



Components of a **Machine** And how it works



|| Mr. Robo: Machine

Mr. Robo is a Machine. Mr. Robo can be given instructions to perform many tasks.



Mr. Robo: Machine

These are the possible **words** that Mr. Robo understands.



Instructions	
Action Code	
Move	0 0 0 1
Charge	0 1 0 0
Direction Code	
Left	0 0
Right	0 1
Up	1 0
Down	1 1
Step Code	
Zero Step	0 0 0
One Step	0 0 1
Two Step	0 1 0
Three Step	0 1 1

Mr. Robo: Operation Code

Mr. Robo understands only a predefined set of words. These set of words are called **Operation Codes**.



Instructions	
Action Code	
Move	0 0 0 1
Charge	0 1 0 0
Direction Code	
Left	0 0
Right	0 1
Up	1 0
Down	1 1
Step Code	
Zero Step	0 0 0
One Step	0 0 1
Two Step	0 1 0
Three Step	0 1 1

Mr. Robo: Instruction

We combine these operation code to make a complete **Instruction**. For example,

0001 **01** **001**

is a complete instruction that move Robo one step right.



Instructions	
Action Code	
Move	0 0 0 1
Charge	0 1 0 0
Direction Code	
Left	0 0
Right	0 1
Up	1 0
Down	1 1
Step Code	
Zero Step	0 0 0
One Step	0 0 1
Two Step	0 1 0
Three Step	0 1 1

Mr. Robo: Instruction

Mr. Robo only understands the **Instruction** when you give in the specific order.

Action Code	Direction Code	Step Code
-------------	----------------	-----------



Instructions	
Action Code	
Move	0 0 0 1
Charge	0 1 0 0
Direction Code	
Left	0 0
Right	0 1
Up	1 0
Down	1 1
Step Code	
Zero Step	0 0 0
One Step	0 0 1
Two Step	0 1 0
Three Step	0 1 1

Mr. Robo: Input

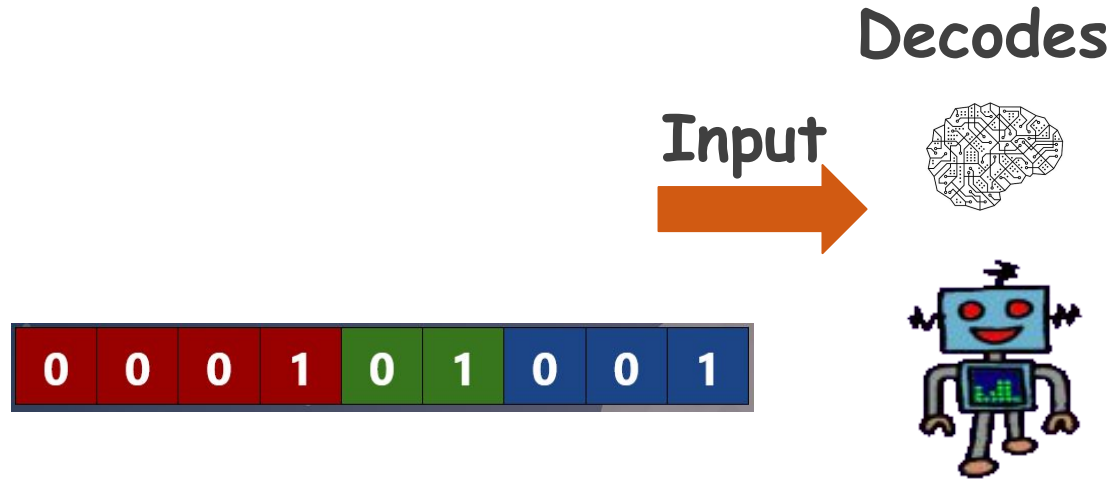
Mr. Robo takes these instructions as 0 or 1 through its tray. The tray is called **Input Device**.



Instructions	
Action Code	
Move	0 0 0 1
Charge	0 1 0 0
Direction Code	
Left	0 0
Right	0 1
Up	1 0
Down	1 1
Step Code	
Zero Step	0 0 0
One Step	0 0 1
Two Step	0 1 0
Three Step	0 1 1

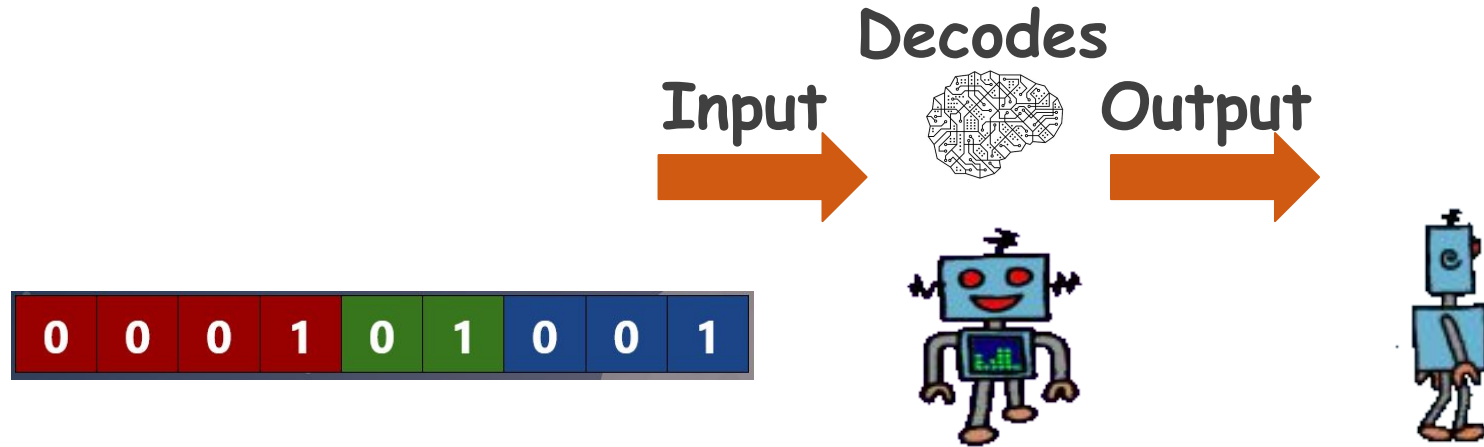
Mr. Robo: CPU (Brain)

The Brain (**Central Processing Unit**) of Mr. Robo **Decodes** the instruction.



