Introduction to Programming

Canberra Institute of Technology

aie

SPECIALIST EDUCATORS IN
GAMES, ANIMATION & FILM VFX

Contents

- What is programming?
- History of Programming Languages
- A look at the computer
    - Hardware and software
- What is an application?
- Types of Programming Languages

What is Programming?

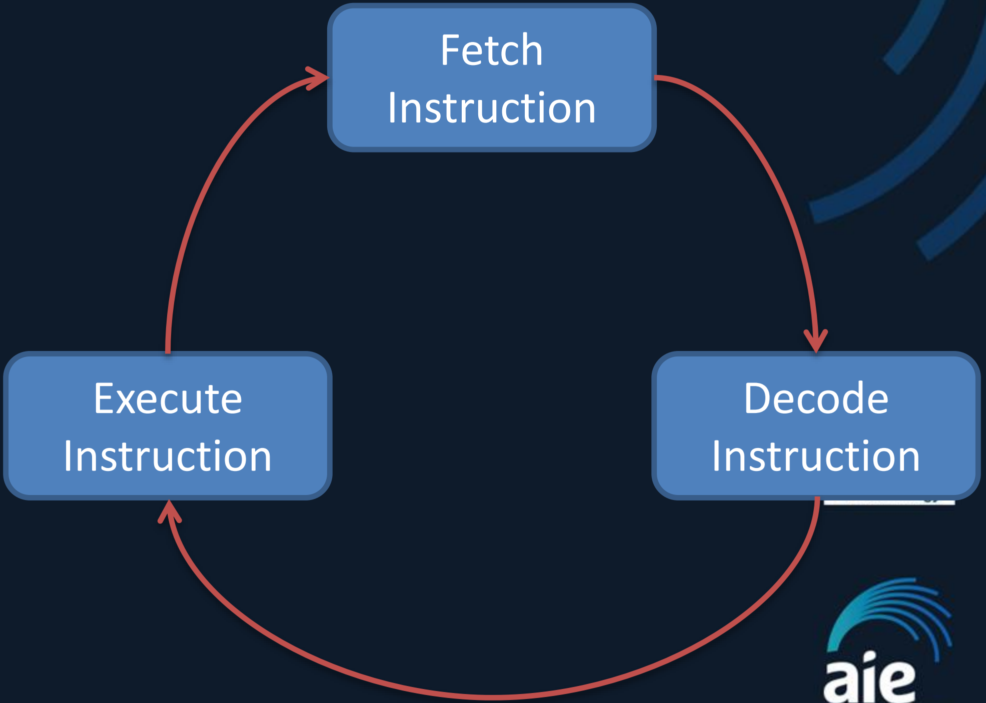
- Programming is telling a computer what to do.

- Computers are really good at following small, simple instructions.

- We, as programmers write out those instructions for the computer to follow.

What is Programming?

- Computers can’t actually do all that much.
- All a computer can ever do is load a big list of instructions, and execute them one after another.
- The kind of instructions a computer can execute are actually very limited.



Fetch Instruction

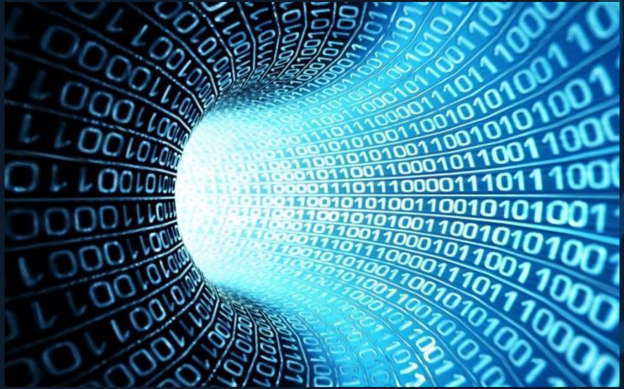
Decode Instruction

Execute Instruction

What is Programming?

