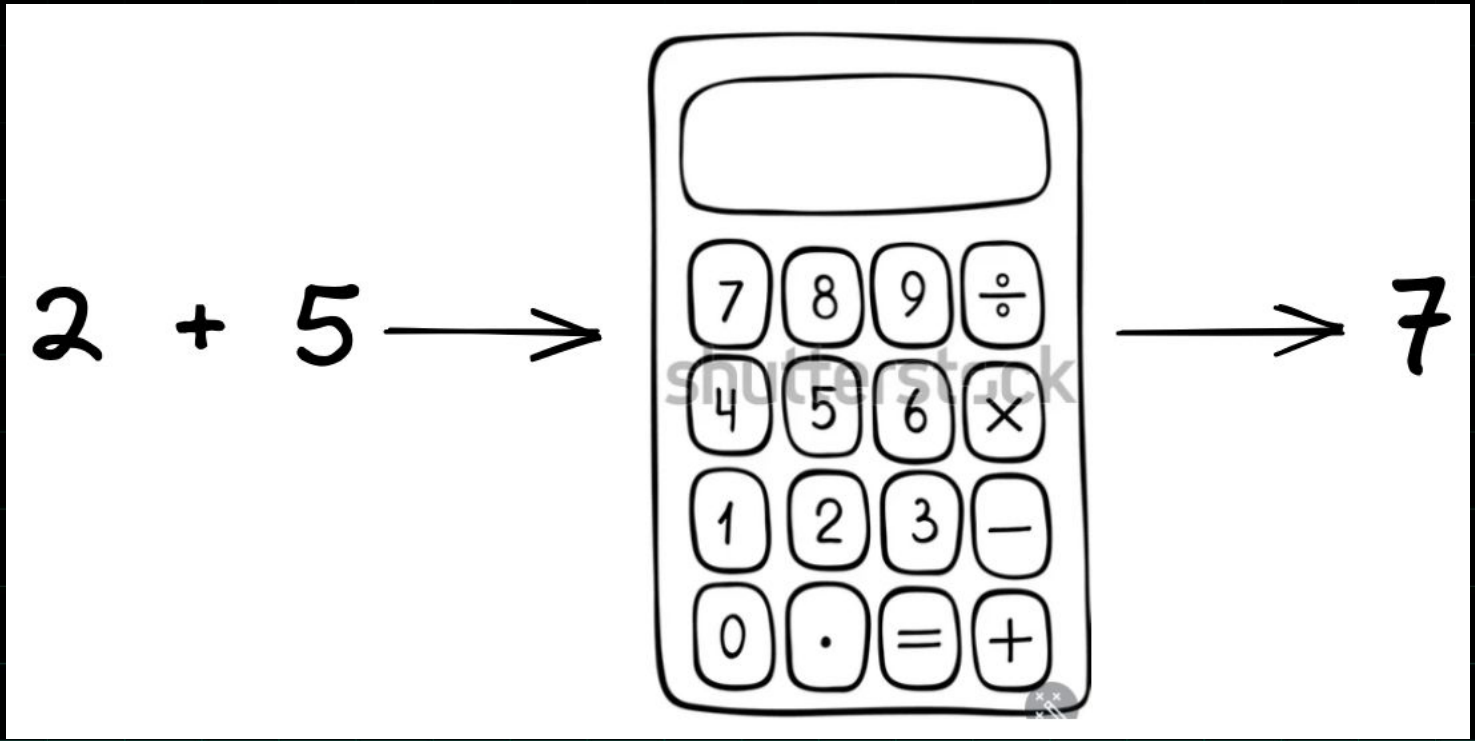
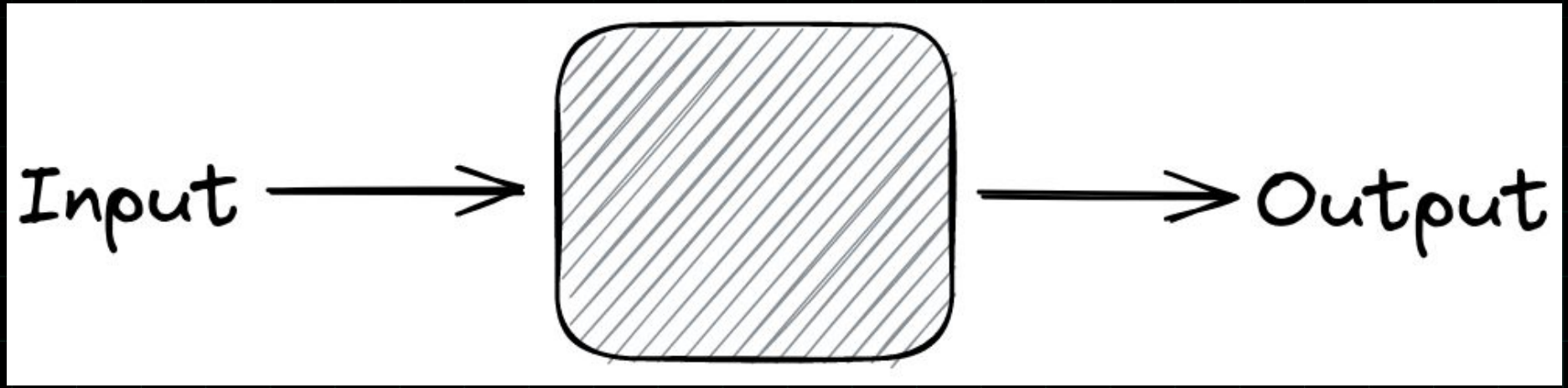
PROGRAM COMPILATION & EXECUTION