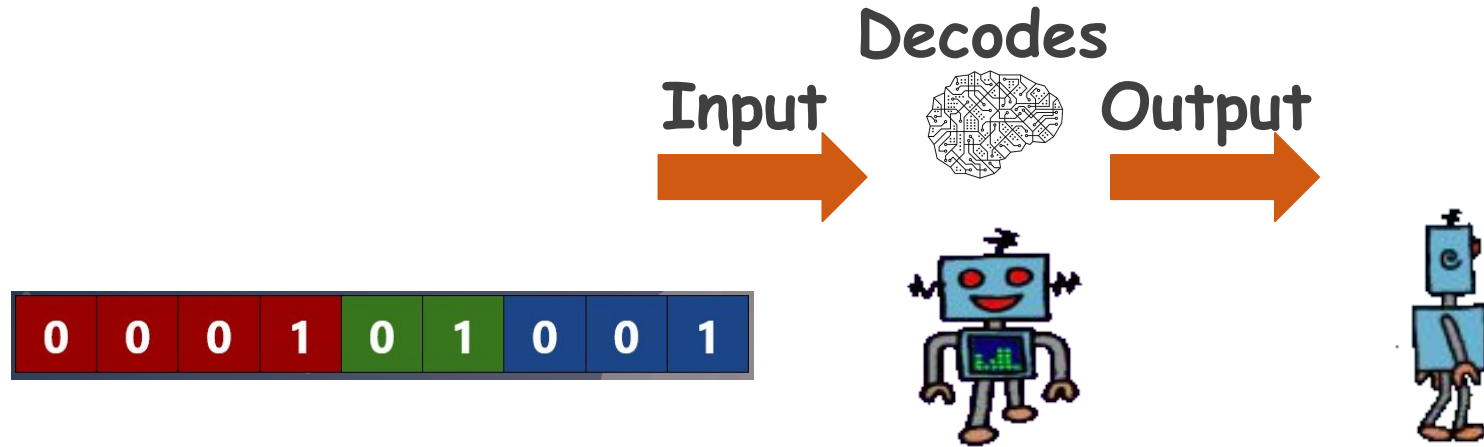
Mr. Robo: Computation Step

One instruction is called one **Computation Step** because this is a unit task that CPU (Brain) can perform.



Mr. Robo: Instruction Cycle

In order to process the information, CPU (brain) first **fetches** the instruction, **decodes** it and then **executes** it to give output. This is called **Instruction Cycle**.



Mr. Robo: Why Understand 0 and 1?

Why Mr. Robo understands only 0 and 1?



Mr. Robo: Why Understand 0 and 1?

Because Mr. Robo is an **Electric Machine** that can understand **0** (no or low electricity in wires) and **1** (electricity or high electricity in wires) naturally.



Mr. Robo: Binary Language

Thus, Mr. Robo understands a language that has only two alphabets (0 and 1). This language is called **Binary Language**.



Mr. Robo: Machine Language

So, binary is the **Mother Tongue** or **Natural Language** of the Mr. Robo. Technically, it is also called **Machine Language**.



Mr. Robo: Binary Language

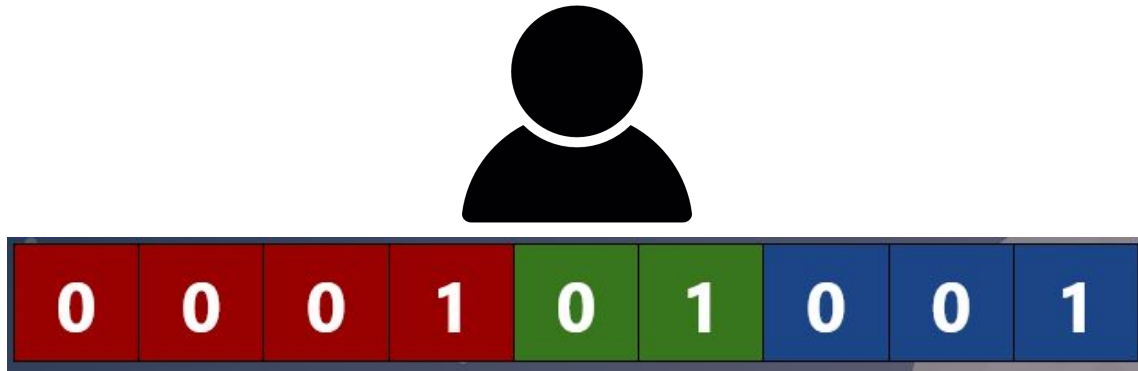
Why **Binary**? Why not any other language? Binary is so difficult for Us to Remember.



Mr. Robo: Binary Language

What if there is someone, who understands binary language and we tell him the instruction in **English** and he converts that into **Binary**.