- A computer can:
  - Do basic maths (add, subtract, divide, multiply)
  - Save numbers for use later.
  - Change the value of numbers they’ve saved
  - Compare the value of numbers (equal, less than, greater than, not equal)
  - Change where in the big list of instructions they’re up to.
- It might not seem like much, but every program you’ve ever used – no matter how complex is made up of nothing but these kinds of instructions.
- A large part of being a programmer is trying to break down large problems into the small steps that the computer knows how to execute.

What is Programming?



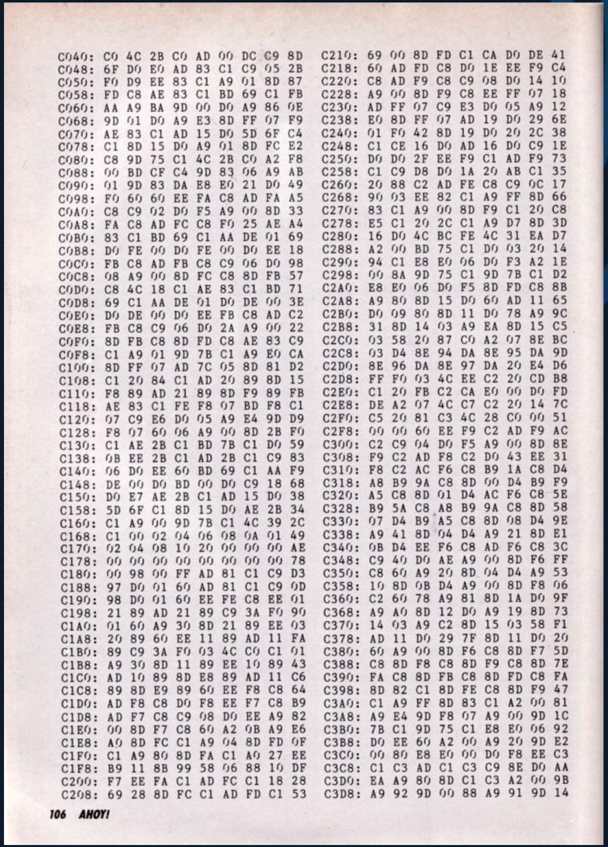
- One of things you might have noticed about the instructions – they all had to do with manipulating numbers.
- One of the most important things to realize about computers is that *everything in a computer is a number*.
- But what about colours? Or sound? Or text? Or video?
  - Anything that doesn't look like a number is a number representing that thing.
- Even the instructions themselves are just numbers.

Canberra Institute of Technology

aie SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

What is Programming?