College of Science & Technology, Royal University of Bhutan

PROGRAM



PROGRAM



PROGRAM

Instructions to computers telling them what to do. (It is very very specific)

- With the input and what the output should be

You write a program using a computer programming language

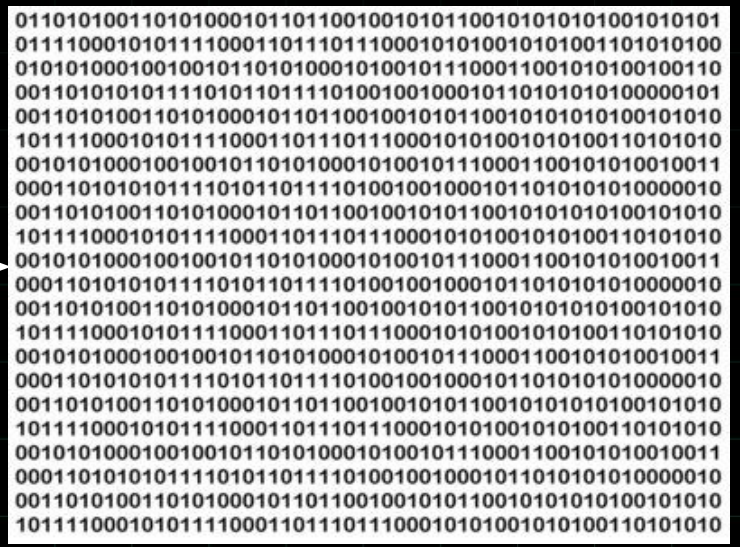

```
// Carbon:
package Geometry api;
import Math;

class Circle {
  var r: f32;
};

fn PrintTotalArea(circles: Slice(Circle)) {
  var area: f32 = 0;
  for (c: Circle in circles) {
    area += Math.Pi * c.r * c.r;
  }
  Print("Total area: {0}", area);
}

fn Main() -> i32 {
  // A dynamically sized array, like `std::vector`.
  var circles: Array(Circle) = ({.r = 1.0},
                                {.r = 2.0});
  // Implicitly converts `Array` to `Slice`.
  PrintTotalArea(circles);
  return 0;
}
```