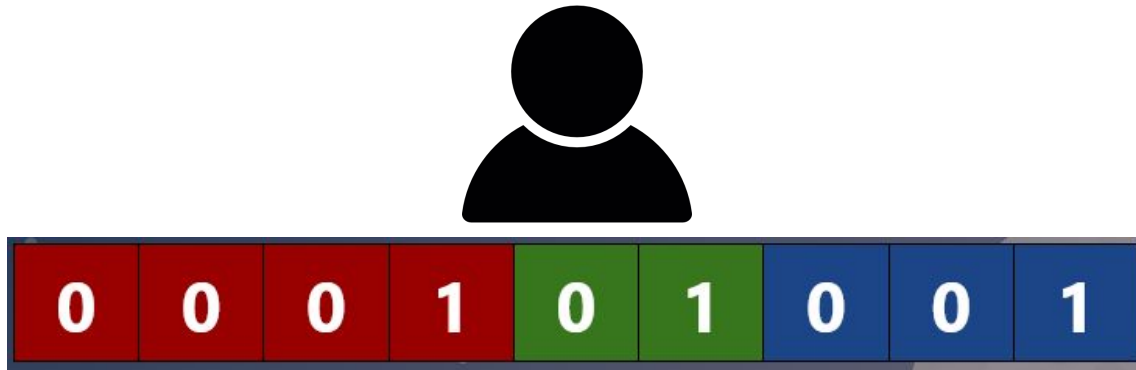
Move **Right** **One** **Step**



Mr. Robo: Compiler

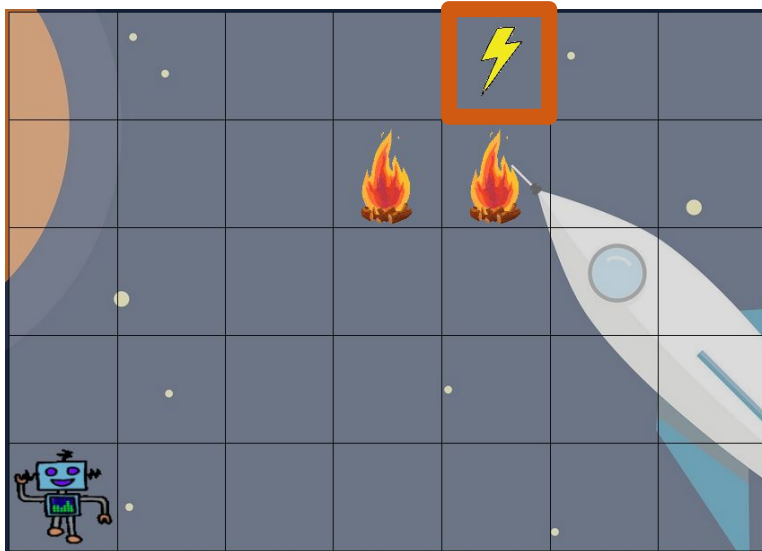
Such translator is called **Compiler**.

Move Right One Step



Mr. Robo: How to move?

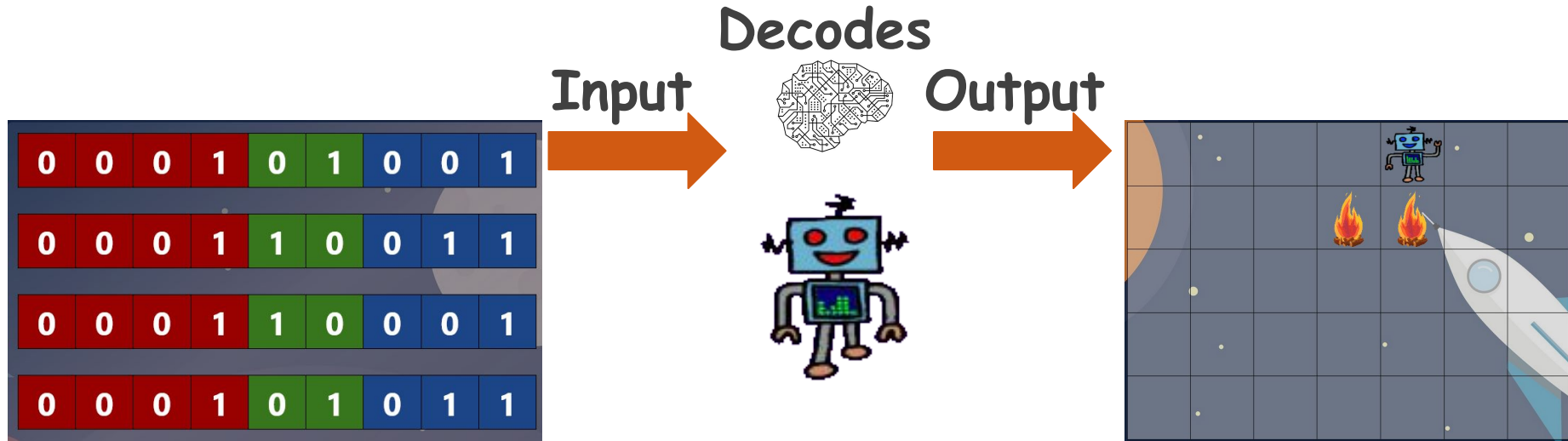
We need Mr. Robo to go to Charge location with these available Operation Codes. What do we do?



Instructions	
Action Code	
Move	0 0 0 1
Charge	0 1 0 0
Direction Code	
Left	0 0
Right	0 1
Up	1 0
Down	1 1
Step Code	
Zero Step	0 0 0
One Step	0 0 1
Two Step	0 1 0
Three Step	0 1 1

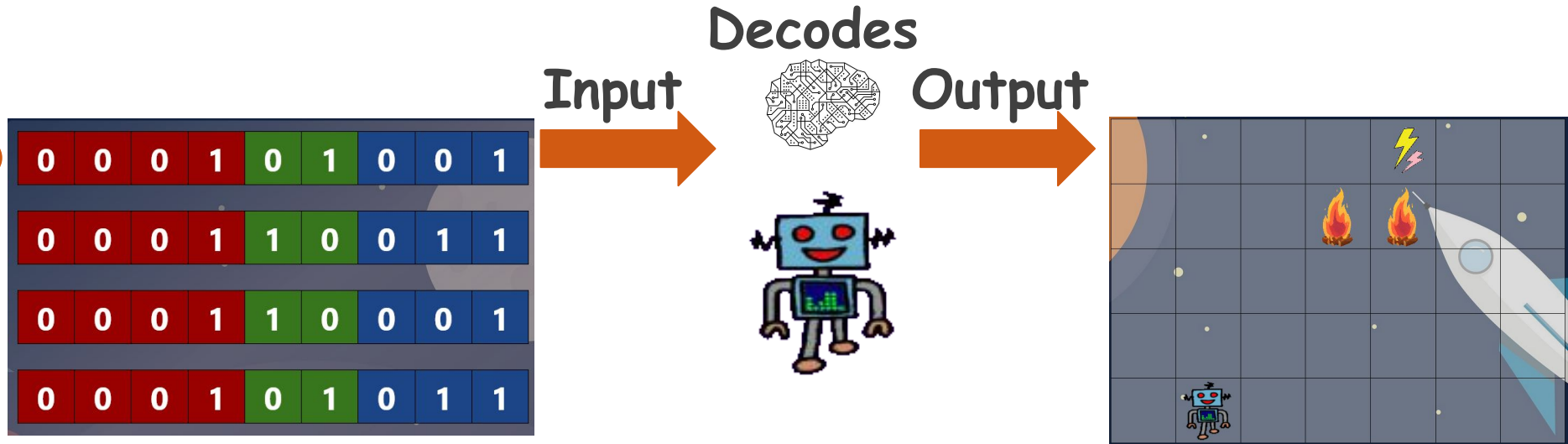
Mr. Robo: Multiple Computation Steps

If we need Mr. Robo to move at some location for that we do not have single instruction, we need to instruct in terms of **Multiple Computation Steps**.



