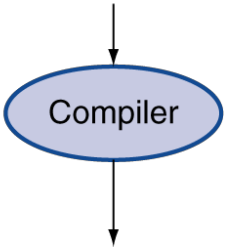
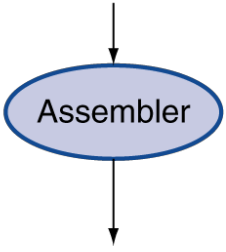
High-level language program (in C)

```
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
```

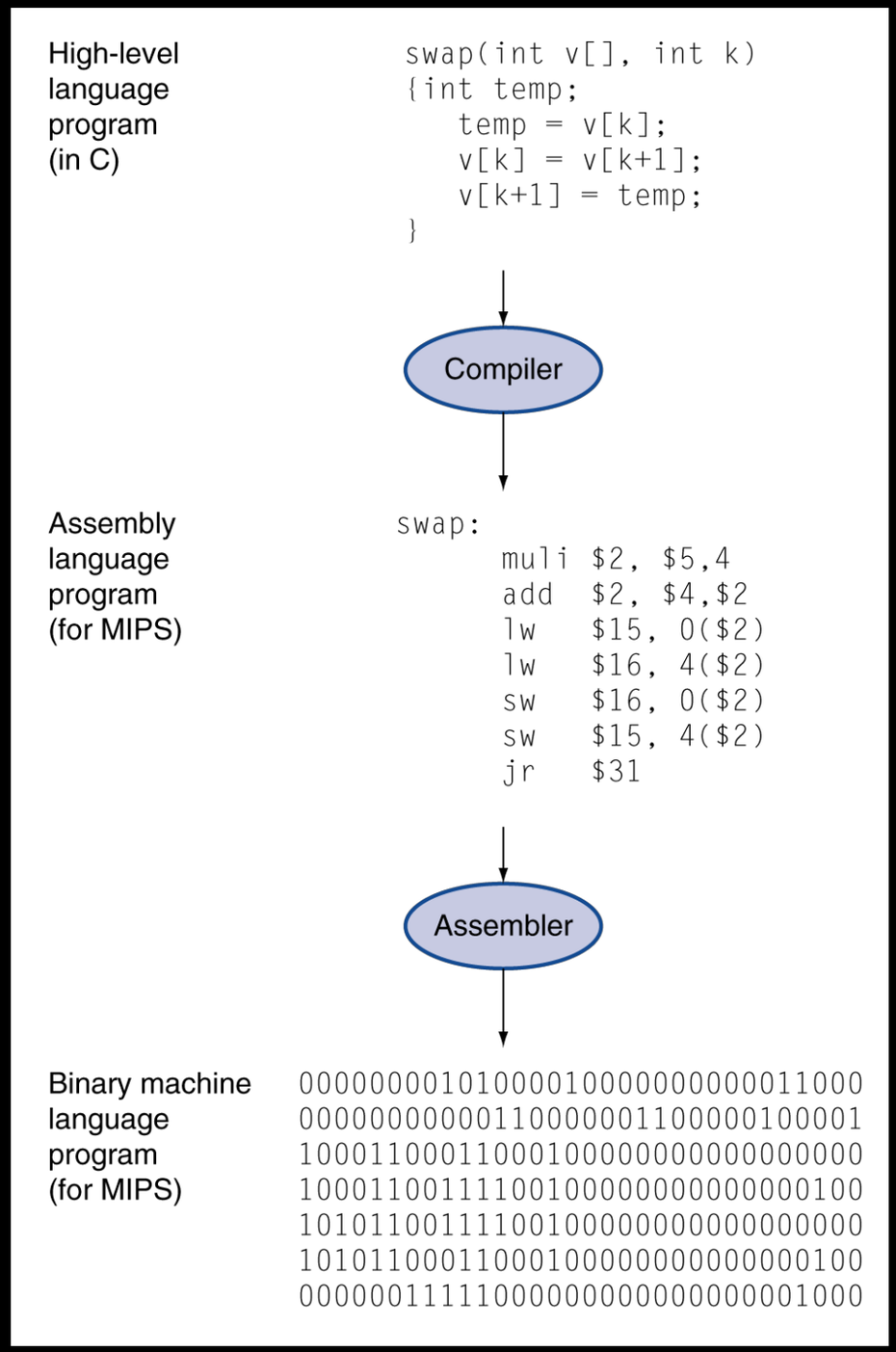


Binary machine language program (for MIPS)