wait it's all a computer program?
Always has been

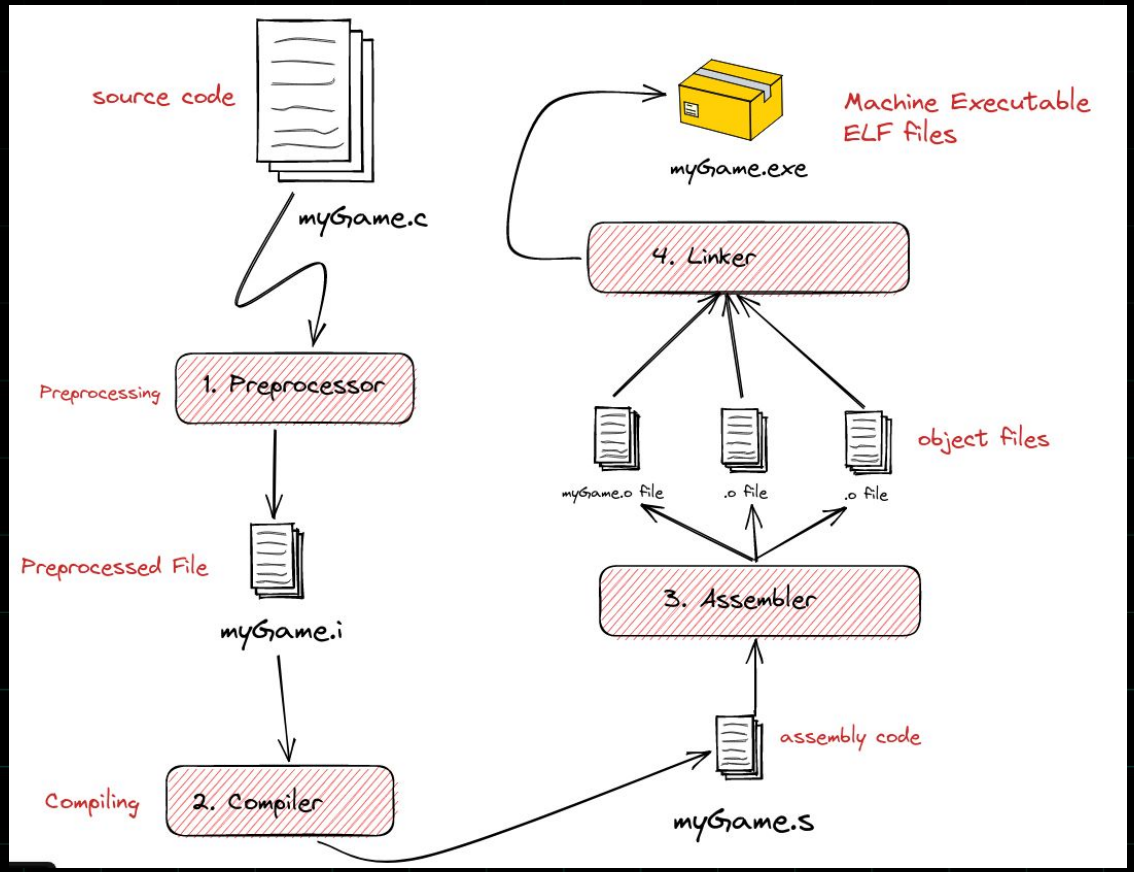
PROGRAM → MACHINE CODE

```
// Carbon:
package Geometry api;
import Math;

class Circle {
  var r: f32;
};

fn PrintTotalArea(circles: Slice(Circle)) {
  var area: f32 = 0;
  for (c: Circle in circles) {
    area += Math.Pi * c.r * c.r;
  }
  Print("Total area: {0}", area);
}

fn Main() -> i32 {
  // A dynamically sized array, like `std::vector`.
  var circles: Array(Circle) = ({.r = 1.0},
                                {.r = 2.0});
  // Implicitly converts `Array` to `Slice`.
  PrintTotalArea(circles);
  return 0;
}
```