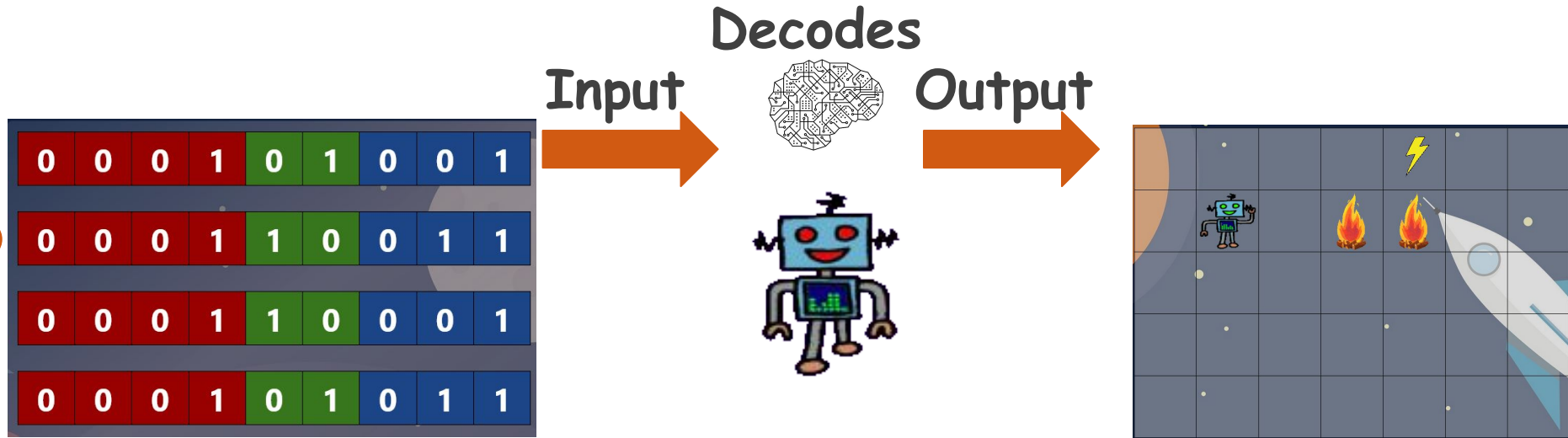
Mr. Robo: After 1st Instruction

These set of Instructions are executed by brain (CPU) one by one in the given sequence.



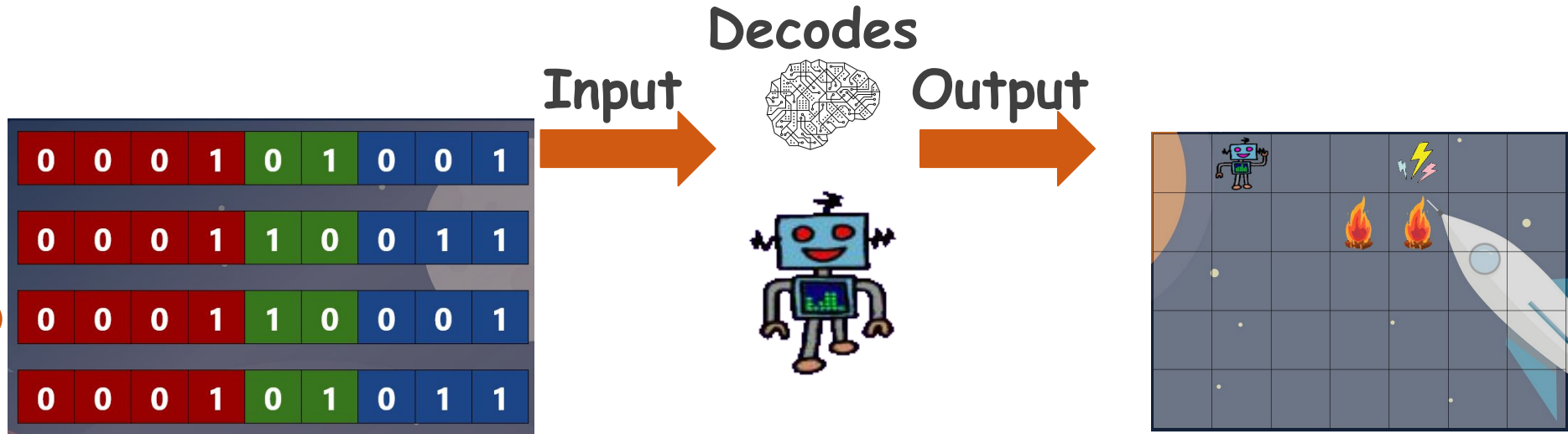
Mr. Robo: After 2nd Instruction

These set of Instructions are executed by brain (CPU) one by one in the given sequence.



