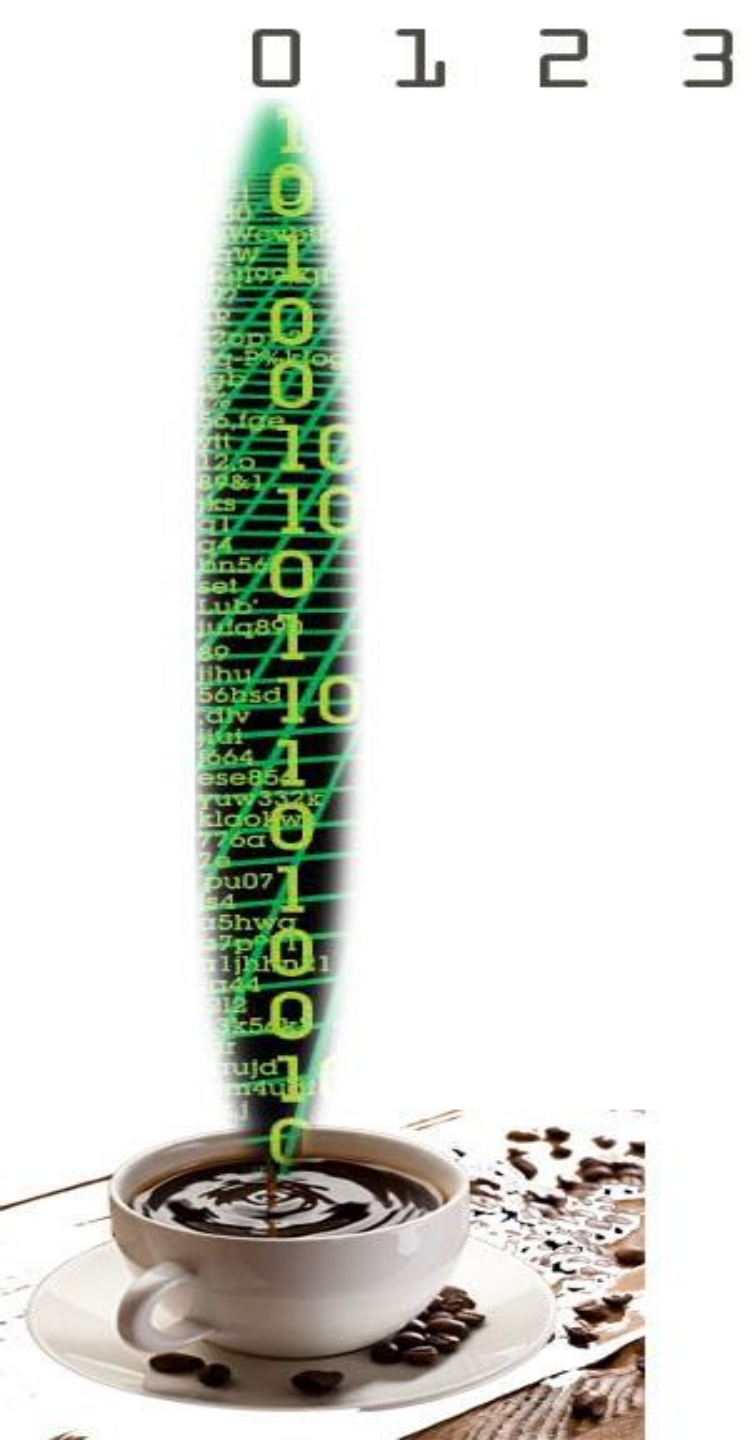


ABI & API Interfaces

Application Binary Interface

Application Programming Interface

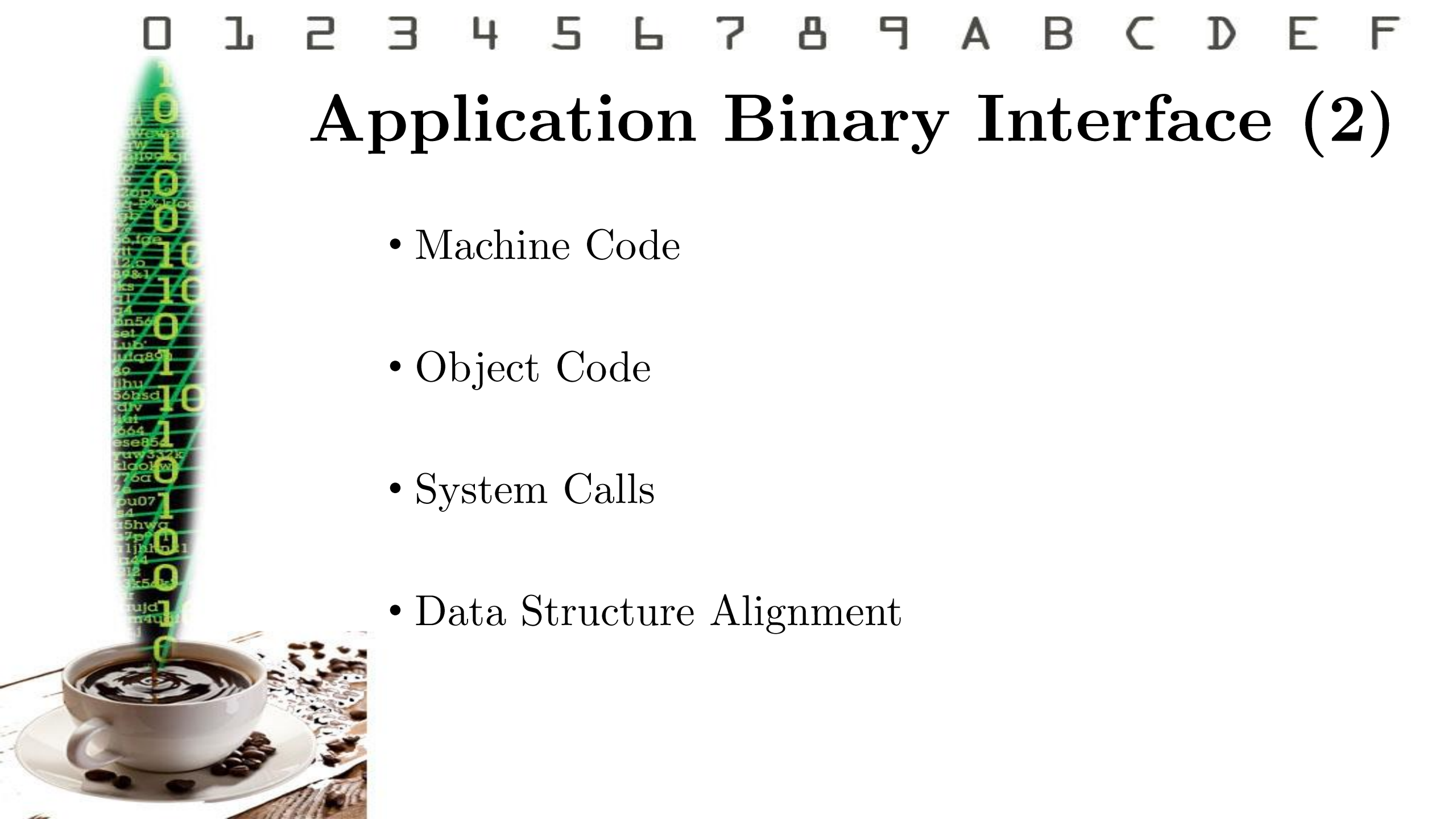
Dimitrios N. Terzopoulos

