u3a Computing Group

Alan Hopwood, 4 January 2024

Agenda

Welcome
Current News, Issues and Questions
Future Topics & Next Meeting
Topic: Computer Languages
AOB and Follow up

Current News, Issues and Questions

Anything to discuss?

Future Topics

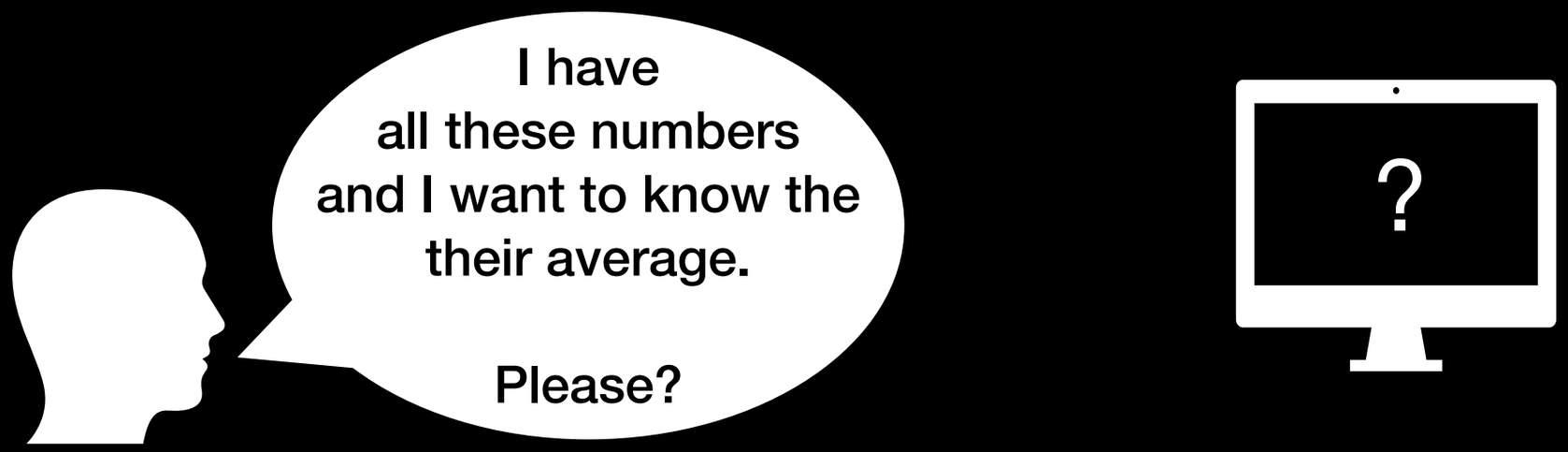
Topic	Votes
Laptop vs Tablet / smartphone	4
Microchip design	2
Being safe on the internet	2
History of computer development	2
Bluetooth	1
Digital communications / information encoding	1
Chromebook	1
Mac vs Windows vs Linux	1

Presentation
Computer languages
A very shallow dip into a very deep subject.

(Original question: Which language should I learn - functionality vs ease of use?)

The Purpose of a Computing Language

- The Challenge is to have a computer perform a specific task:



- How do you give instructions to a computer - that it can understand?