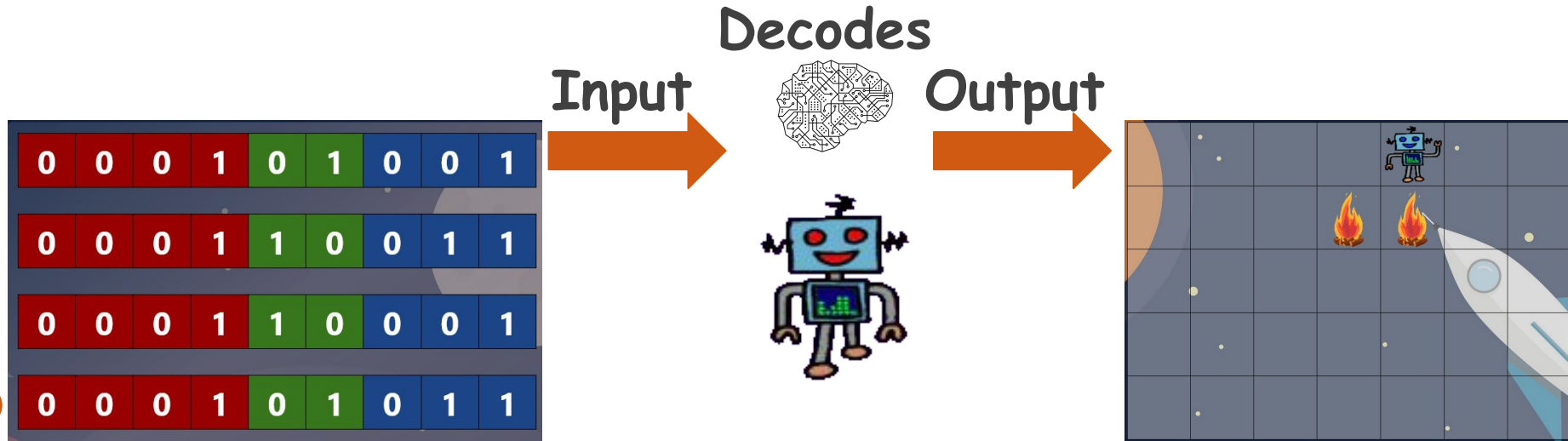
Mr. Robo: After 3rd Instruction

These set of Instructions are executed by brain (CPU) one by one in the given sequence.



Mr. Robo: After 4th Instruction

These set of Instructions are executed by brain (CPU) one by one in the given sequence.



|| Mr. Robo: Why we are Studying this?

You Should ask why we are studying Mr. Robo? What it has to do with Computers and Programming?
i.e.

Your subject: Programming Fundamentals

Computer: Similarity with Mr. Robo?

Computer is also an **Electronic Machine** which means it understands **Binary Language**. It takes input, processes on it and gives output.



Computer: Similarity with Mr. Robo?

Computer takes **Input**, processes on it and gives **Output**.



Input

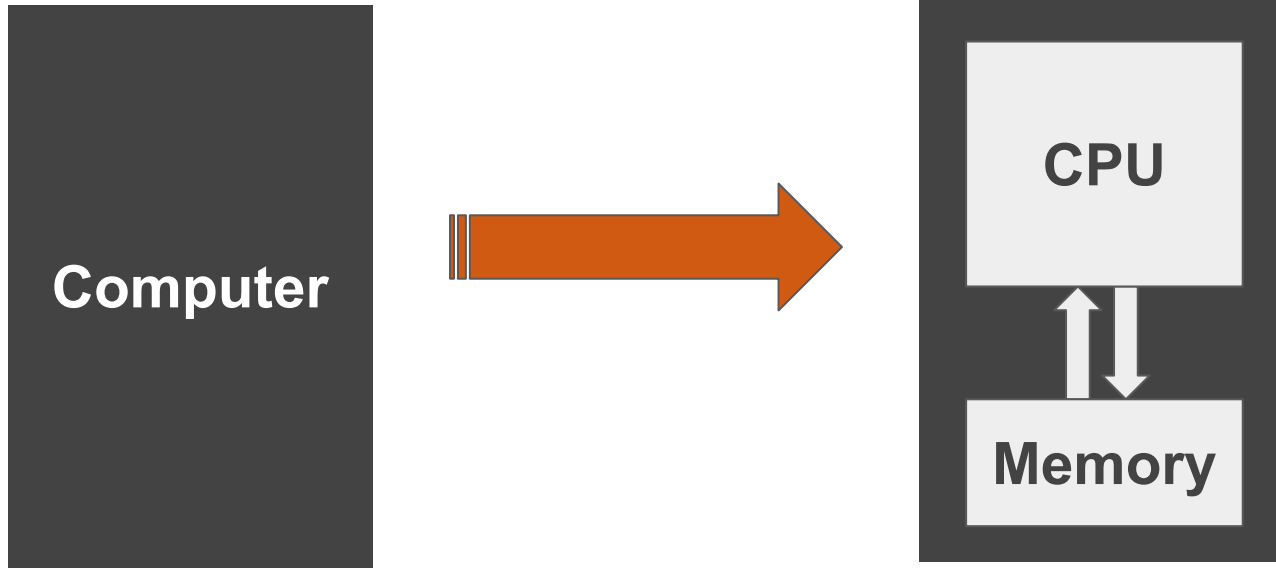


CPU



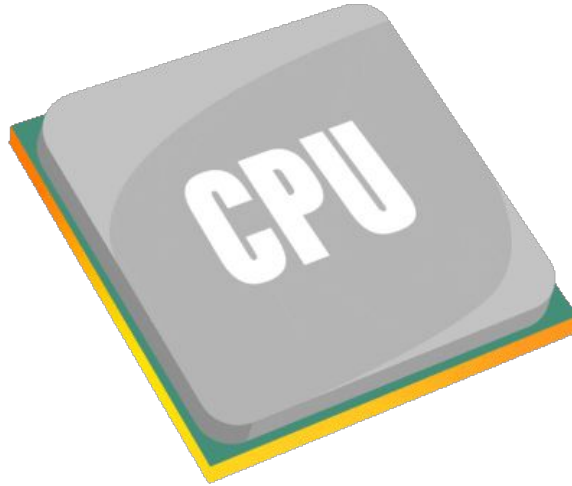
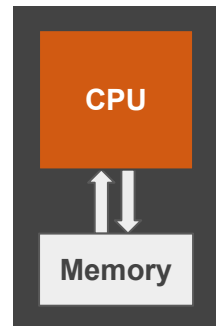
Output

What is inside the Computer?



