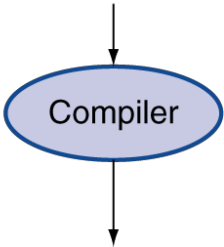


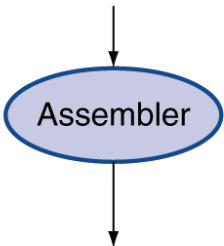
High-level
language
program
(in C)

```
swap(int v[], int k)
{int temp;
  temp = v[k];
  v[k] = v[k+1];
  v[k+1] = temp;
}
```



Assembly
language
program
(for MIPS)

```
swap:
    muli $2, $5, 4
    add  $2, $4, $2
    lw   $15, 0($2)
    lw   $16, 4($2)
    sw   $16, 0($2)
    sw   $15, 4($2)
    jr   $31
```

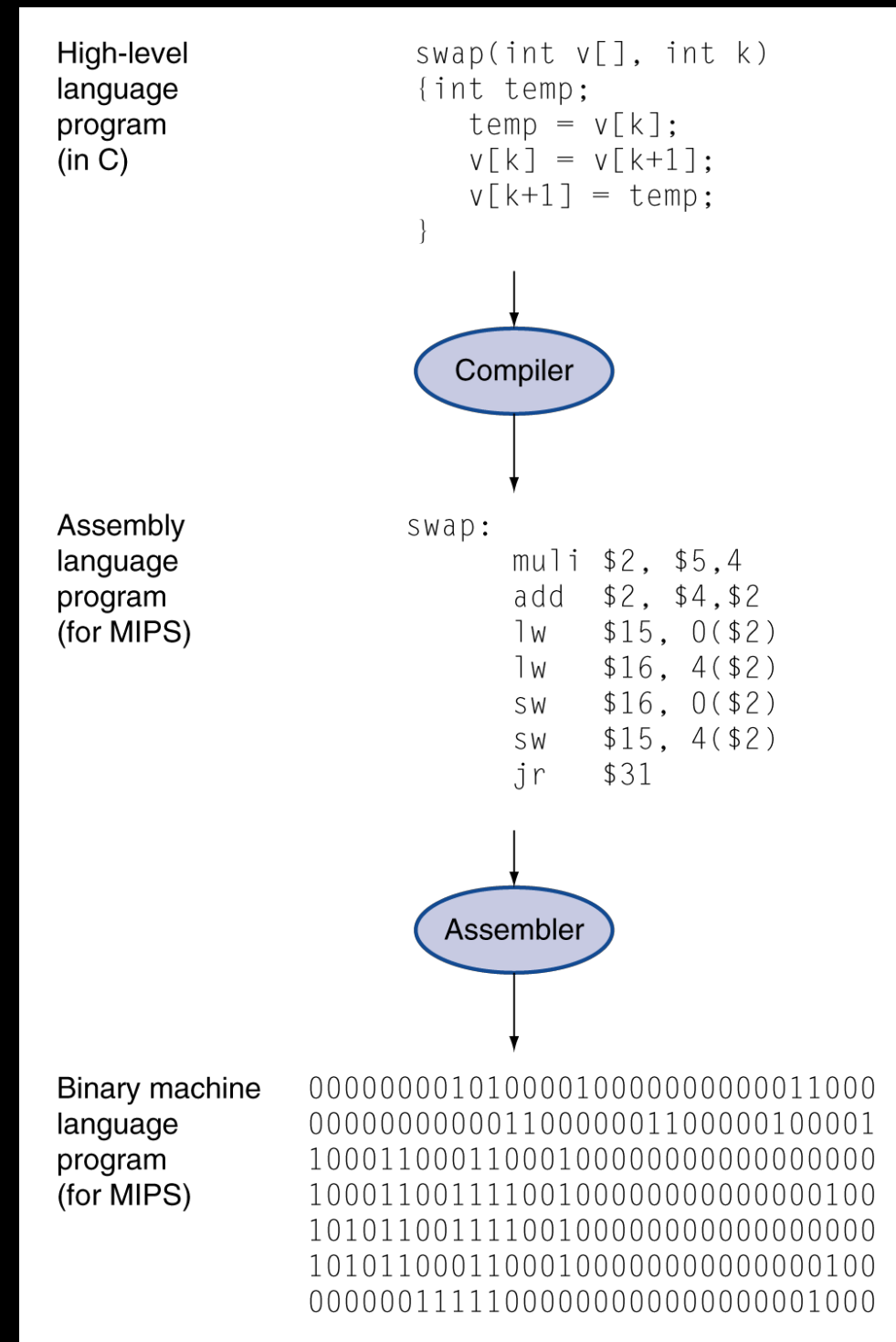


Binary machine
language
program
(for MIPS)

