline workflows, and customize system behavior.
- This flexibility and extensibility make the Linux shell an indispensable tool for both novice users seeking to automate routine tasks and seasoned developers crafting sophisticated automation solutions.

Shell Code Demo

```
#!/bin/bash

# Prompt the user for their name
echo "What's your name?"

# Read the user's input into a variable
read name

# Greet the user
echo "Hello, $name! Welcome to the Bash scripting world."
```

GUI vs CLI

- GUI is generally considered more user-friendly and accessible, especially for beginners.
- CLI offers faster execution speeds and greater control over system processes, making it a preferred choice for advanced users and power users.
  - For software developers, data analysts, and IT professionals, proficiency in Command Line Interface (CLI) is an essential skill to possess.
  - Shell is a MUST-to-have for most operating systems

Applications: Local or Cloud

- Productivity Software: Helps with tasks like word processing, spreadsheet analysis, and presentation design. Examples: Microsoft Office, LibreOffice.
- Graphics and Multimedia Software: Used for image and video editing, graphic design, and audio editing. Examples: Adobe Photoshop, Audacity.
- Games: Entertainment software that provides interactive experiences. Examples: Minecraft, Fortnite.
- Educational Software: Teaches new skills or subjects, often used in schools. Examples: Duolingo, Khan Academy.
- Utility Software: Performs maintenance or management tasks, like disk cleanup or virus scanning. Examples: Norton Antivirus, CCleaner.

Programming/Coding

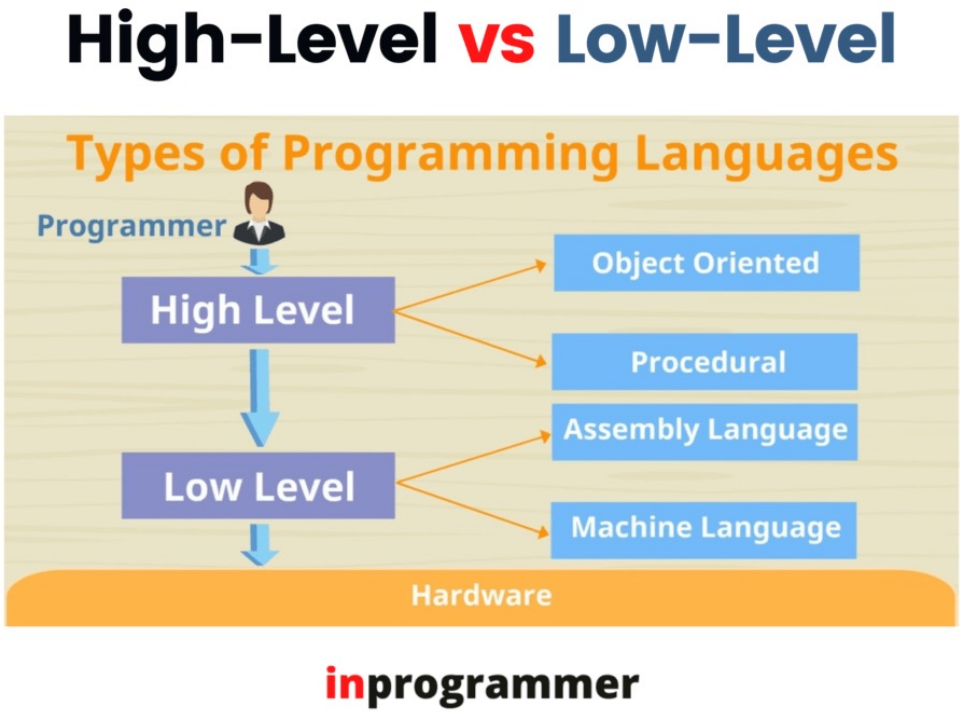
- It is a systematic process that involves designing, writing, testing, and deploying sequences of instructions that computers can execute to perform specific tasks.
- Core Activities
  - Designing algorithms
  - Writing code
  - Testing
  - Deploying