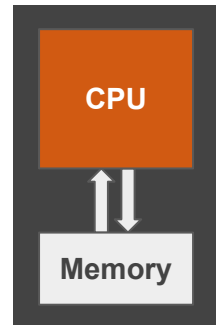
CPU: Brain of the Computer

- CPU is the **main** processing unit
- It has **predefined** set of instructions



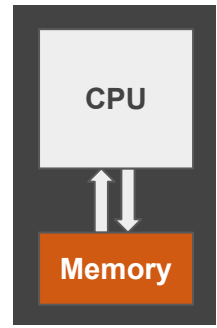
CPU Operations

- Some of these **operations** include
 1. Addition (0010)
 2. Multiplication (0011)
 3. Take Input (1100)
 4. Give Output (0110)
 5. Store Data (1110)
 6. Load Data (0111)



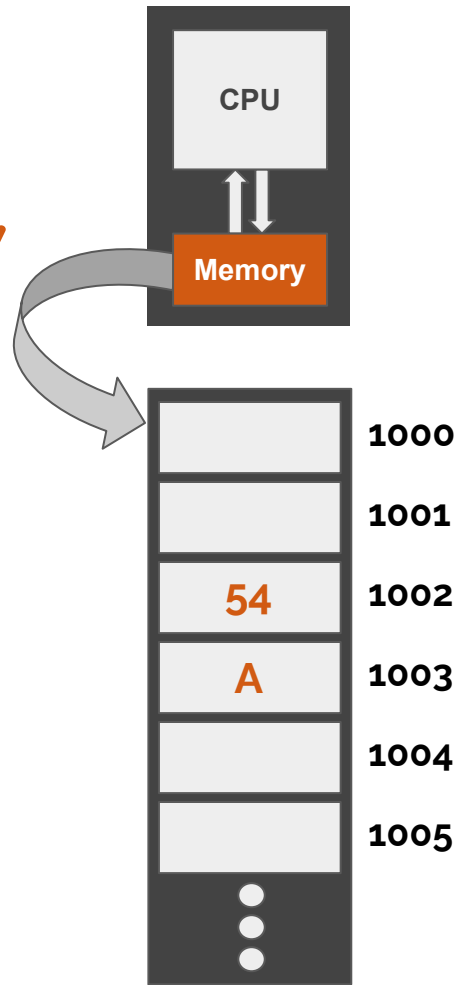
Memory

- When CPU takes input from devices, it stores information into **memory** before processing it.
- CPU stores intermediate results of the processing into the **memory**.
- CPU stores information into the **memory** before sending it to output devices.



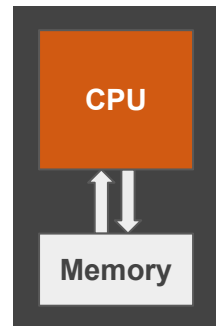
Main Memory

- Memory is called **Main Memory**, **Primary Memory** or **RAM**.
- This memory is divided into **different cells**.
- Each cell has an **address** like we have address of our house numbers or PO Boxes
- CPU **stores** data into these cells and **loads** data from these cells whenever it is required.

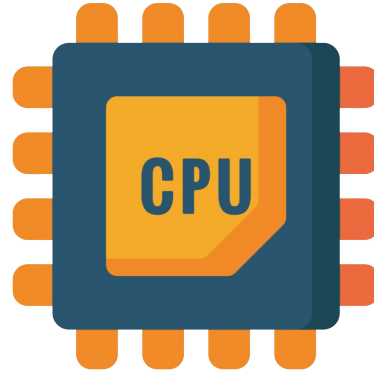
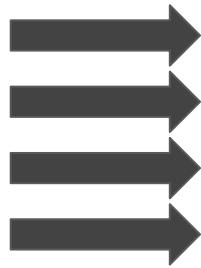


CPU: Operation Code

- If we need CPU to perform some **operation** then we send it a **signal**.
- This signal is called **operation code**.
- All CPU operations have **unique** operational code.