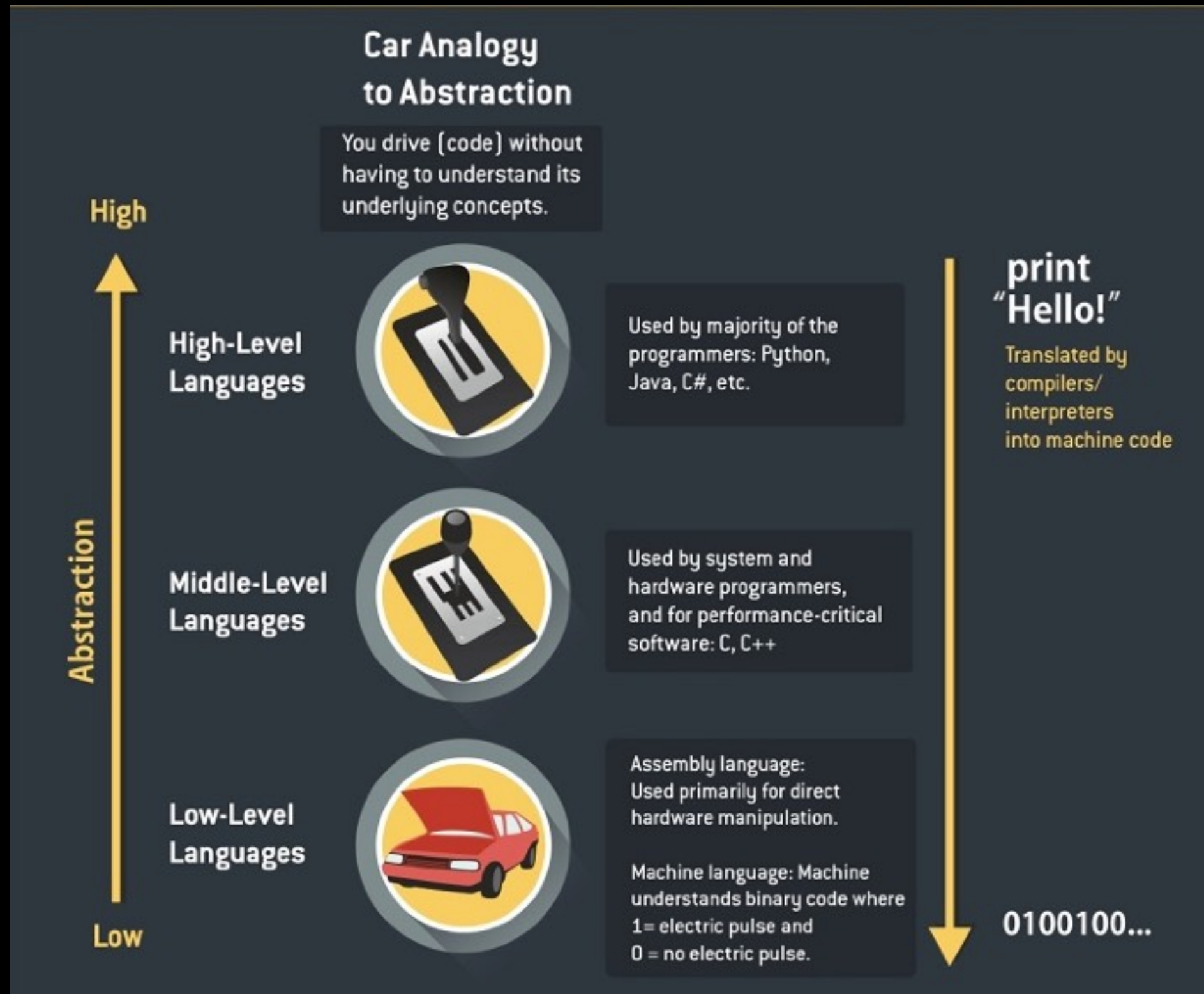
```
000000001010000100000000000011000
000000000000110000001100000100001
100011000110001000000000000000000
100011001111001000000000000000100
101011001111001000000000000000000
101011000110001000000000000000100
00000011111000000000000000001000
```

HOW DOES PROGRAMMING ACTUALLY MAKE A COMPUTER WORK?

- High-level and middle-level languages
 - Level of abstraction closer to problem domain.
 - Provides for productivity and portability.
 - Examples include C and C++.
- Assembly language
 - Textual representation of instructions for a computer.
- Hardware representation
 - Binary digits (bits, e.g., 1s and 0s).
 - Encoded instructions and data.



OR, IN OTHER WORDS...



Abstraction

ROM

Programming
ROM-level



0 = no electric bike
1 = electric bike and
understands binary code where
Machine language: Machine

hardware manipulation
used binary for direct
Assembly language:

Programming
Middle-level



software: C, C++
and for performance-critical
hardware programmers
used in system and

Programming
High-level



high-level
programming and

0100100...

high-level code
understands

code

code

code

code