terzopod@math.auth.gr

4/11/2015

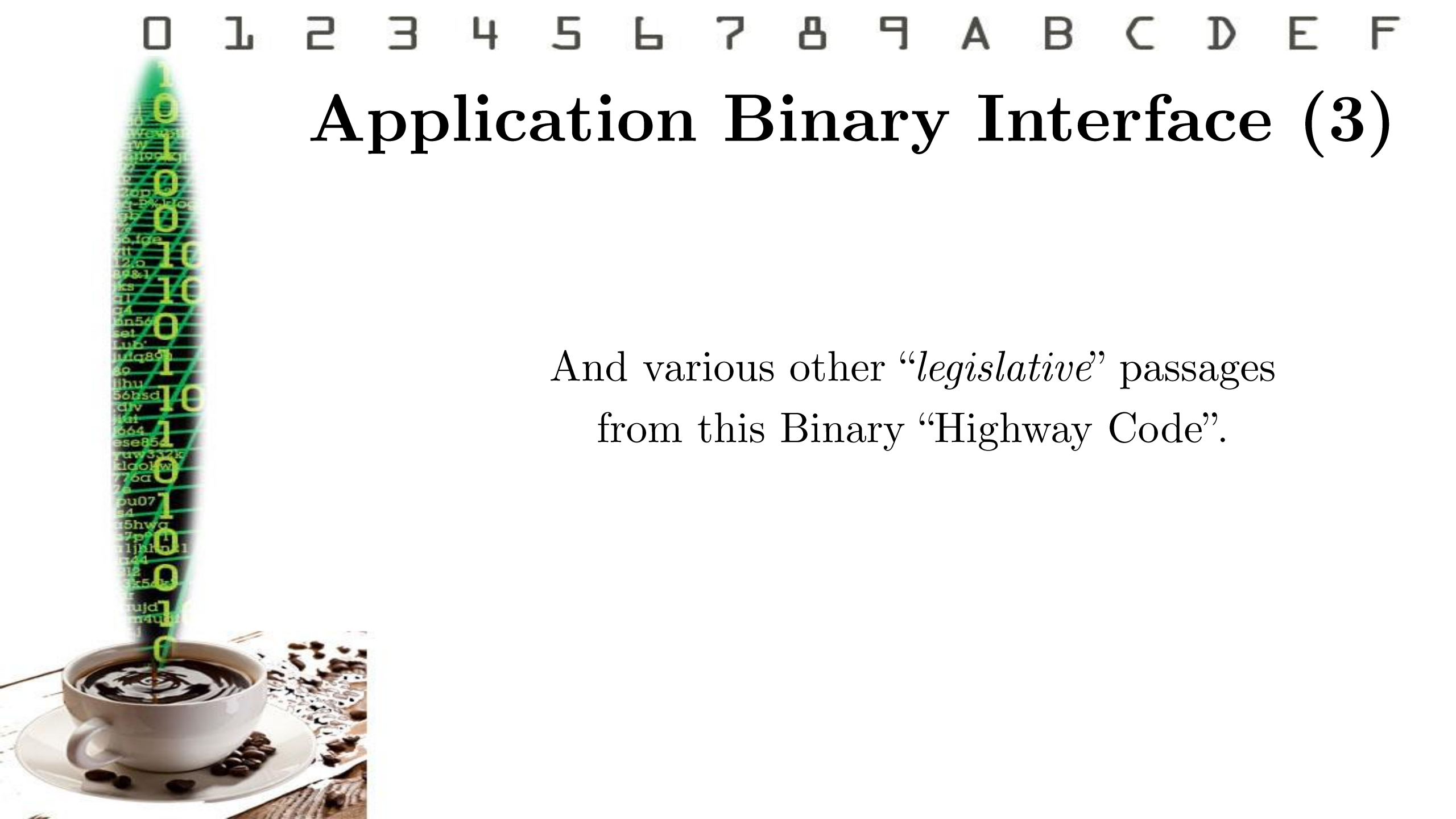
Application Binary Interface

- Calling Convention
- Call Stack
- Name Mangling
- Register Allocation



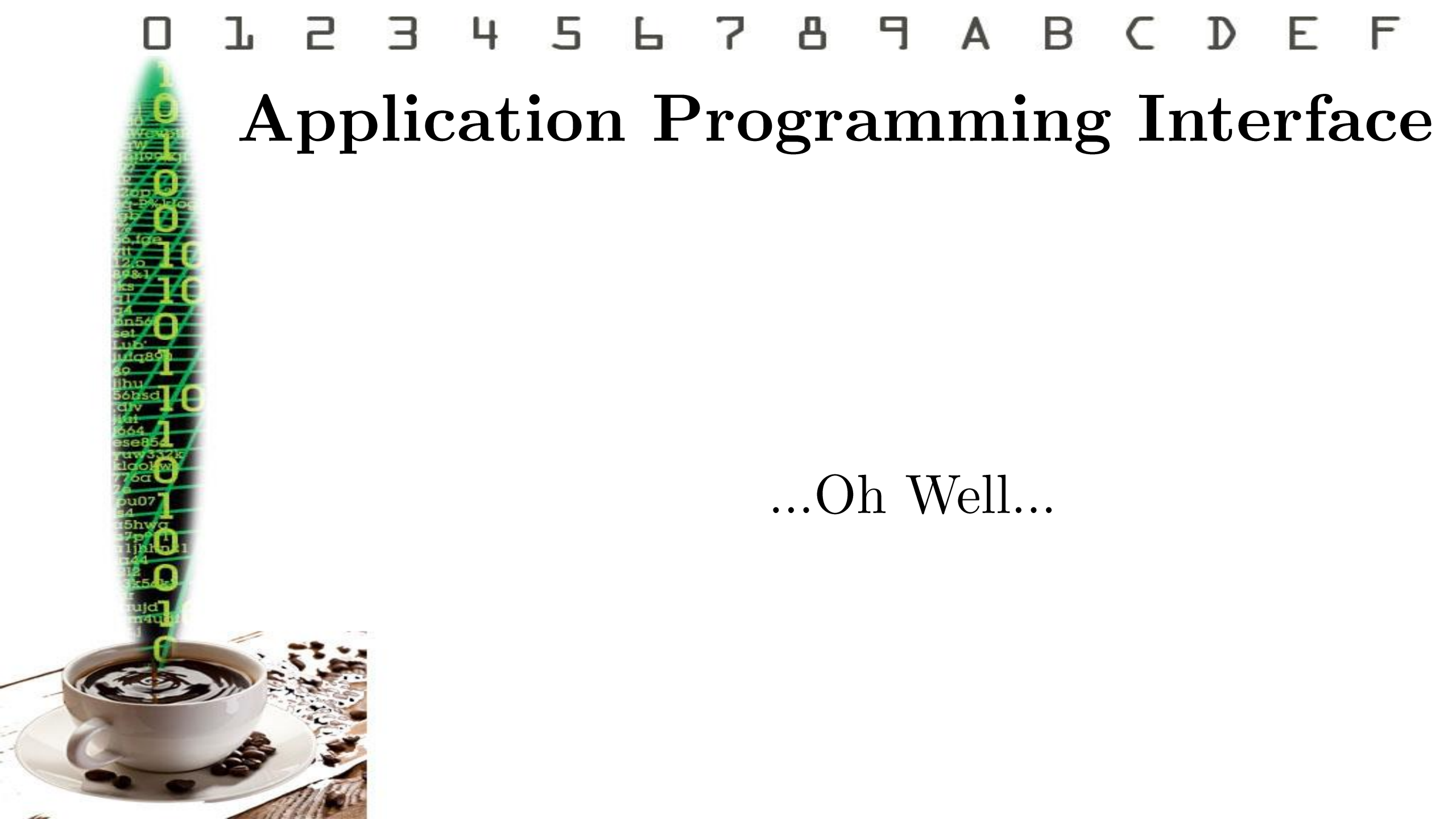