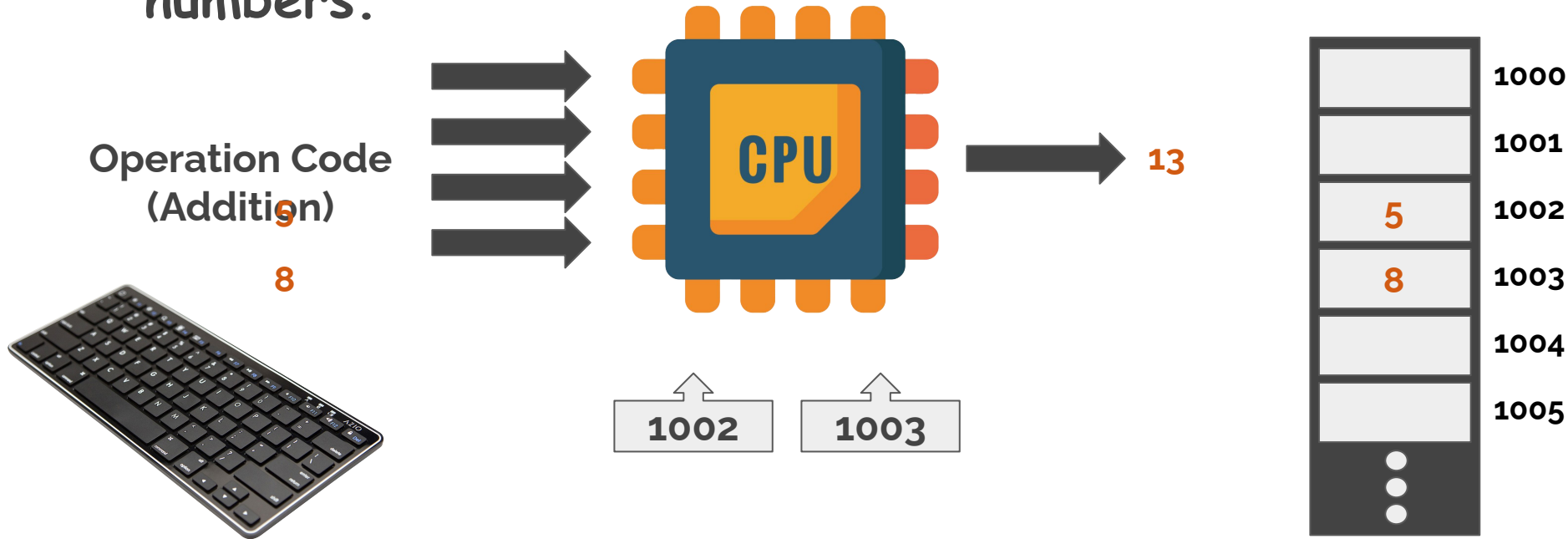


Operation Code
Addition
Multiplication
Store
Load



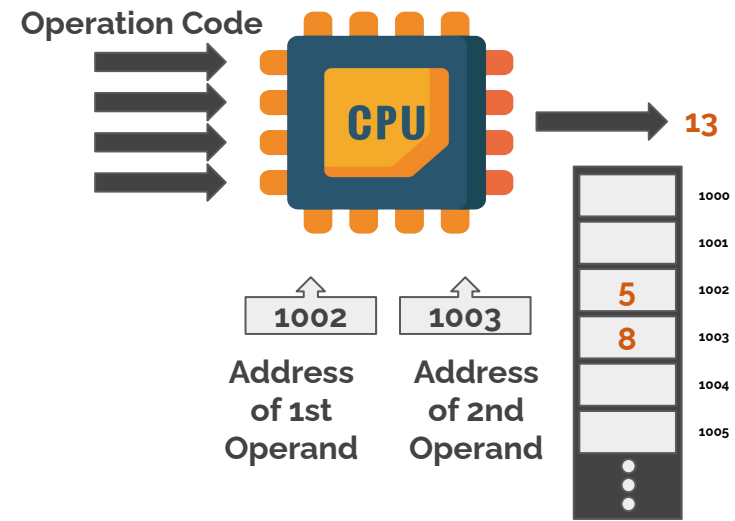
How CPU Works

- Let see how to **instruct** CPU to add two numbers.



Instruction Code

An **Instruction** consists of **Operation Code** and **Addresses of the Operands** on which the Operation has to be performed.



Operation
Code

Address of 1st
Operand

Address of 2nd
Operand

Op Code

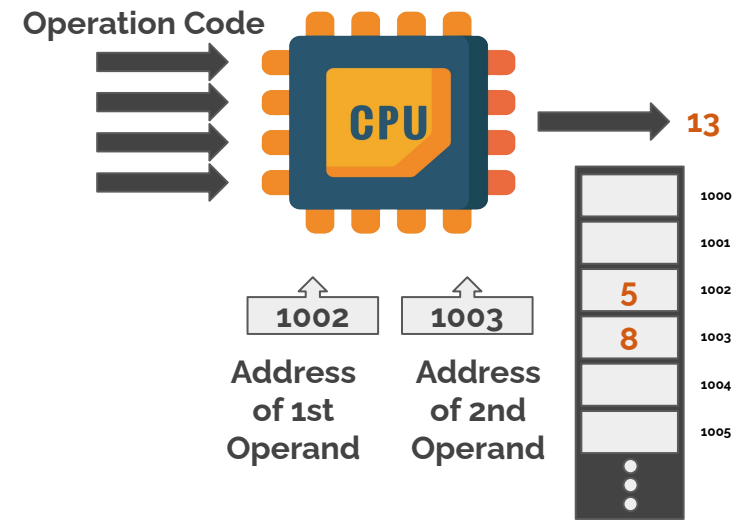
Operand Reference

Operand Reference



Computational Step

- One Instruction Code is called one **Computational Step**.
- It is a step that CPU takes in **Single Unit Time**.



Operation
Code

Address of 1st
Operand

Address of 2nd
Operand

Op Code

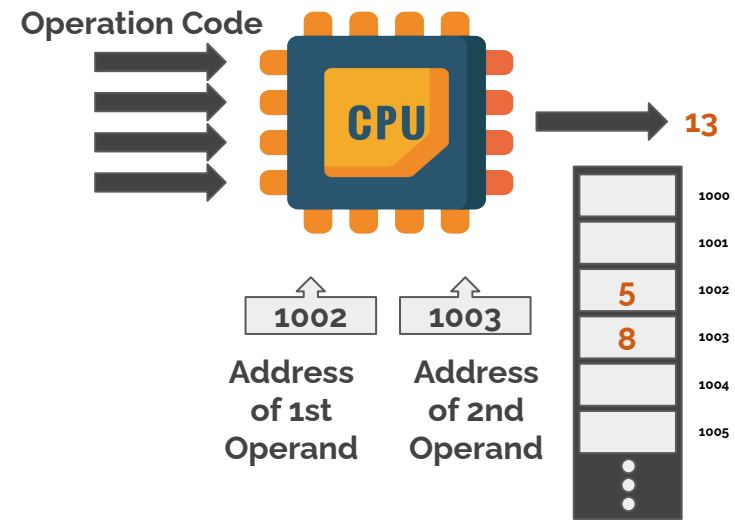
Operand Reference

Operand Reference



Program

To do a task we need to write Multiple **Instructions**. These multiple instructions are collectively called a **Program**.