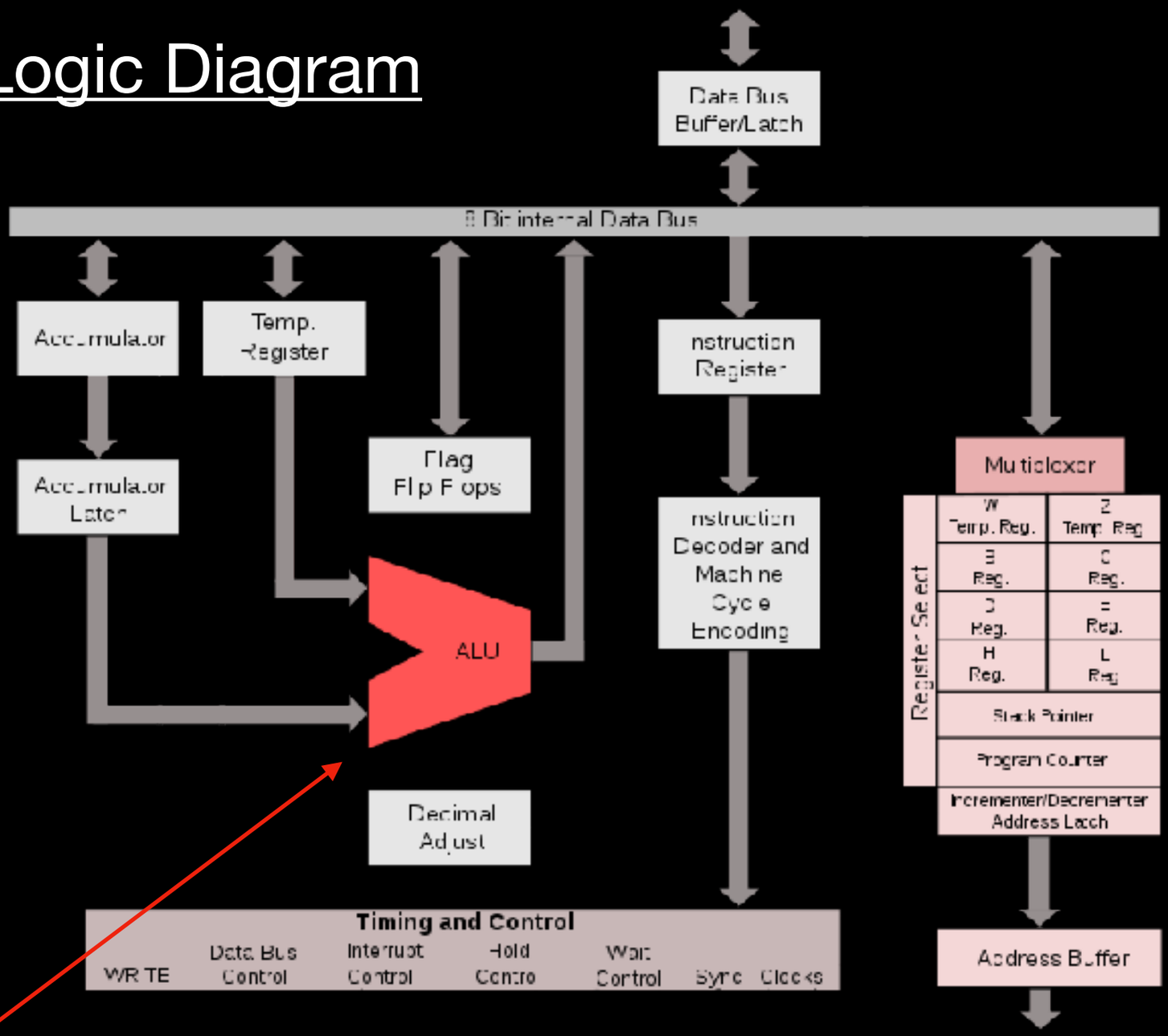
What does a Computer understand?

Computers have a CPU at their centre. We will use the intel 8080-8085 family to consider how they take instructions.

- The Intel 8080 (8085) is one of the simplest microprocessors still in use.
- 8-bit microprocessor (1974)
- One of the original microprocessors launching personal computing (also Zilog Z80, Motorola 6800, MOS Technology 6502)
- Although more recent CPUs are more complex, the RISC architectures (e.g. ARM) have a similar number of instructions.

The 8080/85 Microprocessors

Logic Diagram



Instructions (Machine Code)

```
1000
1000
1000  78
1001  B1
1002  C8
1003  1A
1004  77
1005  13
1006  23
1007  0B
1008  78
1009  B1
100A  C2 03 10
100D  C9
```

8085 has 246 instructions
Each instruction is an 8-bit binary value

The arithmetic logic unit (ALU) performs arithmetic and logic operations.