More Than Coding

- Performance
- Security
- Scalability
- Maintainability

Programming Languages

- High-level programming languages are abstracted from the computer hardware, making them easier to read and write.
  - Examples: Python, Java, C#, JavaScript, Ruby, PHP
- Low-level programming languages are closer to the computer hardware, making them more difficult to read and write.
  - Examples: Assembly languages, machine code, C



Hello World! Programm

PYTHON

```
>>> print("Hello World!")
Hello World!
```

C++

```
#include <iostream>

int main() {
    std::cout << "Hello World!";
    return 0;
}
```

JAVA

```
class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, World!");
    }
}
```

ASSEMBELY

Compiling

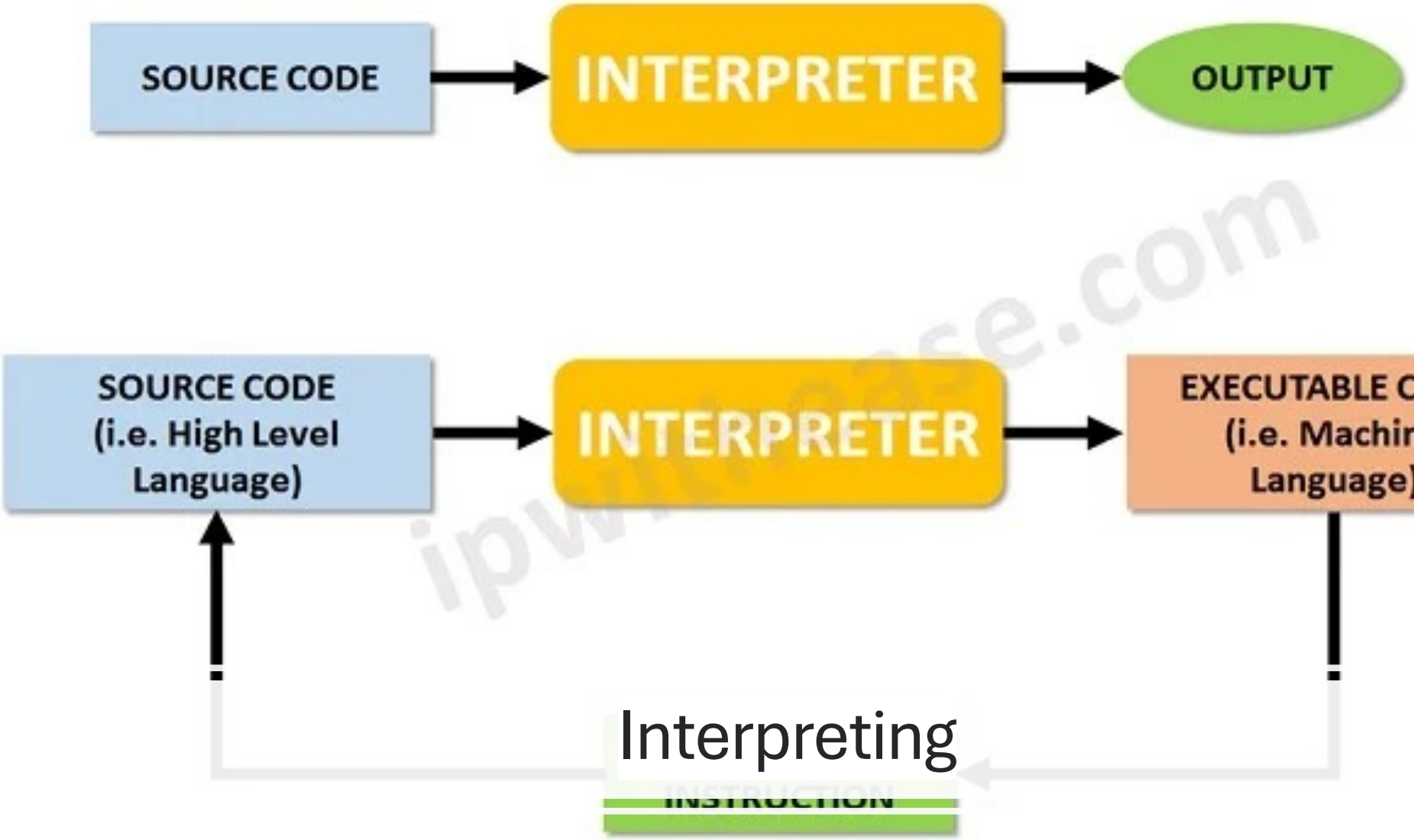
```
#include
<stdio.h>
int main()
{
printf(“Hello”)
;
return 0;
}
```

Source code

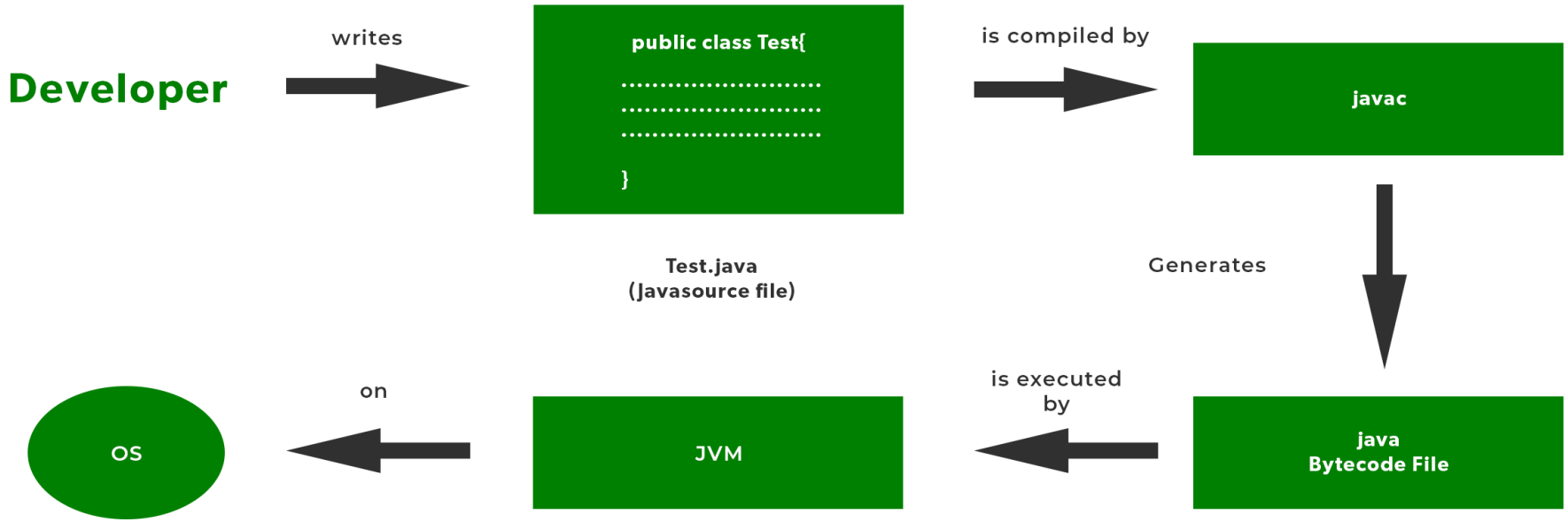

```
1000101010101010
0001001010101011
0111111001100000
0010110011010100
0101110111000110
0111110011110000
0001100111101010
0100100101010000
```

Executable code

How Interpreter works?



ipwithease.com



Step from java.code to machine code

- java source code is compiled.
- Transformed to bytecode by java compiler.
- Interpreted and executed by the java virtual machine on the underlying operating system.