- Of course, typing out a big list of numbers to tell the computer what to do would be really difficult.
- You would have to remember what each number meant.
- Reading a program someone else wrote would take a long, long, *long* time to understand.
- This is why we have programming languages.



```
C040: C0 4C 2B C0 AD 00 DC C9 8D   C210: 69 00 8D FD C1 CA D0 DE 41
C048: 6F D0 E0 AD 83 C1 C9 05 2B   C218: 60 AD FD C8 D0 1E EE F9 C4
C050: F0 D9 EE 83 C1 A9 01 8D 87   C220: C8 AD F9 C8 C9 08 D0 14 10
C058: FD C8 AE 83 C1 BD 69 C1 FB   C228: A9 00 8D F9 C8 EE FF 07 18
C060: AA A9 BA 9D 00 D0 A9 86 0E   C230: AD FF 07 C9 E3 D0 05 A9 12
C068: 9D 01 D0 A9 E3 8D FF 07 F9   C238: E0 8D FF 07 AD 19 D0 29 6E
C070: AE 83 C1 AD 15 D0 5D 6F C4   C240: 01 F0 42 8D 19 D0 20 2C 38
C078: C1 8D 15 D0 A9 01 8D FC E2   C248: C1 CE 16 D0 AD 16 D0 C9 1E
C080: C8 9D 75 C1 4C 2B C0 A2 F8   C250: D0 D0 2F EE F9 C1 AD F9 73
C088: 00 BD CF C4 9D 83 06 A9 AB   C258: C1 C9 D8 D0 1A 20 AB C1 35
C090: 01 9D 83 DA E8 E0 21 D0 49   C260: 20 88 C2 AD FE C8 C9 0C 17
C098: F0 60 60 EE FA C8 AD FA A5   C268: 90 03 EE 82 C1 A9 FF 8D 66
C0A0: C8 C9 02 D0 F5 A9 00 8D 33   C270: 83 C1 A9 00 8D F9 C1 20 C8
C0A8: FA C8 AD FC C8 F0 25 AE A4   C278: E5 C1 20 2C C1 A9 D7 8D 3D
C0B0: 83 C1 BD 69 C1 AA DE 01 69   C280: 16 D0 4C BC FE 4C 31 EA D7
C0B8: D0 FE 00 D0 FE 00 D0 EE 18   C288: A2 00 BD 75 C1 D0 03 20 14
C0C0: FB C8 AD FB C8 C9 06 D0 98   C290: 94 C1 E8 E0 06 D0 F3 A2 1E
C0C8: 08 A9 00 8D FC C8 8D FB 57   C298: 00 8A 9D 75 C1 9D 7B C1 D2
C0D0: C8 4C 18 C1 AE 83 C1 BD 71   C2A0: E8 E0 06 D0 F5 8D FD C8 8B
C0D8: 69 C1 AA DE 01 D0 DE 00 3E   C2A8: A9 80 8D 15 D0 60 AD 11 65
C0E0: D0 DE 00 D0 EE FB C8 AD C2   C2B0: D0 09 80 8D 11 D0 78 A9 9C
C0E8: FB C8 C9 06 D0 2A A9 00 22   C2B8: 31 8D 14 03 A9 EA 8D 15 C5
C0F0: 8D FB C8 8D FD C8 AE 83 C9   C2C0: 03 58 20 87 C0 A2 07 8E BC
C0F8: C1 A9 01 9D 7B C1 A9 E0 CA   C2C8: 03 D4 8E 94 DA 8E 95 DA 9D
C100: 8D FF 07 AD 7C 05 8D 81 D2   C2D0: 8E 96 DA 8E 97 DA 20 E4 D6
C108: C1 20 84 C1 AD 20 89 8D 15   C2D8: FF F0 03 4C EE C2 20 CD B8
C110: F8 89 AD 21 89 8D F9 89 FB   C2E0: C1 20 FB C2 CA E0 00 D0 FD