Operation
Code

Address of 1st
Operand

Address of 2nd
Operand

Op Code

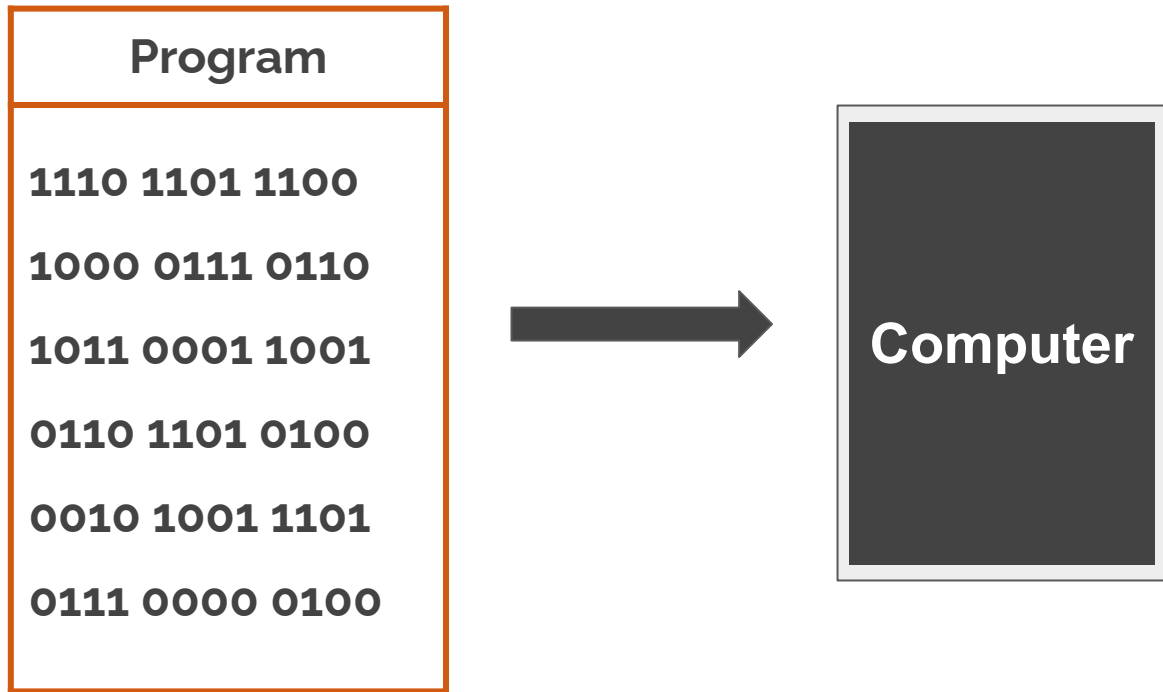
Operand Reference

Operand Reference



Program: Similarity with Mr. Robo?

It's difficult to write Programs in Machine Language.



Program: Similarity with Mr. Robo?

Therefore, scientists have made **High Level Languages** to instruct Computers which are close to English language.

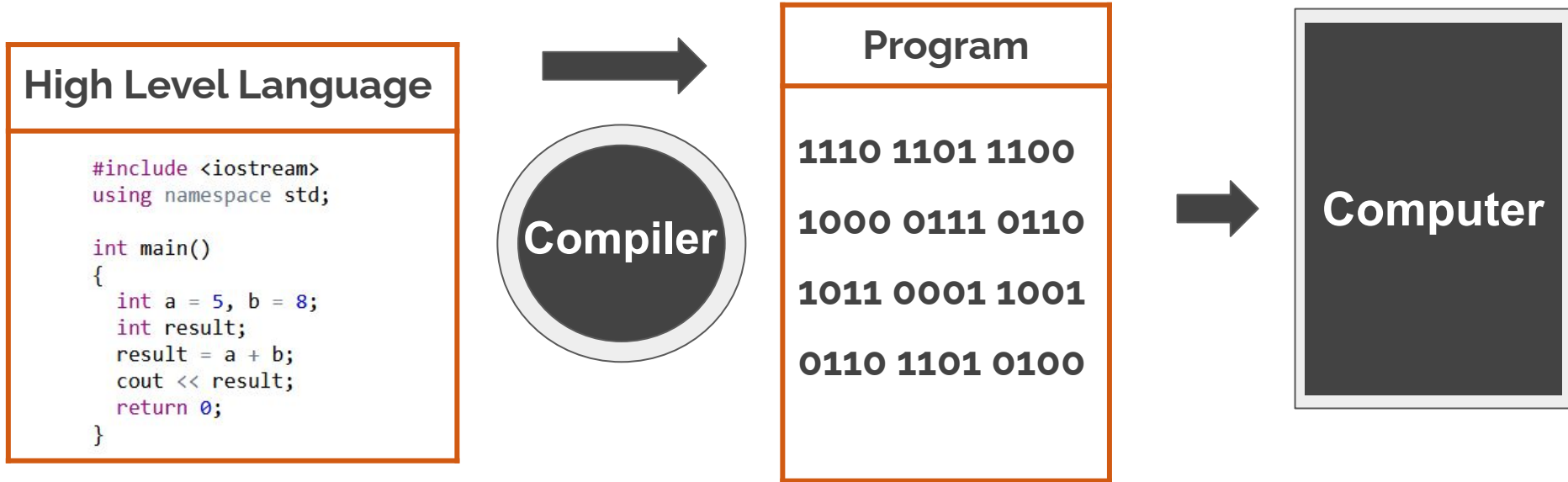
High Level Language

```
#include <iostream>
using namespace std;

int main()
{
    int a = 5, b = 8;
    int result;
    result = a + b;
    cout << result;
    return 0;
}
```

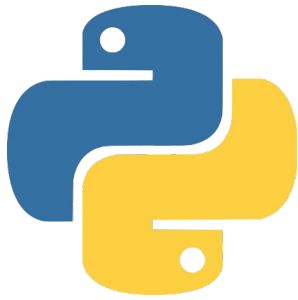
Compiler: Similarity with Mr. Robo?

Compilers convert the **High Level Language** into **Machine Language**.



Language: Similarity with Mr. Robo?

There are many High Level Languages.



Python



C++



Java



C#

Language: Similarity with Mr. Robo?

We will work in C++.



C++

Learning Objective

Define the **Major Components** of the **Computer** and how it works.



Conclusion

- Computer.
- Parts of Computer.
- Input Devices.
- CPU.
- Memory.
- Output Devices.
- Binary Language.
- Machine Language.
- High Level languages.
- Role of Compiler.
- Operation Code.
- Instructions.
- Computation Step.
- Instruction Cycle.
- Multiple Computation Steps
- Program (Set of Sequenced Instructions).

Self Assessment

- What is the **Difference** between single computational step and multiple computational step?
- What is **Machine** Language?
- Why computer use the **Binary Language**?
- What is the Role of **Compiler**?
- In which language, it is easy for **Programmers** to write their **Programs**? Binary or High Level Language?



Self Assessment

- **Identify** the operations and operands from the given

$$2 * 4 + 8$$

- **Store** the above data in memory on any location?