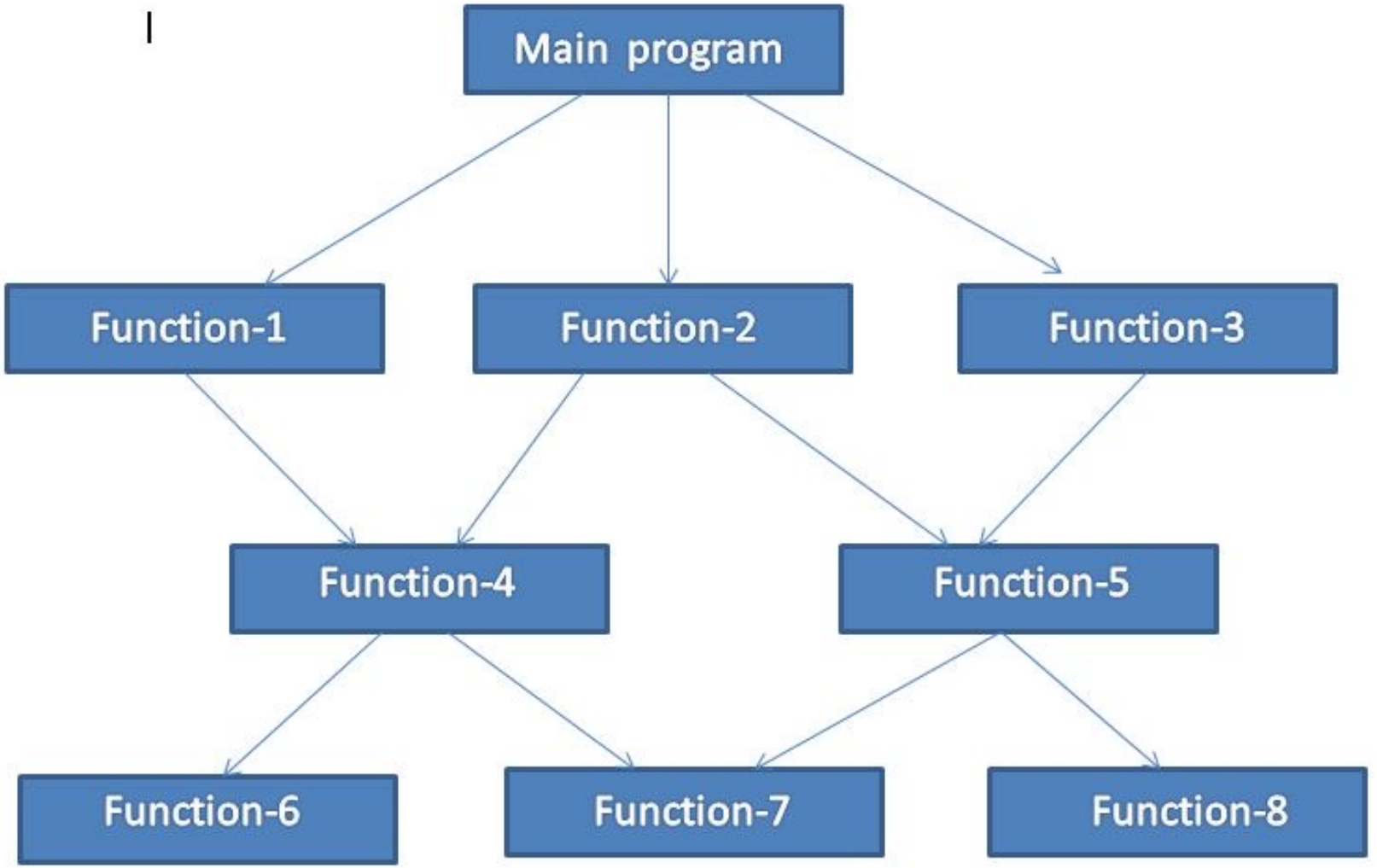
Compiling + Interpreting

Programming Paradigms

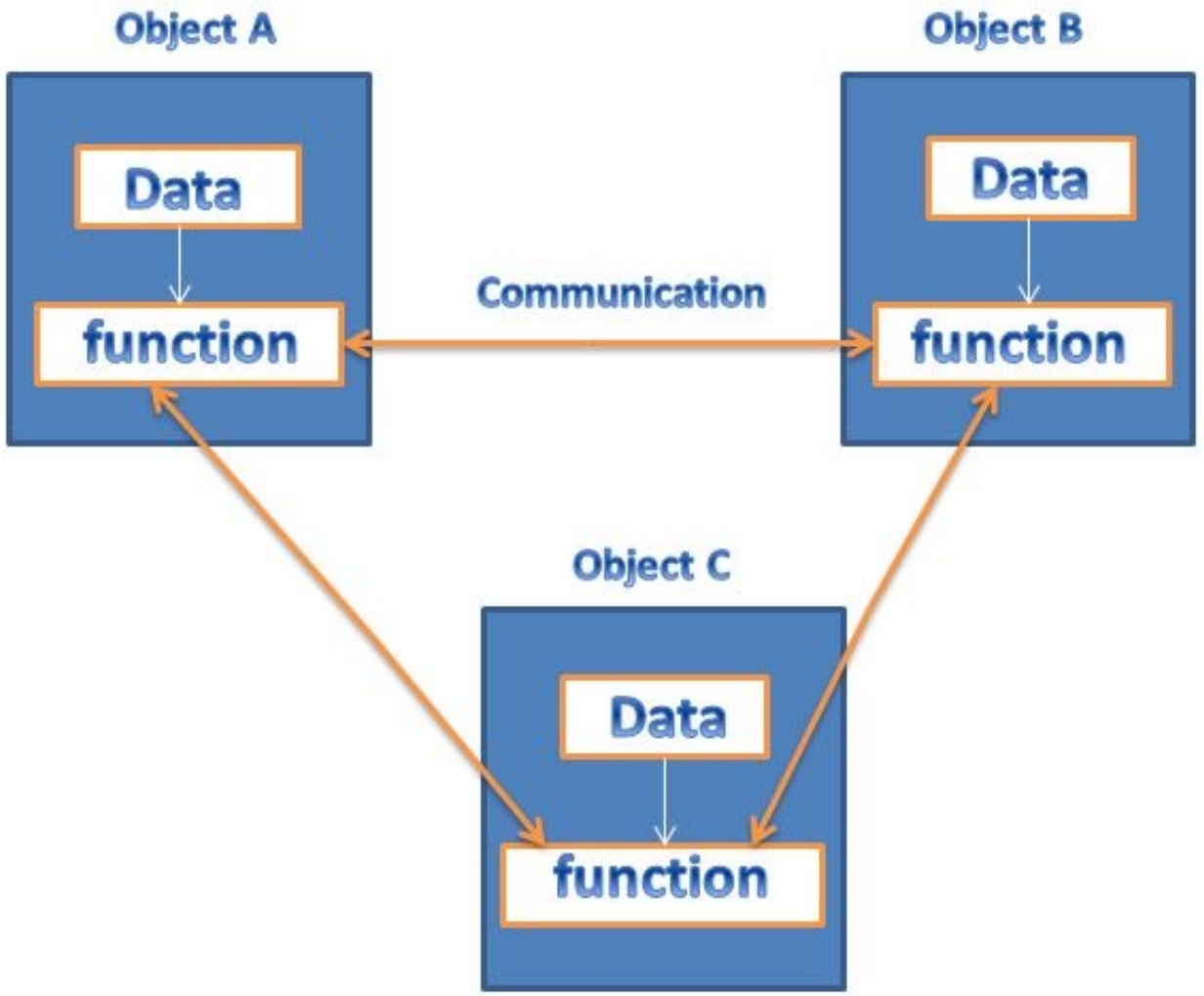
- Styles or approaches to writing software
- Three main programming paradigms have emerged:
  - Procedural Programming
  - Object-Oriented Programming (OOP)
  - Functional Programming.
- Each paradigm has its unique principles, advantages, and applications, which are essential to understand in order to create effective and maintainable software solutions.

Procedural

- It focuses on procedures and functions that perform specific tasks.
- A procedural program is like a recipe, where a set of instructions (procedures) are followed in a specific order to produce a desired outcome.
- This paradigm emphasizes step-by-step execution, sequential flow, and explicit control over data and program flow.