C118: AE 83 C1 FE F8 07 BD F8 C1   C2E8: DE A2 07 4C C7 C2 20 14 7C
C120: 07 C9 E6 D0 05 A9 E4 9D D9   C2F0: C5 20 81 C3 4C 28 C0 00 51
C128: F8 07 60 06 A9 00 8D 2B F0   C2F8: 00 00 60 EE F9 C2 AD F9 AC
C130: C1 AE 2B C1 BD 7B C1 D0 59   C300: C2 C9 04 D0 F5 A9 00 8D 8E
C138: 0B EE 2B C1 AD 2B C1 C9 83   C308: F9 C2 AD F8 C2 D0 43 EE 31
C140: 06 D0 EE 60 BD 69 C1 AA F9   C310: F8 C2 AC F6 C8 B9 1A C8 D4
C148: DE 00 D0 BD 00 D0 C9 18 68   C318: A8 B9 9A C8 8D 00 D4 B9 F9
C150: D0 E7 AE 2B C1 AD 15 D0 38   C320: A5 C8 8D 01 D4 AC F6 C8 5E
C158: 5D 6F C1 8D 15 D0 AE 2B 34   C328: B9 5A C8 A8 B9 9A C8 8D 58
C160: C1 A9 00 9D 7B C1 4C 39 2C   C330: 07 D4 B9 A5 C8 8D 08 D4 9E
C168: C1 00 02 04 06 08 0A 01 49   C338: A9 41 8D 04 D4 A9 21 8D E1
C170: 02 04 08 10 20 00 00 00 AE   C340: 0B D4 EE F6 C8 AD F6 C8 3C
C178: 00 00 00 00 00 00 00 00 78   C348: C9 40 D0 AE A9 00 8D F6 FF
C180: 00 98 00 FF AD 81 C1 C9 D3   C350: C8 60 A9 20 8D 04 D4 A9 53
C188: 97 D0 01 60 AD 81 C1 C9 0D   C358: 10 8D 0B D4 A9 00 8D F8 06
C190: 98 D0 01 60 EE FE C8 EE 01   C360: C2 60 78 A9 81 8D 1A D0 9F
C198: 21 89 AD 21 89 C9 3A F0 90   C368: A9 A0 8D 12 D0 A9 19 8D 73
C1A0: 01 60 A9 30 8D 21 89 EE 03   C370: 14 03 A9 C2 8D 15 03 58 F1
C1A8: 20 89 60 EE 11 89 AD 11 FA   C378: AD 11 D0 29 7F 8D 11 D0 20
C1B0: 89 C9 3A F0 03 4C C0 C1 01   C380: 60 A9 00 8D F6 C8 8D F7 5D
C1B8: A9 30 8D 11 89 EE 10 89 43   C388: C8 8D F8 C8 8D F9 C8 8D 7E
C1C0: AD 10 89 8D E8 89 AD 11 C6   C390: FA C8 8D FB C8 8D FD C8 FA
C1C8: 89 8D E9 89 60 EE F8 C8 64   C398: 8D 82 C1 8D FE C8 8D F9 47
C1D0: AD F8 C8 D0 F8 EE F7 C8 B9   C3A0: C1 A9 FF 8D 83 C1 A2 00 81
C1D8: AD F7 C8 C9 08 D0 EE A9 82   C3A8: A9 E4 9D F8 07 A9 00 9D 1C
C1E0: 00 8D F7 C8 60 A2 0B A9 E6   C3B0: 7B C1 9D 75 C1 E8 E0 06 92
C1E8: A0 8D FC C1 A9 04 8D FD 0F   C3B8: D0 EE 60 A2 00 A9 20 9D E2
C1F0: C1 A9 80 8D FA C1 A0 27 EE   C3C0: 00 80 E8 E0 00 D0 F8 EE C3
C1F8: B9 11 8B 99 58 06 88 10 DF   C3C8: C1 C3 AD C1 C3 C9 8E D0 AA
C200: F7 EE FA C1 AD FC C1 18 28   C3D0: EA A9 80 8D C1 C3 A2 00 9B
C208: 69 28 8D FC C1 AD FD C1 53   C3D8: A9 92 9D 00 88 A9 91 9D 14
```