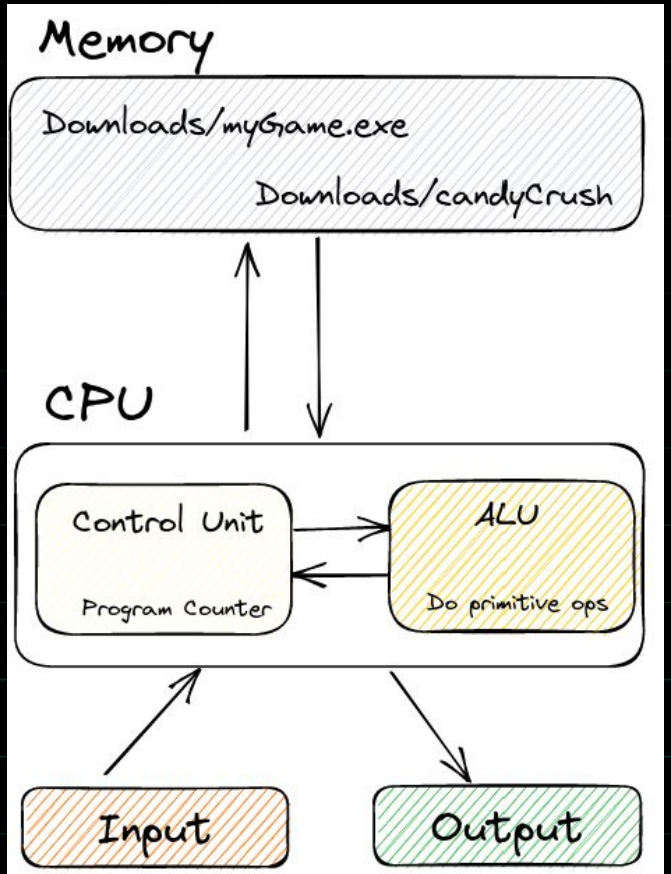
?



COMPILATION



EXECUTION



EXAMPLE

Source Code

game.c

```c
#include <stdio.h>

int main() {
  printf("Hello, World!\n");
  return 0;
}
```

game.c

Output

```
(base) ➜  c_test ./game
Hello, World!
(base) ➜  c_test
```

game

EXAMPLE

game.i

```
688
689 extern void perror (const char *__s);
690
691 extern int fileno (FILE *__stream) __attribute__ ((__nothrow__ , __leaf__)) ;
692
693 extern int fileno_unlocked (FILE *__stream) __attribute__ ((__nothrow__ , __leaf__)) ;
694 # 823 "/usr/include/stdio.h" 3 4
695 extern int pclose (FILE *__stream);
696
697 extern FILE *popen (const char *__command, const char *__modes)
698   __attribute__ ((__malloc__)) __attribute__ ((__malloc__ (pclose, 1))) ;
699
700 extern char *ctermid (char *__s) __attribute__ ((__nothrow__ , __leaf__))
701   __attribute__ ((__access__ (__write_only__, 1)));
702 # 867 "/usr/include/stdio.h" 3 4
703 extern void flockfile (FILE *__stream) __attribute__ ((__nothrow__ , __leaf__));
704
705 extern int ftrylockfile (FILE *__stream) __attribute__ ((__nothrow__ , __leaf__)) ;
706
707 extern void funlockfile (FILE *__stream) __attribute__ ((__nothrow__ , __leaf__));
708 # 885 "/usr/include/stdio.h" 3 4
709 extern int __uflow (FILE *);
710 extern int __overflow (FILE *, int);
711 # 909 "/usr/include/stdio.h" 3 4
712
713 # 2 "game.c" 2
714
715 # 3 "game.c"
716 int main() {
717   printf("Hello, World!\n");
718   return 0;
719 }
```