Structure of procedure oriented programming



ORGANIZATION OF DATA AND FUNCTIONS IN OOPS

Functional Programming

- First-class function: a function works like a data that can be passed as a parameter of another function.
- Pure functions: Each function has no side effects and always returns the same output given the same input.
- Immutability: The data of the program is not modified by the functions.

Software Libs

- A software library is a collection of pre-written code, functions, and tools that developers can use to perform specific tasks or solve common problems.
- It provides a way to reuse code, reducing the need to write everything from scratch, and enables developers to focus on the unique aspects of their project.
- As an example, the Math library is a built-in JavaScript library that provides mathematical functions, such as sqrt() and pow()

Software Packages

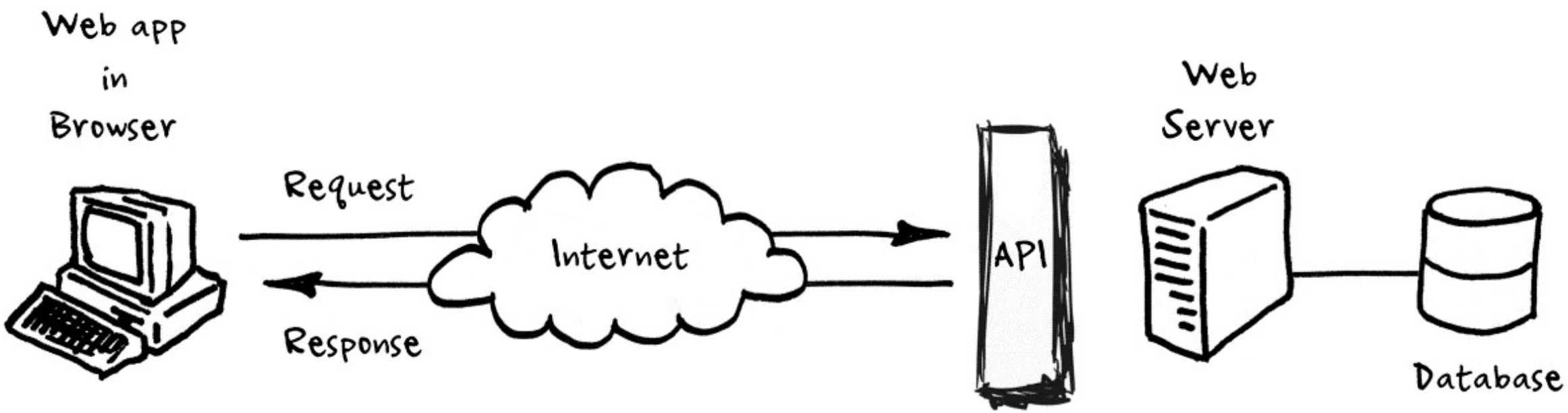
- A software package is a collection of software tools that can be installed and configured in a consistent manner.
- It is typically much larger than a software library. Multiple software libraries can be combined into a single package.
- Software package is not only used in programming. Applications or application components can also be delivered as a software package.

Package Management

- NPM for JavaScript
- Pip for Python
- Gradle or Maven for Java
- Windows installer for Windows
- Brew for MacOS

API

- Software API (Application Programming Interface) is the interface between software systems, enabling them to communicate and exchange data seamlessly.
- Just as a user interface (UI) facilitates interaction between humans and software, API acts as the intermediary between software applications, allowing them to request services, exchange data, and leverage each other's functionality.



API is Everywhere

Software API

- Enabling communication between software systems
- Providing access to pre-built functionality
- Facilitating data exchange and integration
- Saving development time and resources,
- Promoting modularity and reusability of code

- Examples: Linux API, Stripe API.

API Standards

- REST (Representational State of Resource) is a widely used API standard that leverages the HTTP protocol. It identifies resources using URIs and supports CRUD (Create, Read, Update, Delete) operations. RESTful APIs are stateless, meaning each request contains all the necessary information to complete the request. This standard is ideal for simple and lightweight APIs.
- GraphQL, on the other hand, is a query language for APIs that allows clients to specify exactly what data they need. The server then returns only the requested data, reducing bandwidth and improving performance. GraphQL has strong typing and validation, making it a popular choice for complex and scalable APIs.
- gRPC is a high-performance RPC (Remote Procedure Call) framework that uses Protocol Buffers for serialization and deserialization. It offers strong typing and validation, making it suitable for large-scale and mission-critical APIs. gRPC is particularly useful for building micro-services and distributed systems.

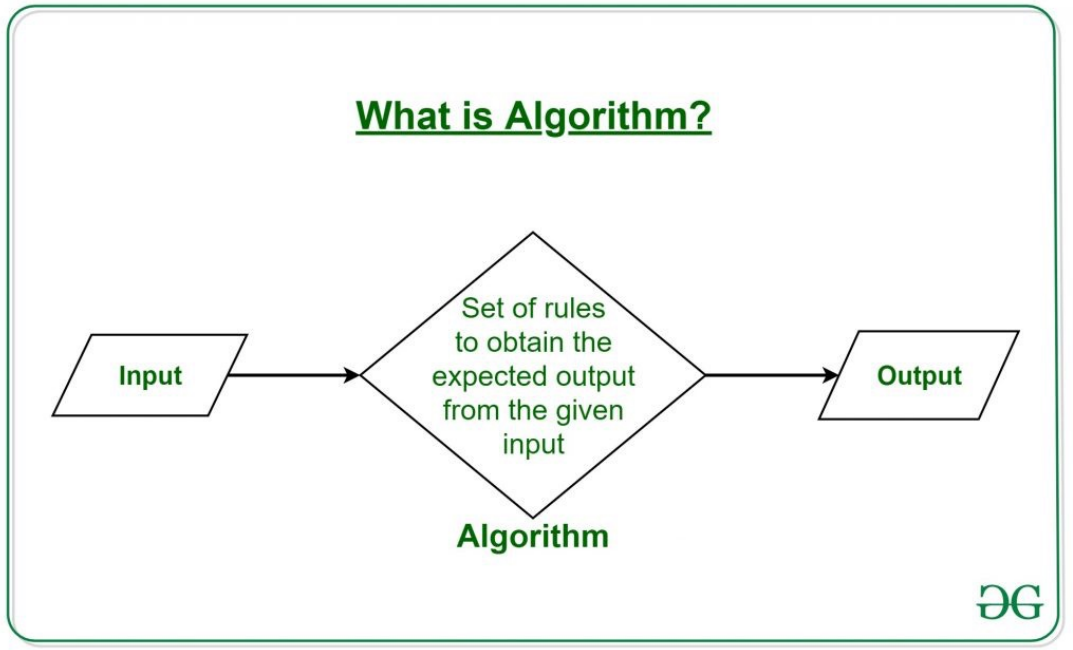
API as Core Business

- At the heart of Stripe's offerings lies its core service: Application Programming Interfaces (APIs).
- Stripe's APIs provide developers with a suite of tools and resources to facilitate secure payment processing, from tokenizing sensitive payment data to authorizing and processing transactions.
- Stripe's commitment to developer-friendly APIs empowers businesses of all sizes to innovate and grow in the digital economy.

- How to Create Stripe Payment Links to Add to Any Website: https://www.youtube.com/watch?v=JqQXgWPZQEw

Algorithm

- In the realm of computer science, an algorithm is a precise, step-by-step procedure or set of rules that defines how to solve a particular problem or perform a specific task.
- Algorithms can range from simple procedures, such as sorting a list of numbers, to complex algorithms used in artificial intelligence and machine learning.
- Algorithms must be
  - well-defined and unambiguous
  - Finite
  - Deterministic



Algorithmic Thinking

- Algorithmic thinking, or computational thinking, is the ability to conceptualize, design, and analyze algorithms to solve problems effectively.
- Algorithmic thinking enables individuals to harness the power of computers to solve a wide range of problems efficiently.
- From optimizing business processes to analyzing large datasets and developing innovative software solutions, algorithmic thinking empowers individuals to leverage technology to drive innovation and progress.