# CS Summer Challenge

## Day 0x2

How does a code work?

# What is a Programming Language?

A language **that can be translated** to a language that the digital machine can understand.

***A language that can be translated to binary code.***

**e.g.  C, Python**

# C Demo

How does code work?

# The First PL

## Assembly Language

A set of basic operations that translate directly to runnable binary encodings.

## Structure of an Assembly Operation

<operator> <*operand_1*> <*operand_2*>

```
add    <arg_1>, <arg_2>
subt   <arg_1>, <arg_2>
and    <arg_1>, <arg_2>
or     <arg_1>, <arg_2>
mov    <arg_1>, <arg_2>
```

# Assembling: Translating Assembly to Binary

Every assembly **operator** has a corresponding **unique binary code**.

```
add     -->     0001
subt    -->     0010
and     -->     0011
or      -->     0100
mov     -->     0101
```

Every assembly **operand** has a **rule of translation** to binary encoding **based on its type**.

```
123     -->     1111011
‘c’     -->     1100011
```

# Assembling: Translating Assembly to Binary

The translated binary operation is ready for sending to the processor to execute.

```
        add      $123,      $15
-->     0001     1111011    0001111
```

How does code work?

# The PL Translator

# Translator Software

- Takes a PL source file as input..
  - e.g. demo.c
- Parses the text of the source file..
- Detects structure in the source code that it understands..
  - e.g. `int main() { ... }`
    - This is a function: a sequence of related instructions).
  - e.g. `int year = 2022;`
    - This is a variable: a piece of identifiable data.
- Translates source code structure to binary code structure..
  - Instructions are decomposed into binary encodings.
  - Data is translated into its binary representation.

# Types of Translators

- Compiler
  - Parses and translates full source file.
  - Produces a binary executable file that can be run separately.
    - e.g. *gcc* compiler for C PL

- Interpreter
  - Parses and translates source file line-by-line.
  - Runs each line after translating it.
  - Does not produce an output binary file.
    - e.g. *python3* interpreter for Python PL

# Types of Translators

- Every translator is programmed to detect the structure of its corresponding PL.
  - We write code in a PL following the language’s rules.
  - The translator detects language rules in the source code.

How does code work?

# Python, C

| Python | C |
| --- | --- |
| `# this is a comment...` | `// this is a comment...` |
| `variable = value` | `type variable = value;`<br>`variable = value2;` |
| `var = val +/-* varX` | `type var = val +/-* varX;` |
| `function(varX, valY, ...)` | `function(varX, valY, ...);` |
| `var = func(varX, valY, ...)` | `type var = func(varX, valY, ...);` |
| `def func(var1, var2, ...):`<br>`  var = var1`<br>`  varX = func2(var2)`<br>`  return var + varX` | `type func(type var1, ...) {`<br>`  type var = var1;`<br>`  type varX = func2(var2);`<br>`  return var + varX;`<br>`}` |