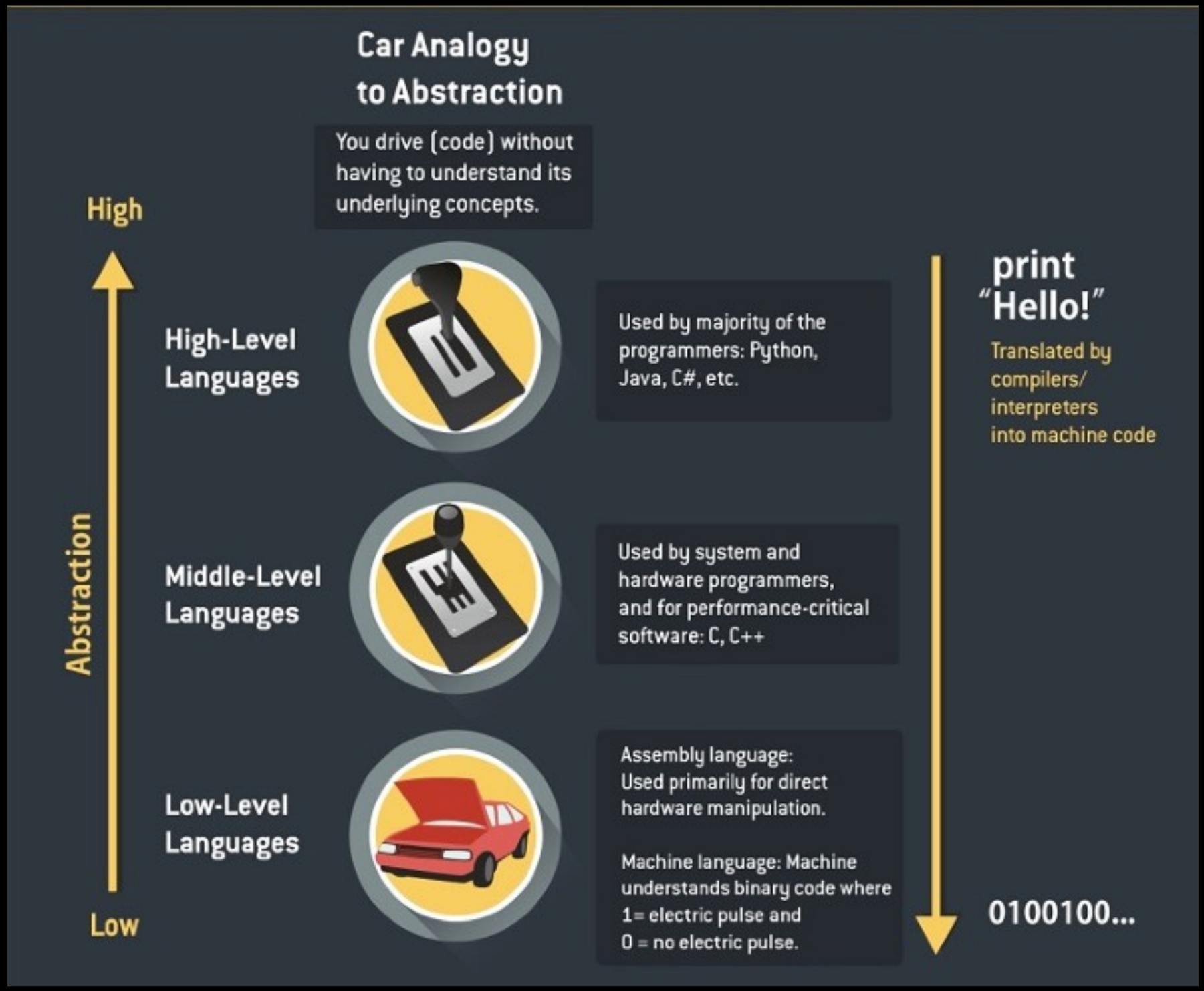
```
00000000101000010000000000011000
00000000000011000000110000100001
10001100011000100000000000000000
10001100111100100000000000000100
10101100111100100000000000000000
10101100011000100000000000000100
00000011111000000000000000001000
```