EXAMPLE

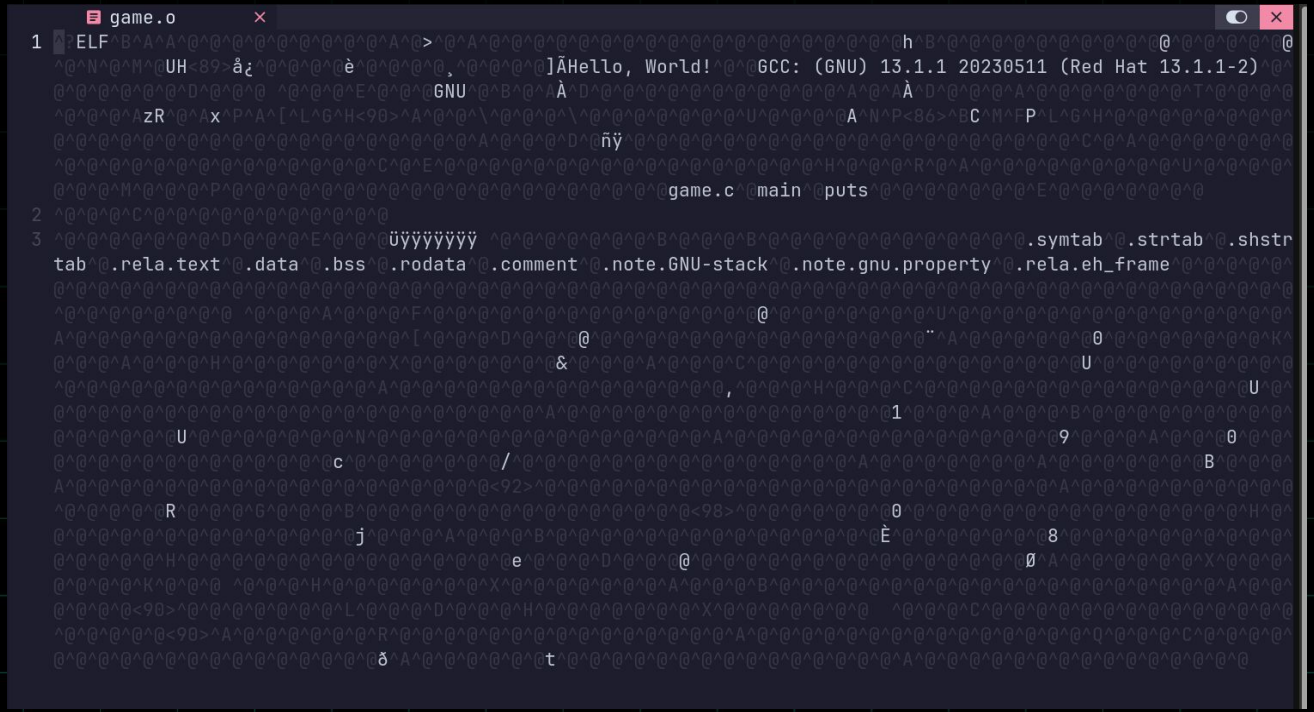
game.s

Assembly Code

game.s

```
1   .file "game.c"
2   .text
3   .section  .rodata
4 .LC0:
5   .string "Hello, World!"
6   .text
7   .globl  main
8   .type main, @function
9 main:
10 .LFB0:
11   .cfi_startproc
12   pushq %rbp
13   .cfi_def_cfa_offset 16
14   .cfi_offset 6, -16
15   movq  %rsp, %rbp
16   .cfi_def_cfa_register 6
17   movl  $.LC0, %edi
18   call  puts
19   movl  $0, %eax
20   popq  %rbp
21   .cfi_def_cfa 7, 8
22   ret
23   .cfi_endproc
24 .LFE0:
25   .size main, .-main
26   .ident  "GCC: (GNU) 13.1.1 20230511 (Red Hat 13.1.1-2)"
27   .section  .note.GNU-stack,"",@progbits
```