106 AHOY!

What is a Programming Language

- A programming language is a *formal language* that lets you write instructions for the computer.
- The purpose of a programming language is to make it easier to tell the computer what to do.
- You type instructions in the programming language, then another program converts the text you type into instructions that the computer can execute.

What is a Programming Language

- Programming languages are *formal languages*.
- This differentiates them from informal languages like English.
- Being a formal language means that the rules of the language – its syntax - are strict and unambiguous.

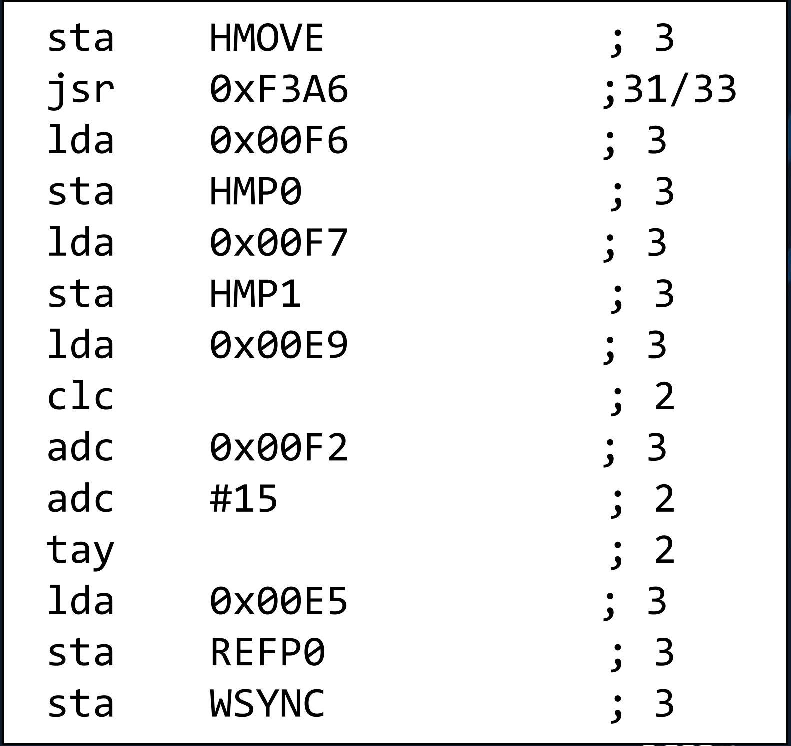
Canberra Institute of Technology

aie

SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

The History of Programming Languages

- The first languages were assembly languages.
- These languages just gave simple names to the instructions the CPU could perform.
- While significantly better than raw numbers, they were still difficult to read and error-prone.



```
sta     HMOVE                  ; 3
jsr     0xF3A6                 ;31/33
lda     0x00F6                 ; 3
sta     HMP0                   ; 3
lda     0x00F7                 ; 3
sta     HMP1                   ; 3
lda     0x00E9                 ; 3
clc                            ; 2
adc     0x00F2                 ; 3
adc     #15                    ; 2
tay                            ; 2
lda     0x00E5                 ; 3
sta     REFP0                  ; 3
sta     WSYNC                  ; 3
```

The History of Programming Languages

- Very quickly, people figured out that better languages would make writing programs a lot easier.

- 1956 – FORTRAN, the first high level language was created. Short for Formula Translation, it was used for science computations.

The History of Programming Languages

- Once people figured out how to make languages, people made more, and more.
  - Lisp – 1958
  - COLBOL – 1959
  - Simula – 1964
  - BASIC – 1964
  - Smalltalk – 1969
  - B – 1969
  - C – 1978
  - C++ - 1983
  - Perl – 1987
  - Python – 1991
  - Ruby – 1993
  - Java – 1995
  - Actionscript – 2000
  - C# - 2003
  - Golang – 2009
  - Rust – 2012

The History of Programming Languages

Many languages tried to improve on previous languages.

- Others tried to just be really good at solving specific kinds of problems.
- Many were even made as jokes or parodies of other languages.
(called esoteric languages)

The list on the previous slide is only a tiny fraction of the programming languages that have been created over the years.

With all these different languages, we can split them up into some broad categories

Low Level vs High Level

- How high or low level a language is, is how directly it maps to machine instructions.

- Assembly is the lowest level language.
  - Each line is an exact specific machine instruction.

- C/C++ is a medium level language.
  - A lot of the time you don’t need to think about machine code when writing C++, but its close enough that if you need to, you can.

Low Level vs High Level

Languages like Python, Lua and Javascript are high level languages.

- These languages are usually Interpreted and have a higher level of abstraction from assembly.

High level languages are usually easier to write programs with and run slower than lower level languages such as C++

Compiled vs Interpreted

- One of the other big differences between languages is how the code gets transformed from the text you write to the machine code that executes.