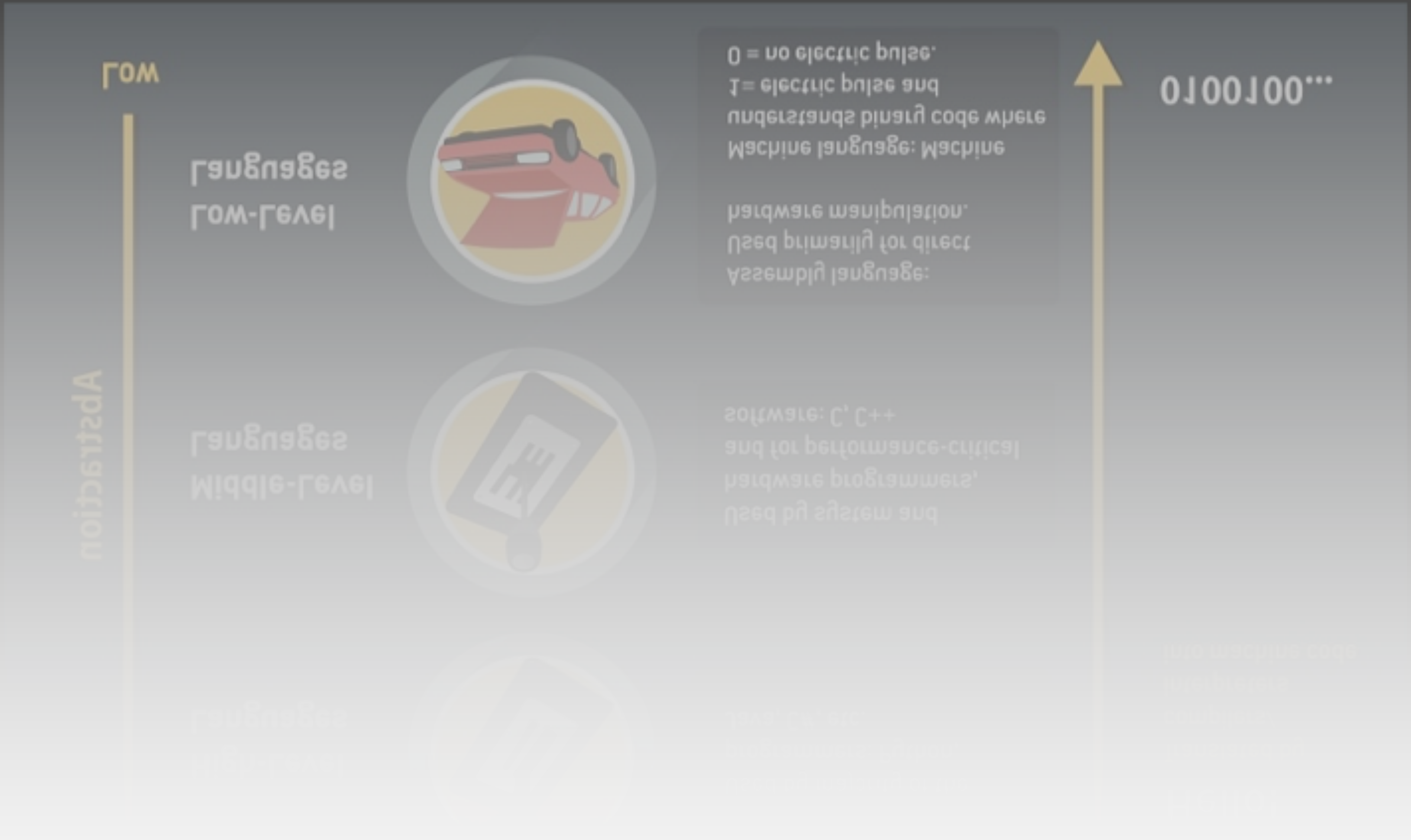
HOW DOES PROGRAMMING ACTUALLY MAKE A COMPUTER WORK?

- 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...




Low
Abstraction
Low-Level Languages
Assembly language: Used primarily for direct hardware manipulation.
Machine language: Machine understands binary code where 1 = electric pulse and 0 = no electric pulse.
0100100...
Middle-Level Languages
Used by system and hardware programmers, and for performance-critical software: C, C++