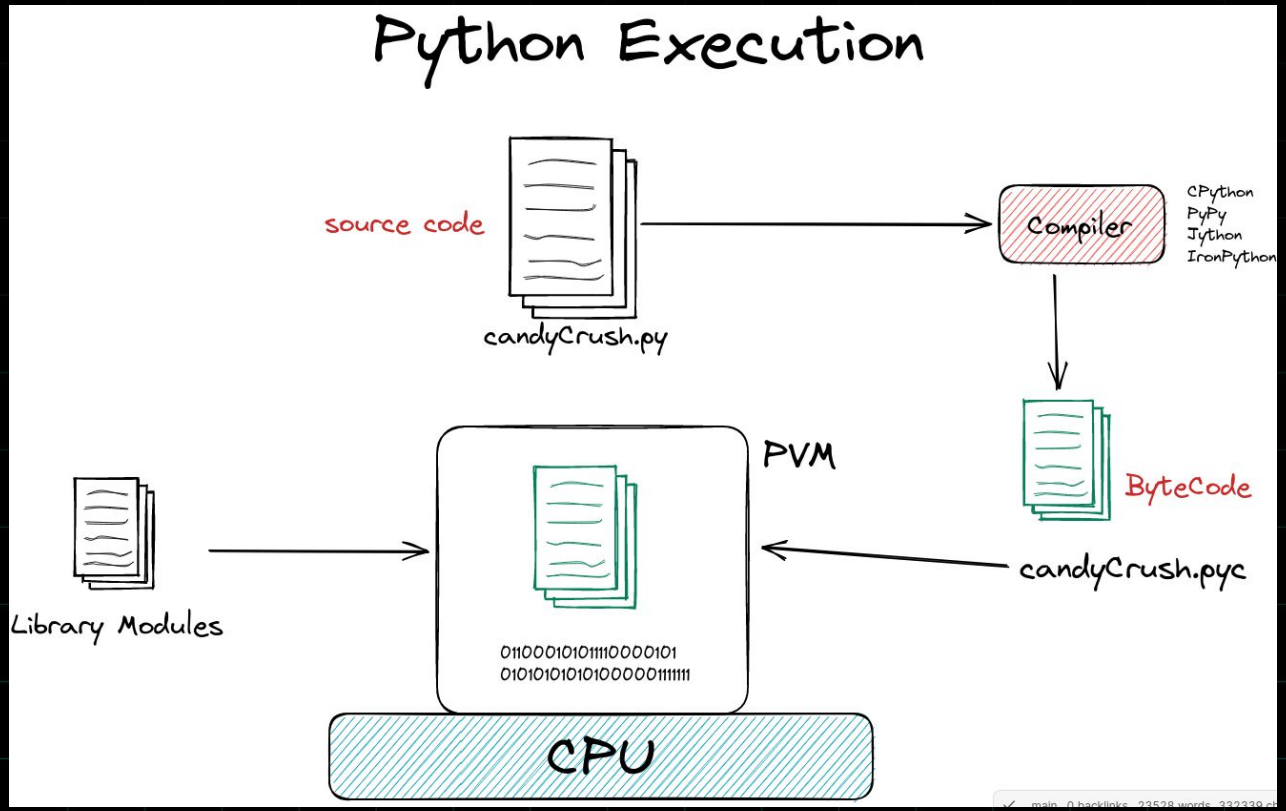
EXAMPLE

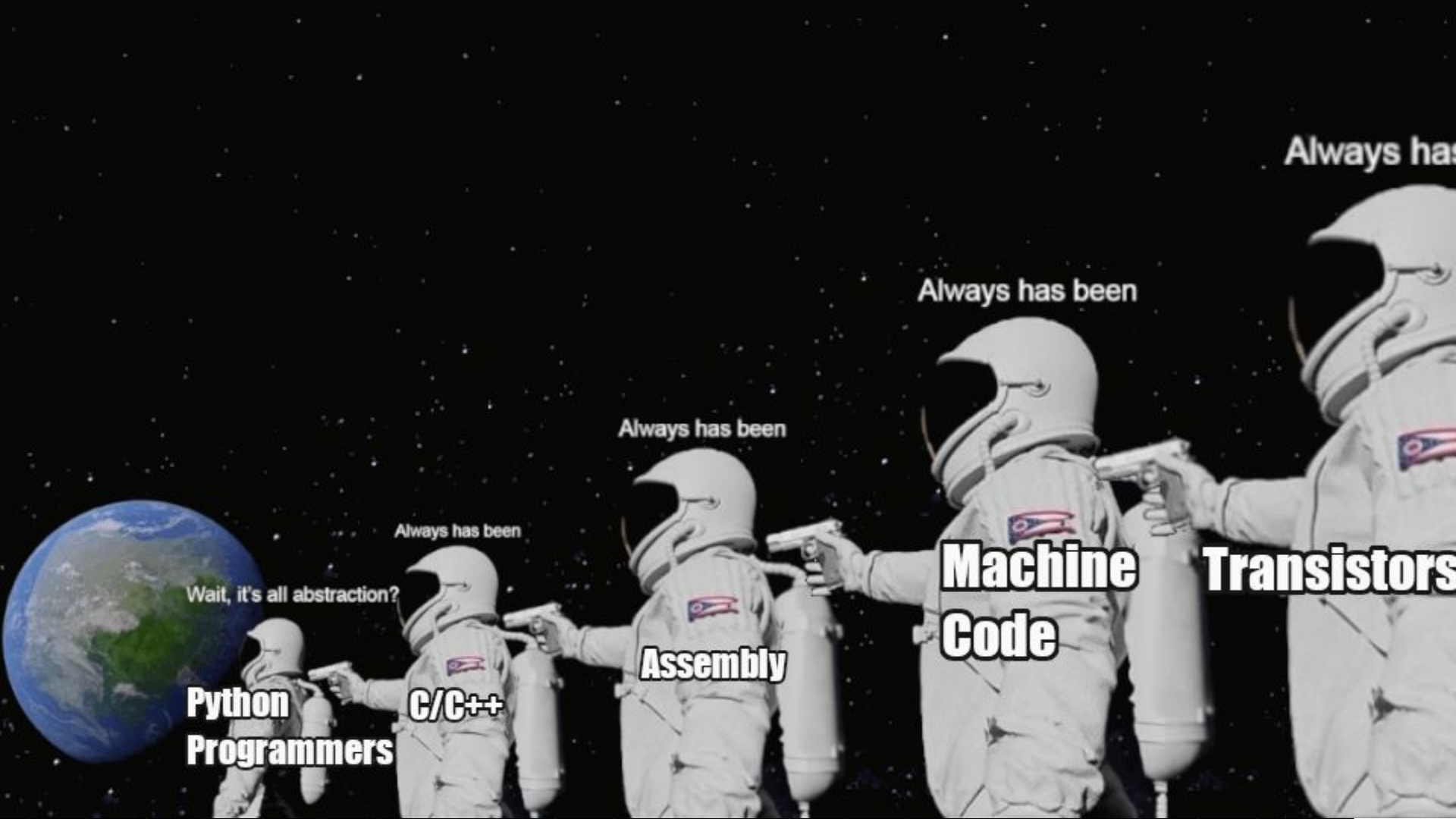


game.o

Machine Instructions

PYTHON EXECUTION




Always has
Always has been
Always has been
Always has been
Wait, it's all abstraction?
Python Programmers
C/C++
Assembly
Machine Code
Transistors

NEXT WEEK

1. Data Types
2. Strings
3. Arrays, Lists & Dictionaries
4. I/O (Input/Output)

Learn Python in 1 hour - YouTube

https://learn-python.adamemery.dev/basics

Thank you!

douglas.cst@rub.edu.bt
kamalacharya.cst@rub.edu.bt