- There are two main ways to do this.
  - Compilation
  - Interpretation

Compilation

- For compilation the process goes like so:
  - Source code is given to a *compiler* as input.
  - The compiler reads all the source code.
  - The compiler converts the source code to machine code.
  - The compiler outputs a file made of all the machine code.
  - Once the compiler has finished, the machine code can now be executed.
  - The machine code can be copied to other computers that use the same machine code and executed on them.

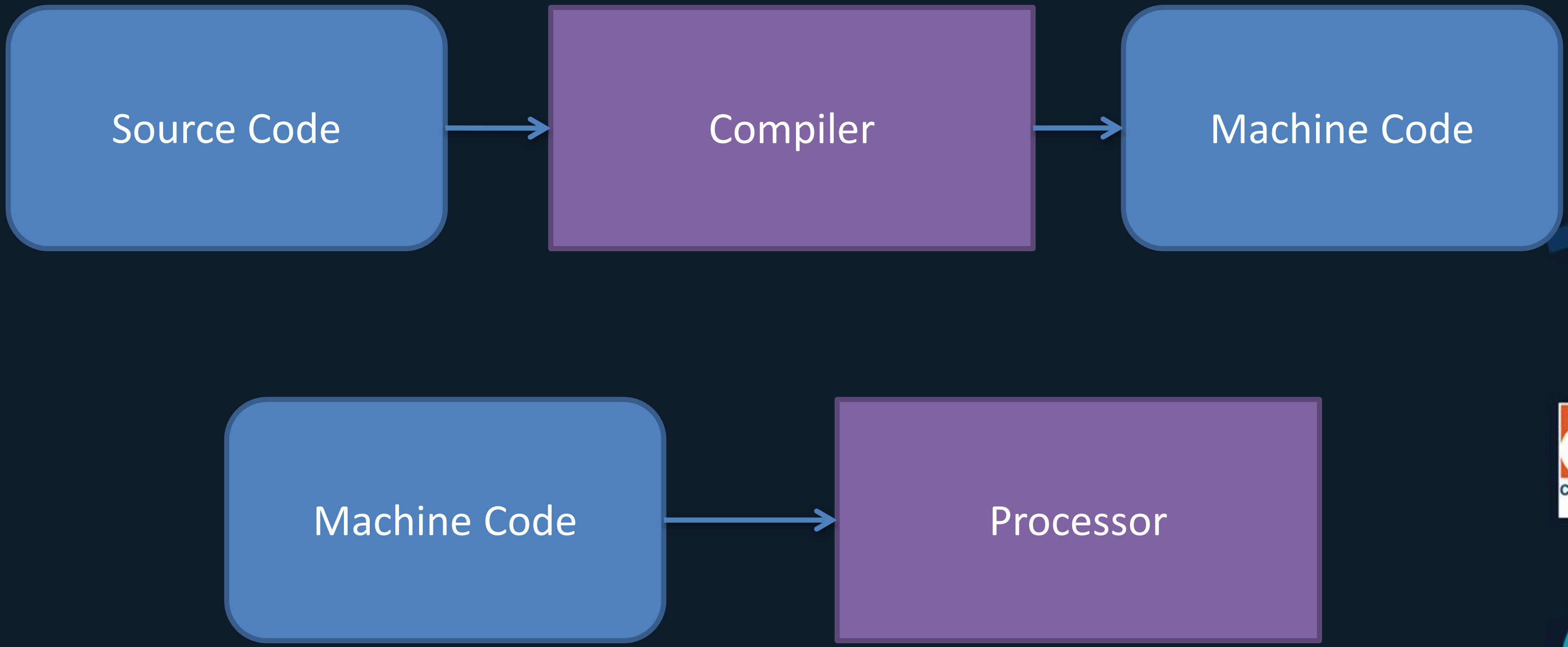
- C, C++, Go, and Rust are examples of compiled languages.