Application Binary Interface (2)

- Machine Code
- Object Code
- System Calls
- Data Structure Alignment



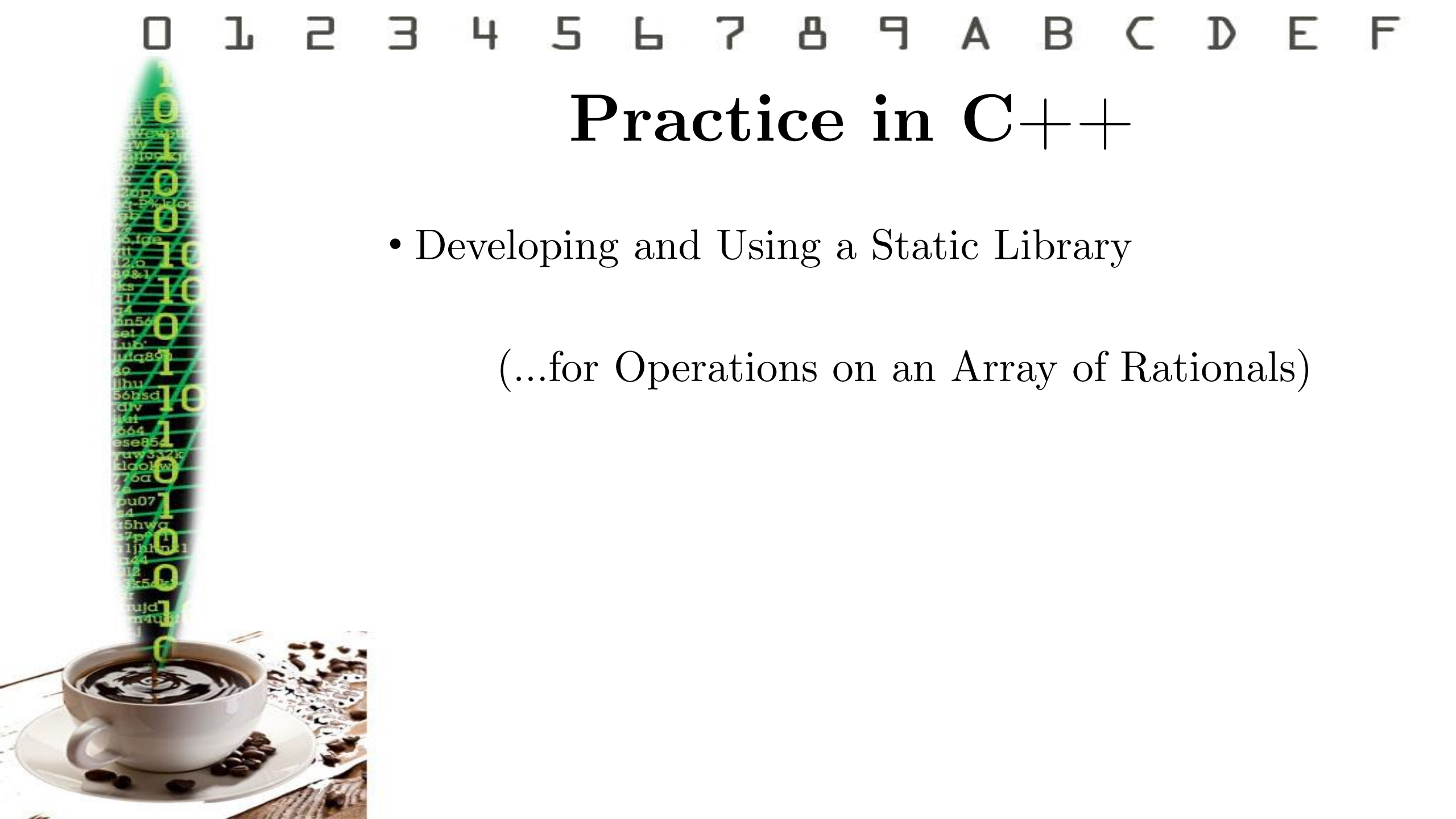
Application Binary Interface (3)

And various other "*legislative*" passages
from this Binary "Highway Code".



Application Programming Interface

...Oh Well...



Practice in C++

- Developing and Using a Static Library

(...for Operations on an Array of Rationals)

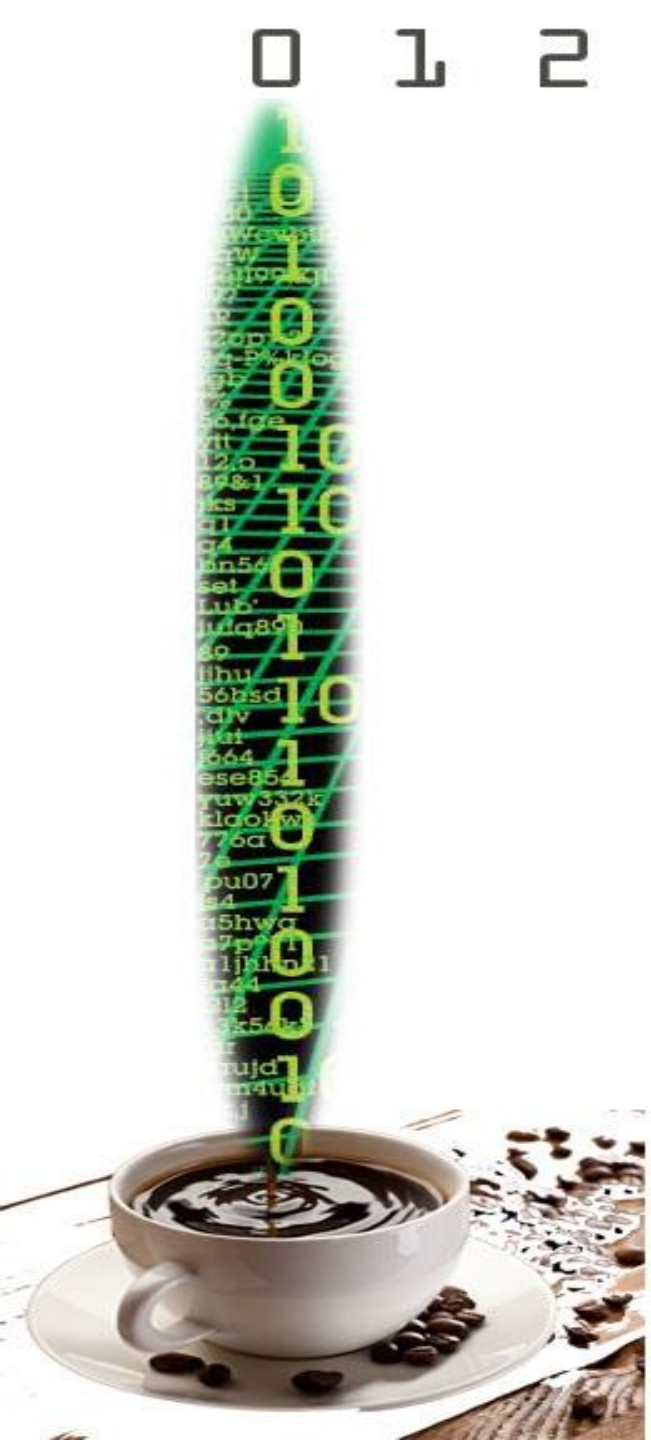
0 1 2 3 4 5 6 7 8 9 A B C D E F

Application Binary Interface

• The *Interface* between two Programs

At the level of **Machine Code**

Program	**Organism**
• *Machine Code*	• *DNA*
• Bytes	• Elements
• Variables (e.g. Integer)	• Molecules (e.g. NH_3)
• Functions	• Genes
• External Library Linking	• Hormones
...	...
• Module Communication	• Organism Interactions



Application Binary Interface

- The *Interface* between two Programs

At the level of **Machine Code**

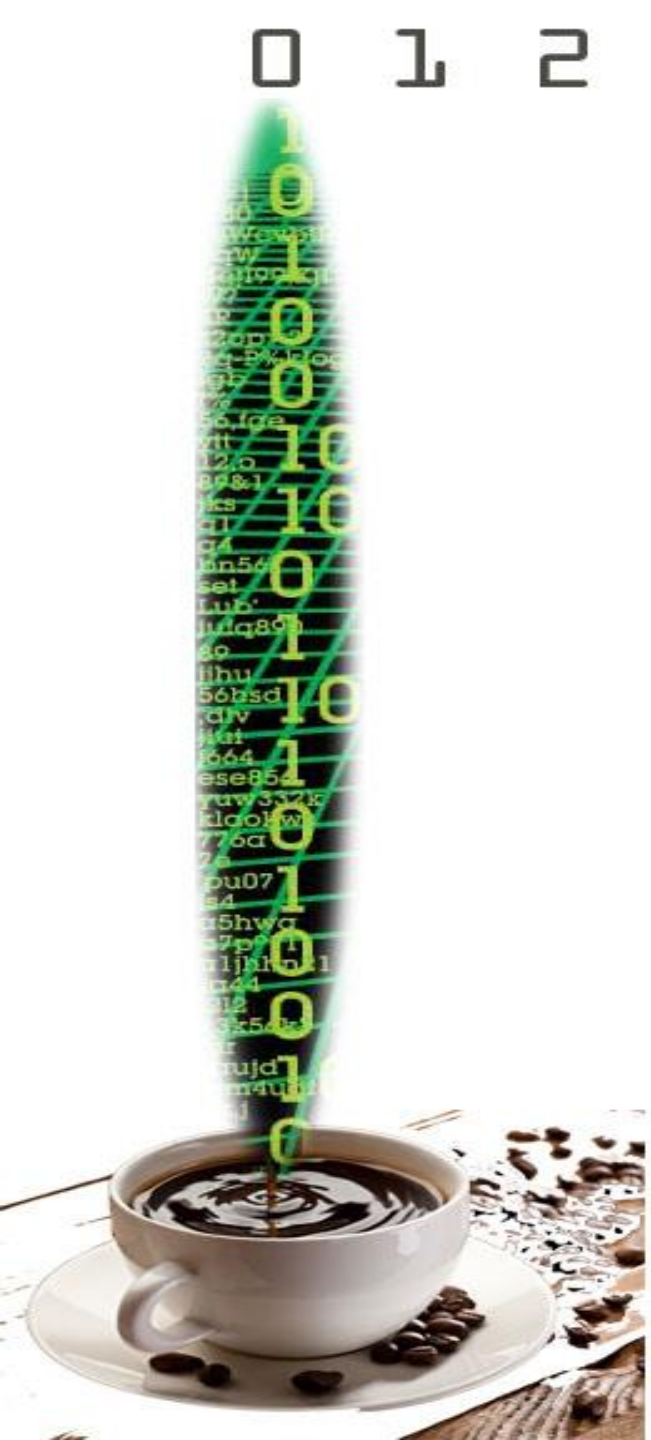
Program

Two programs must speak the "same" language

Organism

Chemistry IS a language!

(It is hard-wired in all organisms)



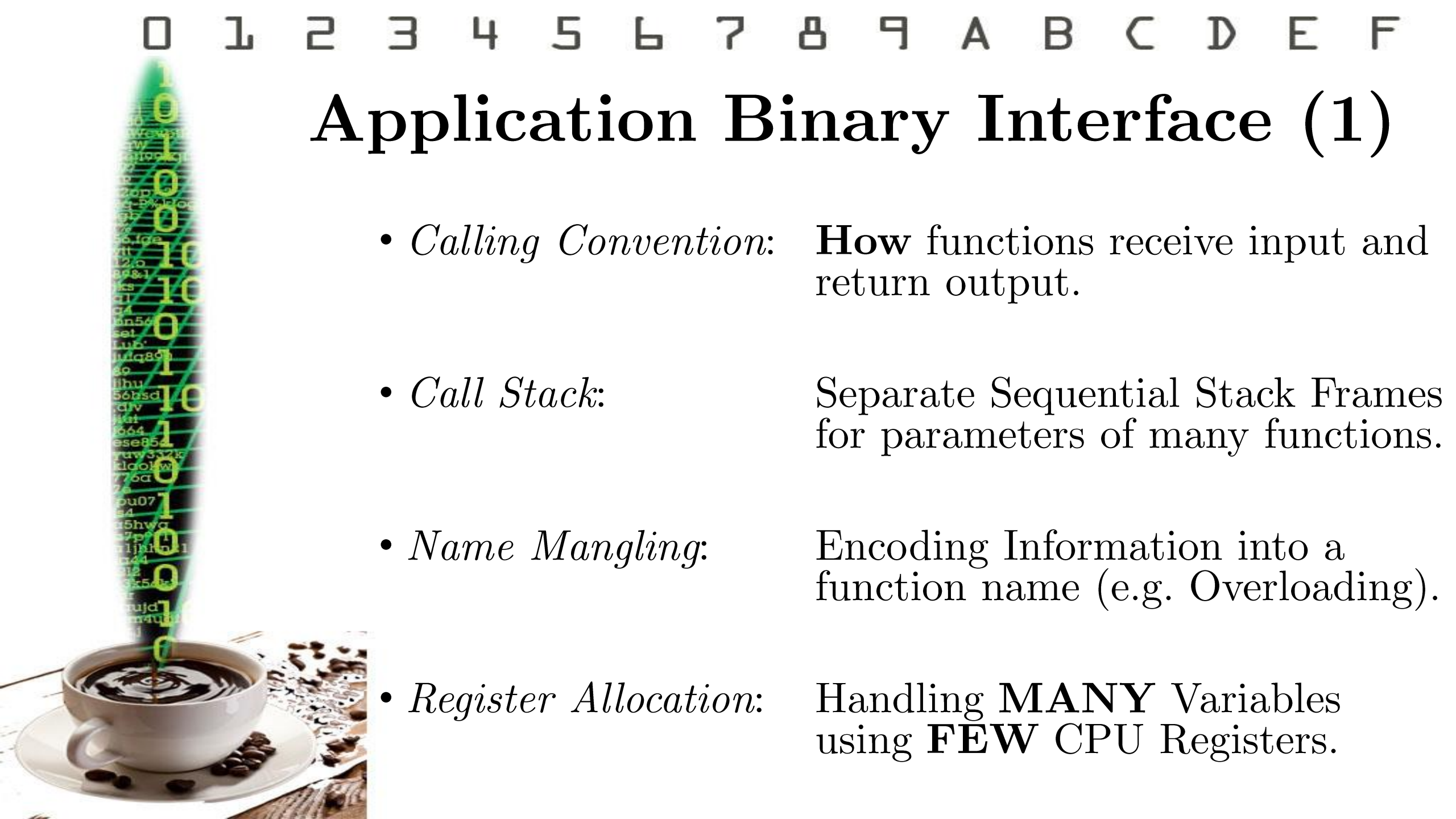
Application Binary Interface

We write high-level code in ...English (or so)

. . . . . .

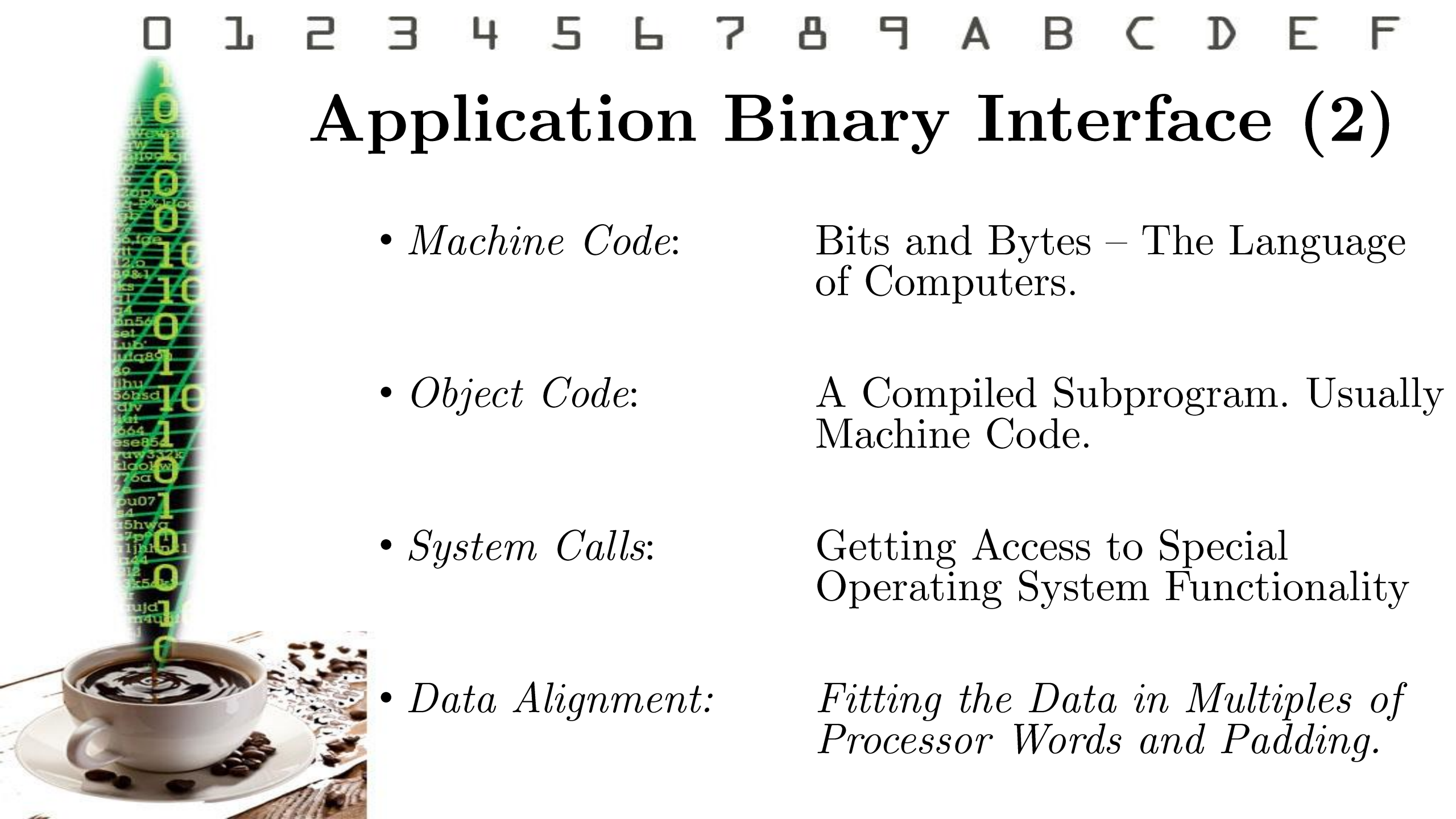
The compiler performs "*microsurgery*" on our program to make it fit in the "*computer world*".

Computer World → Programs **TALK** to each other!



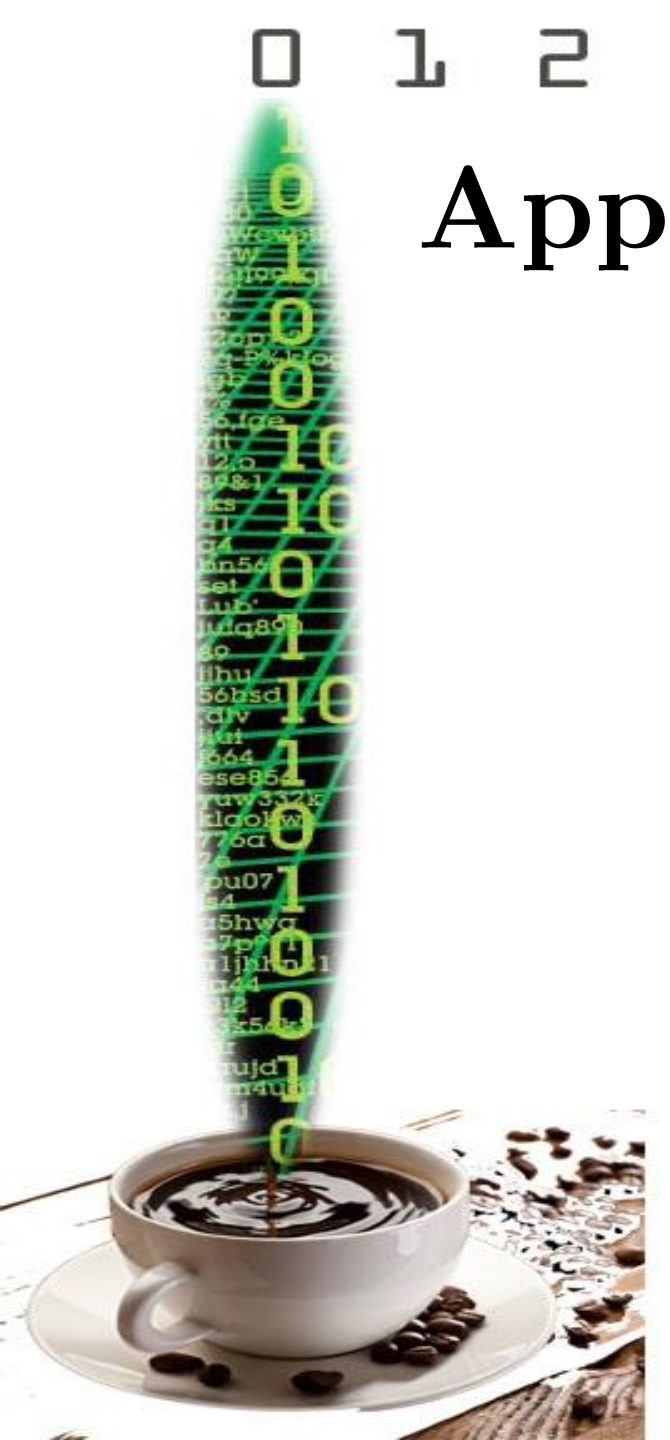
Application Binary Interface (1)

- *Calling Convention*: **How** functions receive input and return output.
- *Call Stack*: Separate Sequential Stack Frames for parameters of many functions.
- *Name Mangling*: Encoding Information into a function name (e.g. Overloading).
- *Register Allocation*: Handling **MANY** Variables using **FEW** CPU Registers.



Application Binary Interface (2)

- *Machine Code*: Bits and Bytes – The Language of Computers.
- *Object Code*: A Compiled Subprogram. Usually Machine Code.
- *System Calls*: Getting Access to Special Operating System Functionality
- *Data Alignment:* *Fitting the Data in Multiples of Processor Words and Padding.*

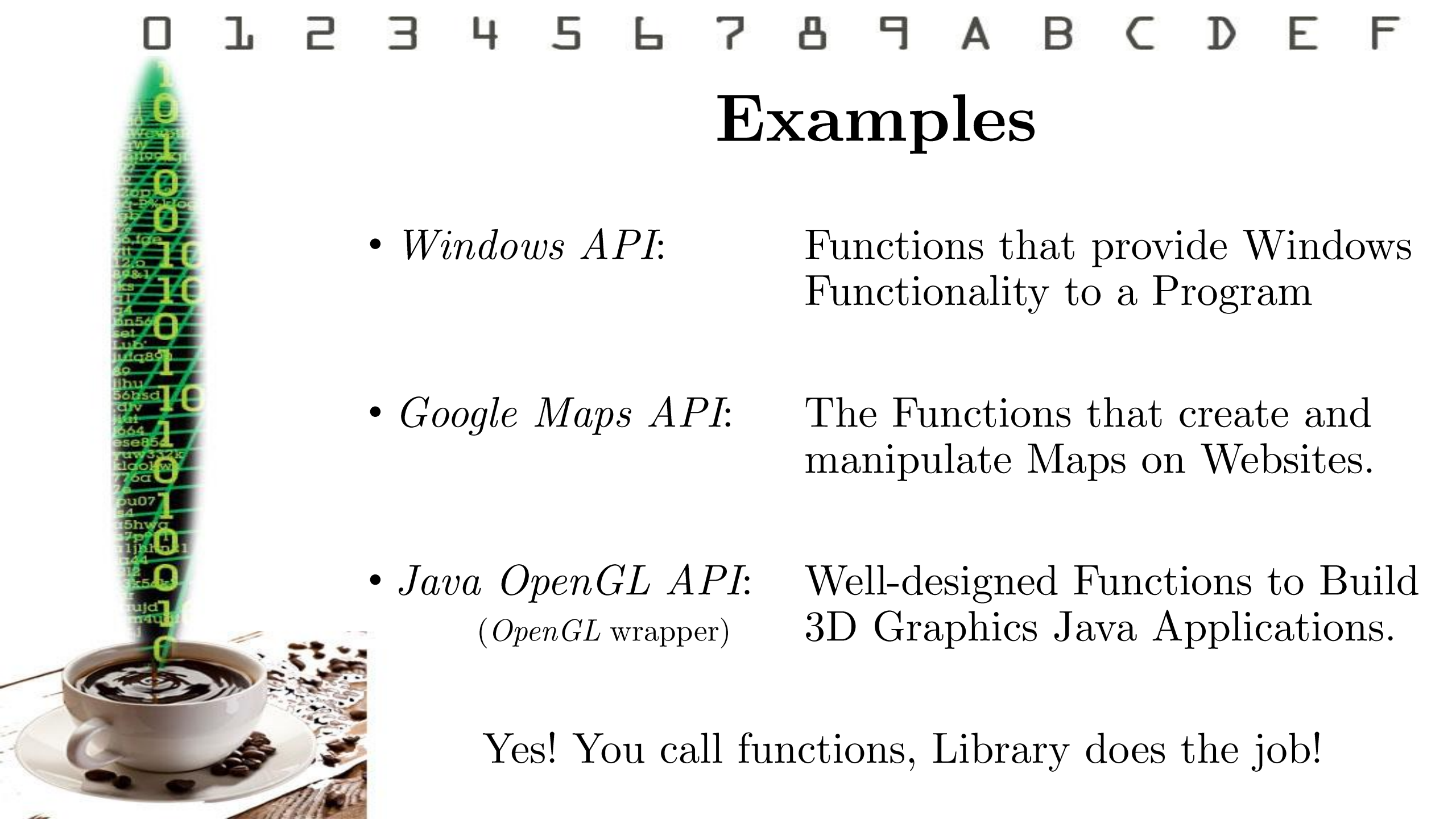


Application Programming Interface

- In short... The Instruction Manual for a Library.

- A Library Designer (remember?) assembles a Library.
- A Client Designer uses Libraries to build a Program.

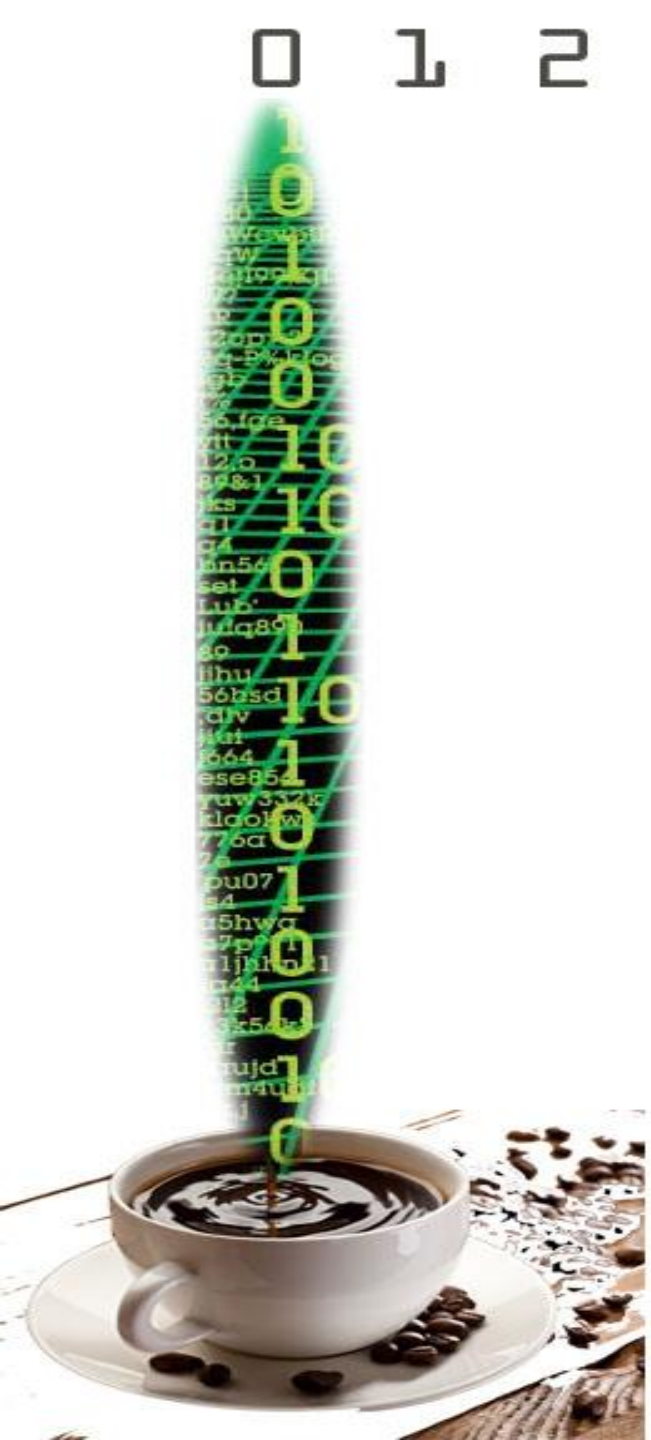
- API: A Description of the Functions of a Library, along with their Input/Output Types.



Examples

- *Windows API*: Functions that provide Windows Functionality to a Program
- *Google Maps API*: The Functions that create and manipulate Maps on Websites.
- *Java OpenGL API*: (*OpenGL* wrapper) Well-designed Functions to Build 3D Graphics Java Applications.

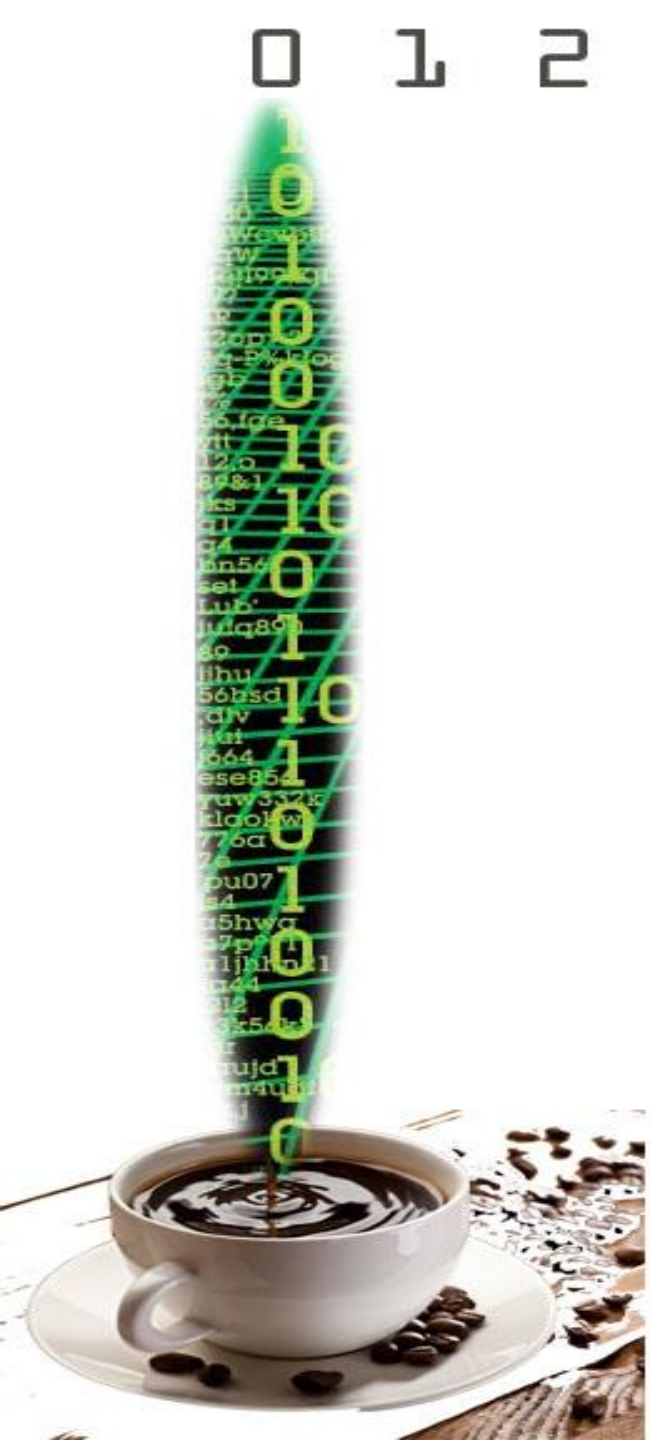
Yes! You call functions, Library does the job!



Coming Soon

Building a small Windows Program!

(That's right, no Console sham)



Thank you for your undistracted attention!

Let’s write some code...