8080 programming in Assembler

Memory Address

Machine Code

Assembler Opcode - Operand

```
                ; memcpy --
                ; Copy a block of memory from one location to another.
                ;
                ; Entry registers
                ;       BC - Number of bytes to copy
                ;       DE - Address of source data block
                ;       HL - Address of target data block
                ;
                ; Return registers
                ;       BC - Zero

1000                    org     1000h           ;Origin at 1000h
1000            memcpy  public
1000  78                mov     a,b             ;Copy register B to register A
1001  B1                ora     c               ;Bitwise OR of A and C into register A
1002  C8                rz                      ;Return if the zero-flag is set high.
1003  1A        loop:   ldax    d               ;Load A from the address pointed by DE
1004  77                mov     m,a             ;Store A into the address pointed by HL
1005  13                inx     d               ;Increment DE
1006  23                inx     h               ;Increment HL
1007  0B                dcx     b               ;Decrement BC   (does not affect Flags)
1008  78                mov     a,b             ;Copy B to A    (so as to compare BC with zero)
1009  B1                ora     c               ;A = A | C      (are both B and C zero?)
100A  C2 03 10            jnz     loop          ;Jump to 'loop:' if the zero-flag is not set.
100D  C9                  ret                   ;Return
```

8085 Instruction Set in Assembler

Assembler is:

- Human readable notation for binary instructions understood by the CPU
- Translated to machine code by an assembler

Opcode	Operand	Description
MOV	Rd, Rs	Copy from source to destination.
ADD	R	Add register or memory to accumulator
ADC	R	Add register or memory to accumulator with
INR	R	Increment register or memory by 1
CPI	8-bit data	Compare immediate with accumulator
Jx	16-bit address	Jump conditionally

Types of Instructions

- **Data Transfer**: Copy data between registers and memory
- **Arithmetic Instructions**: Add, Subtract, Increment, Decrement
- **Logical Instructions**: AND, OR, XOR, Rotate, Compare, Complement
- **Branching Instructions**: Change program sequence location
- **Control Instructions**: Halt, enable/disable interrupt, read interrupt

Need for Programming Language

All programmes can be created in assembler

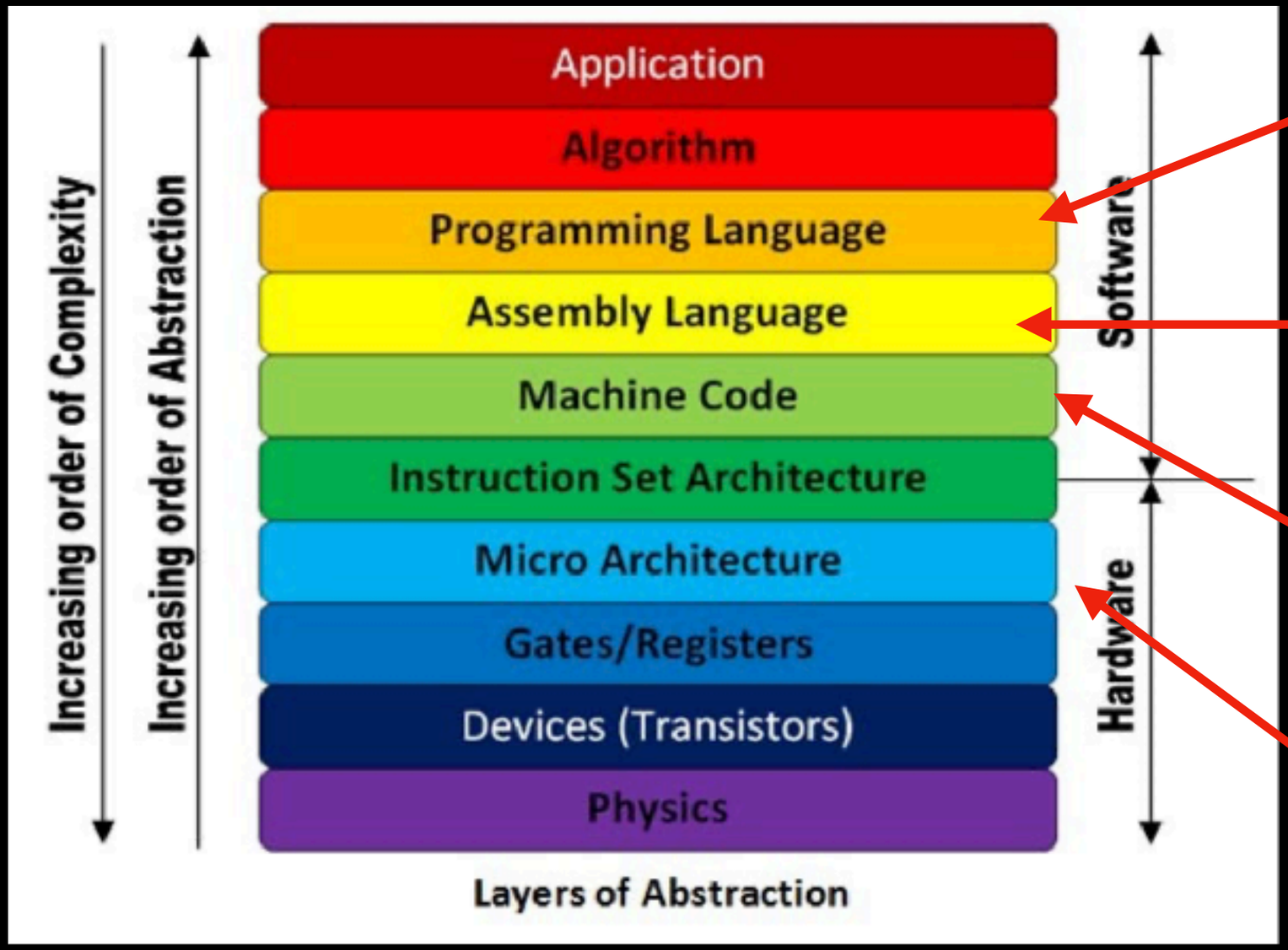
But

- Extremely complex
- Extremely time consuming
- Extremely error prone

- Need a “higher level” programming language
- Provide programmers with a more functional, more easily understood instruction set to define a program
- Have a compiler or interpreter program able to translate higher level program into assembler to run.

Computing Abstraction Model

Each layer communicates with adjacent layers



- Applications written in programming language
- Compiled or interpreted into assembler
- Assembled into machine code
- Run by the CPU+

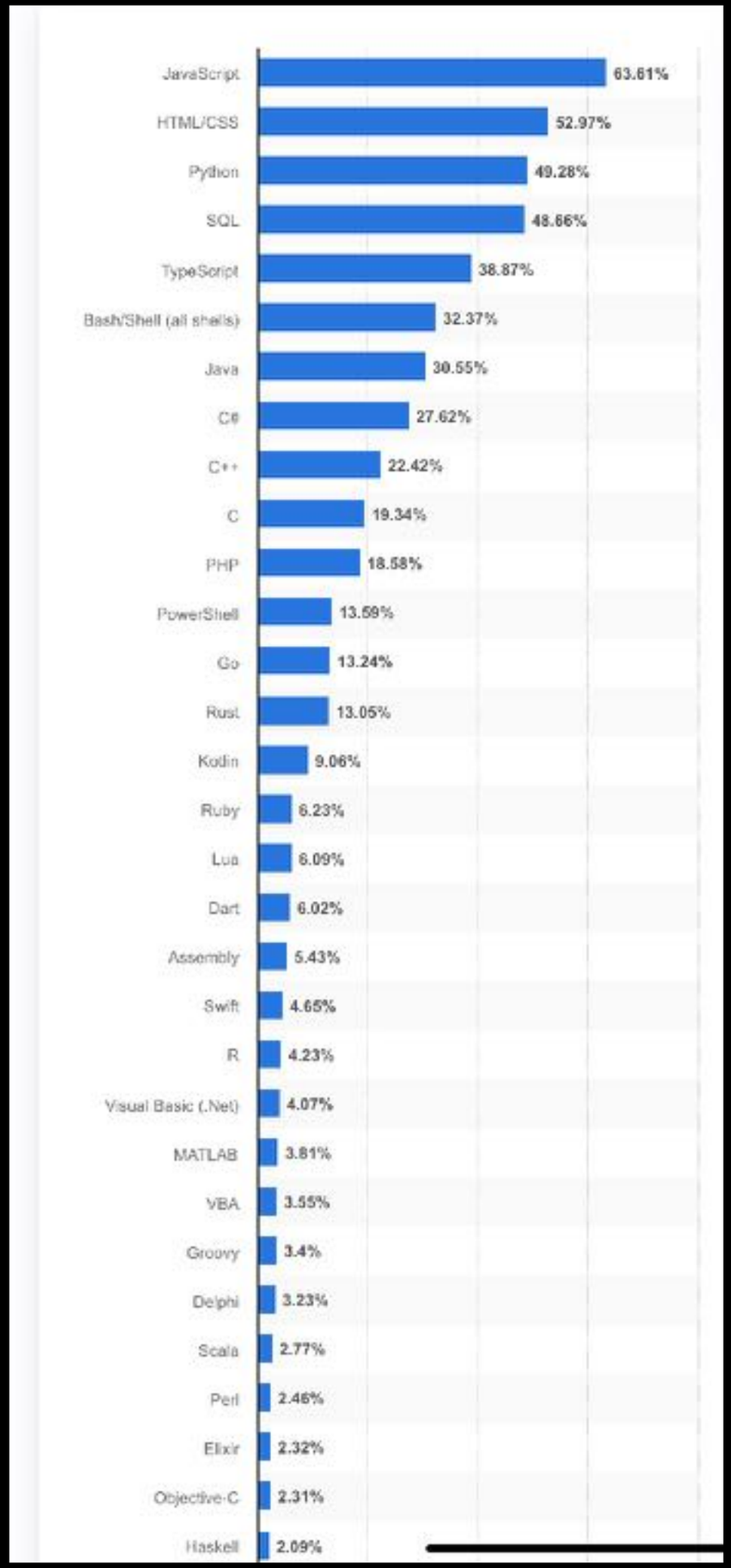
History of Computing Languages

- 1940s: Machine Code
- 1950s: Assembly Language
- 1960s:
  - Fortran
  - Cobol
  - Basic
- 1970s:
  - C
- 1980s:
  - Object Oriented languages - C++
  - Scripting, Web, component based
    Java, Perl, Python, Visual Basic, Java Script

Most used Languages

There are over 1,000 computing languages. Some of the most used are:

1. JavaScript
2. HTML/CSS
3. Python
7. Java
8 - 10. C and variants
13. Go
20. Swift



Computer Language Concepts - 1

Variable declaration

- Variables are containers for storing values.
- Variable declarations defines whether the variable is accessible throughout the program, or only within a specific Class or Method.

Control structures

- The Control structure specifies the flow of the program.
- For example: sequential, skip on condition, repeat until.

Data structures

- The way of storing data, e.g.
  - Arrays
  - Stacks: Last in, first out
  - Queues: first in, first out
  - Linked lists: use of pointers to memory locations

Computer Language Concepts - 2

Object-orientation

- Use of Objects and Classes
- An Object is a self contained combination of variables, functions, and data structures with specified input and output communication
- A class provides a predefined design base for an Object
- Encapsulation: Isolates variables, properties and methods with one “unit”
- Abstraction: Hiding details within an Object and only showing anything necessary for input & output.
- Inheritance: Creating a Class by linking to a “parent” Class and adding detail.
- Polymorphism: Making use of different Objects that have the same interface to achieve different results.

Computer Language Concepts - 3

Programming tools - Integrated Development Environments (IDEs)

- Applications allowing programmers to write, compile and execute code.
- They can:
  - Provide access to code dictionary/definitions
  - Identify syntax errors and auto-completion of code.
  - Provide GUI development tools.
  - Provide access to libraries and plug-ins.
  - Allow testing in a safe environment, including stepping through the program.

The traditional languages

- Fortran, 1957 (Formula Translating System)
  - Developed by IBM in the 1950s for scientific and engineering applications.
  - Used (arguably) the first optimising Compiler.
  - has been in use for over seven decades in computationally intensive areas such as numerical weather prediction, finite element analysis, computational fluid dynamics, geophysics, computational physics, crystallography
- Cobol, 1959 (common business-oriented language)
  - English-like programming language designed for business use
  - designed to be self-documenting and highly readable.

The traditional languages

- Basic, 1963 (Beginners' All-purpose Symbolic Instruction Code)
  - Designed for ease of use to enable students in non-scientific fields to use computers.
  - Became the de facto programming language for home computer systems.
  - Hobbyist computers almost always had a BASIC interpreter installed by default.
- C, 1972 (successor to B)
  - Developed at Bell Labs by Ritchie alongside the Unix project.
  - The Unix operating system was originally implemented in assembly language.
  - Richie wanted a programming language for developing utilities for the new platform

Deciding Which Language to Use

<u>Considerations</u>

- Target System
  - Smartphone, tablet - IOS, Android
  - Computer - Windows, Apple OSX, Unix
  - Internet / Website
  - DYI computing - Raspberry Pi, Arduino, BBC MicroBit
- Cross Platform capability
- What sort of programme
  - Computer Game
  - Artificial Intelligence
  - Data analysis
  - Mathematical algorithms
- Ease of Coding
- Development environment
  - Utilities and libraries
  - GUI development support

Target System

• Smartphone IOS	Swift
• Smartphone Android	Java
• Computer Windows	Various
• Computer OSX	Swift
• Unix	C/C++
• Internet/Website	JavaScript
• Raspberry Pi	Python
• BBC Microbit	Python
• Arduino:	C/C++

The Top 6 - #6 C (++ or #)

For Unix, Games

- Use If you are a masochist & want to know what most systems are programmed in
- Can be used with all operating systems
- It is a procedural programming language that forms a good foundation for software development, kernel development, and operating system.
- Occupies a substantial portion of the tech world.
- C++ was developed as an extension to the C language. It has evolved into a multi-model, general-purpose programming language. It is mostly used in Microsoft products and desktop applications. Over the last decade, C++ has grown into one of the most well-known and widely used programming languages.

The Top 6 - #5 Java

Use for Android and most other target platforms https://www.java.com/en/

- Java, another high-level programming language that was developed in the 1990s, is the most popular among modern programmers.
- initially developed for cable boxes and hand-held devices. However, it has
- upgraded so much that today, it is almost everywhere, from the World Wide Web to smartphones to computers.
- Supported on any platform that runs the Java Virtual Machine.
- Java is Open Source, but supported by Oracle
- Extensive documentation and resources available

The Top 6 - #4 Go

Use for network environment https://go.dev

- Go is a procedural and open-source programming language built for the making of simple, reliable software systems.
- Go was designed at Google in 2007 to improve programming productivity in an era of multicore, networked machines and large codebases. Is open source
- The Go language works best for:
  - Cloud-native development
  - Distributed network services
  - Utilities and stand-alone tools
- Go is meant to be simple to learn, straightforward to work with, and easy to read by other developers. Go does not have a large feature set, especially when compared to languages like C++. Go is reminiscent of C in its syntax, making it relatively easy for longtime C developers to learn.

The Top 6 - #3 JavaScript

Use for Web development https://www.javascript.com

- JavaScript is a programming language that is exclusively designed for creating network-centric applications.
- It is an interpreted and lightweight programming language used for scripting the webpages.
- Almost all the websites on the internet today are built on JavaScript.
- Your web browser will run JavaScript programs.
- The major JavaScript releases have added a lot of modern features, and the JavaScript today has vast differences compared to the Javascript of the previous decade.

The Top 6 - #2 Swift

Apple Ecosystem: https://developer.apple.com/swift/

- A general-purpose, compiled programming language that also offers high developer productivity.
- Swift was developed mainly to replace Objective-C in the Mac and iOS platforms.
- Simple, precise, and clean syntax as well as developer ergonomic features, it offers a more productive alternative to Objective-C in the Apple Ecosystem.
- Apple provide excellent development tools and learning environment.

“The powerful programming language that’s also easy to learn”

The Top 6 - #1 Python

All round best - https://www.python.org/

- Python is now one of the most popular programming languages.
- As a general-purpose programming language, Python is designed to be used in many ways. You can build web sites or industrial robots or a game for your friends to play, and much more, all using the same core technology.
- Python Packaging offers huge flexibility to develop programs for different systems.
- It has excellent code readability, vast libraries, and framework. Some of the noteworthy features of Python are:
  - Open-source programming language
  - Extensive support modules and community development
  - Easy integration with web services
  - User-friendly data structures
  - GUI-based desktop applications
- Supported by excellent guides and resources.

Comparison

Comparison of Top Programming Languages

Metrics	Python	JavaScript	Java	C#	C	C++	Go	R	Swift
Typing Discipline	Strong, dynamically typed	Weakly typed	Statically typed	Statically typed	Weakly typed	Weakly typed	Statically typed	Strong but dynamically typed	Statically, strong and inferred type
Platform	Linux, graphical user interface, macOS	Visual studio code, Linux, Windows, Mac	Java SE, Java EE, Java ME, Java FX	MonoDe velop, Rider, Visual studio code	WPerl, Cygwin, Linuxeakly typed	Cross-platform software, Cygwin, Perl	PowerPC, FreeBSD, OpenB SD	Windows, MacOS, Windows	iOS, iPadOS, macOS, tvOS and watchOS
Best For	Data analytics, machine learning, even design	Creation of web pages	Creation of complete dynamic applications	Development of desktop applications, web services and web applications	Scripting of system applications and coding embedded systems	Supports object-oriented programming features	Development of cloud applications, DevOps, command line tools	Supports statistical computing and graphics by R core team	System programming, development of mobile and desktop applications and cloud services
Availability	Writing of python scripts, automating tasks, and conduction of data analysis	Available for making interactive web pages	Designing of web applications that may run on a single computer	Used for flexible memory manage ment	Writing the code for operating systems, muchmore complex programs	Widely used for embedded devices and OS kernels	Used for cloud and server side applications, command line tools	Statistical computing, data miners for developing statistical software and data analysis	Used in a wide range of Apple devices like iOS, iPadOS, macOS, tvOS
Designed By	Guido van Rossum	Brendon Eich	James Gosling	Anders Hejlsberg	Dennis Ritchie	Bjarne Straustrup	Rob Pike, Ken Thomps on	Ross Ihaka	Chris Lattner, John McCall, Doug Gregor
Advantages	Enhanced productivity, easy to learn and write, dynamically typed, vast library support, hassle-free portability	Increased interactivity, richer and enhanced interfaces, less server interaction, great career opportunities	Object-oriented, easy programming language, Supports portability feature, platform independent	Effective memory management, fast and powerful, standard library, object-oriented	Funda mental block for many other programming languages, Portable language, middle-level and structural language, built-infunctions	Mid-level programming language, high portability, fast and powerful, standard library, multi-paradigm	Easy to learn, open sourced, concurrency, static code analysis, fast and hassle-free code implementation	Used for enhanced statistical computing and analysis, supports various data-types, open-source platform, powerful graphics, highly supportive community	Enables a high level of interactivity, fast and modern programming language, much easier to use, easy to locate and correct errors
Disadvantages	Limitations of database, Slower runtime speed, high memory consumption, runtime errors	Browser support Security in the client-side, single inheritance, object-oriented capabilities, lack of debugging facility	Slow and poor performance, no backup facility, provides verbose and complex codes	Run-time checking, test and correct errors, no garbage collection, unsafe, complex	Insufficient memory management, absence of exception handling, Run-time checking, lack of constructor and destructor	No support for garbage pickup, uninsured system security, can cause overload functions	New programming language with notmuch libraries and information, limitedscope, defective dependency management	Used for enhanced statistical computing and analysis, supports various data-types, open-source platform, powerful graphics, highly supportive community	Enables a high level of interactivity, fast and modern programming language, much easier to use, easy to locate and correct errors

Thank You