Compilation

- For compilation the process goes like so:
  - Source code is given to a *compiler* as input.
  - The compiler converts the source code into machine readable instructions
  - The final executable is produced ready for direct execution.
  - On windows this is the PE (Portable Executable) File format (*.exe)

- C, C++, Go, and Rust are examples of compiled languages.

Compilation



Source Code → Compiler → Machine Code

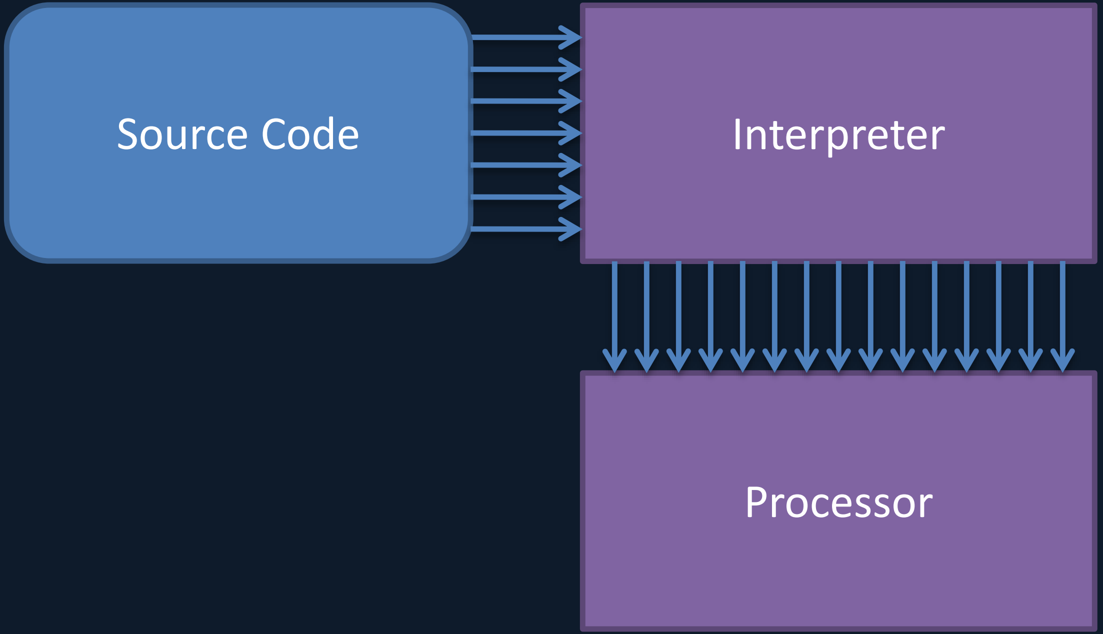
Machine Code → Processor

Interpretation

- For interpretation the process goes like so:
  - Source code is given to an interpreter as input.
  - The interpreter reads in small chunks of the source code.
  - As it reads each chunk, the interpreter will generate the correct machine code for immediate execution
  - The source code can be distributed to any computer that has a valid interpreter installed.

- JavaScript, Python, Lua, Ruby, and PHP are examples of interpreted languages.

Interpretation



Source Code

Interpreter

Processor

Hardware and Software

- The code you will write runs on top of physical hardware
  - The different parts of the computer work with your code in different ways
- The CPU is the brain of the computer. It is the part that actually executes the instructions
- RAM is the memory of the computer. All of the numbers a program stores and modifies, including the instructions themselves are stored in RAM.
- RAM is a limited resource, as soon as one program finishes, a different one will likely overwrite all the original program's data with its own. We use the hard drive to save out more permanent files that can be loaded later.

Summary

- Programming is the act of writing instructions for a computer.
- Instructions are stored in the computer as numbers called machine code.
- Programming languages make in easier for us to write the instructions for the computer in a way we can understand.
- Code can be compiled or interpreted.
- There are lots and lots of different programming languages.

Canberra Institute of Technology

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX