flickr.com/photos/davidyuweb/14377756196

Hiun Kim

Introduction to Software Engineering

July 10, 2015 @ Visang Computer Club

Hiun Kim

現 Computer Science Undergraduate @ Sejong Univ.

現 Programmer @ Divtag Computer Research Group

前 Member @ Visang Computer Club

JavaScript, WAS Architecture Enthusiast

Software

Software is any set of machine-readable instructions that directs a computer's processor to perform specfic operation - via Wikipedia

http://download.intel.com/pressroom/kits/xeon/5500series/images/NehalemWaferShot1.jpg

11:13 PM 33

82 Visang 11

No
음영준 3/10/15, 4:17 PM

음영준
광사요? 3
12:46 AM

구교진
ㅇㅇ 3
12:46 AM

음영준

요즘 덥네여. Send

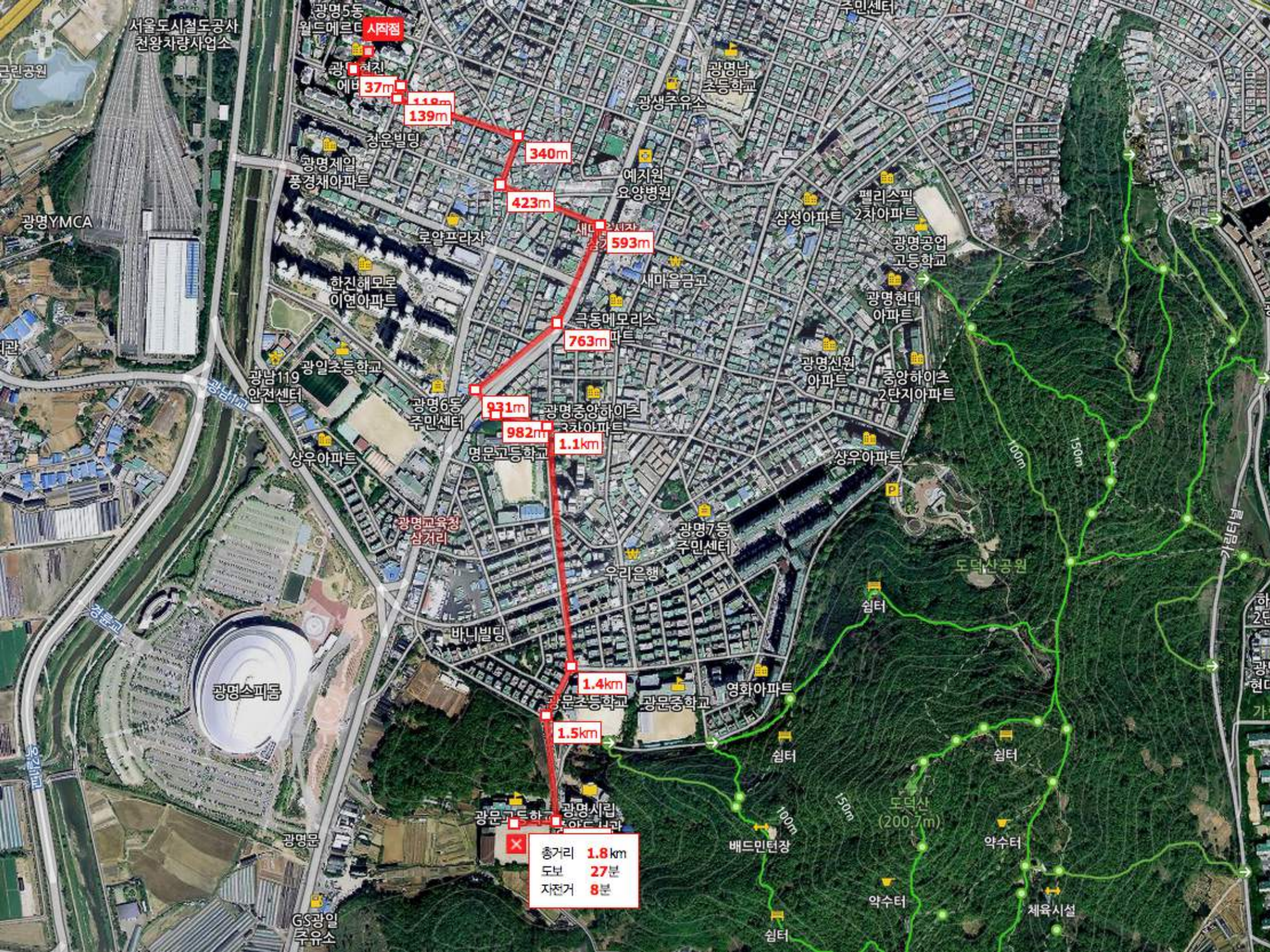
1 2 3 4 5 6 7 8 9 0

A single touch.

https://www.flickr.com/photos/wilhelmja/330965980

https://scontent.cdninstagram.com/hphotos-xpf1/t51.2885-15/e15/925342_1563221940557554_1351635329_n.jpg

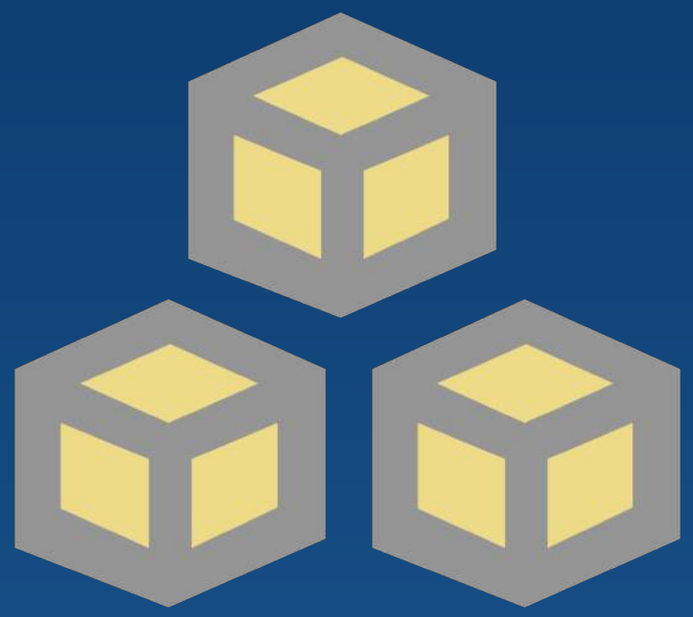
I'll go to school


시작점
37m
139m
340m
423m
593m
763m
931m
982m
1.1km
1.4km
1.5km
총거리 1.8km
도보 27분
자전거 8분
서울도시철도공사
천왕차량사업소
광명5동
주민센터
광명남
초등학교
광생주유소
청운빌딩
광명제일
풍경채아파트
예지원
요양병원
광명YMCA
로얄프라자
삼성아파트
펠리스필
2차아파트
광명공업
고등학교
한진해모로
이연아파트
새마을금고
광명현대
아파트
극동메모리스
광일초등학교
광남119
안전센터
광명6동
주민센터
광명중앙하이츠
2차아파트
광명신원
아파트
중앙하이츠
2단지아파트
명문고등학교
상우아파트
광명교육청
삼거리
광명7동
주민센터
우리은행
바니빌딩
광명스피돔
광문초등학교
광문중학교
영화아파트
쉼터
도덕산공원
도덕산
(200.7m)
배드민턴장
약수터
체육시설
광명문
GS광일
주유소
100m
150m

© WWF-Canon / André Bärtschi
For private use only - No reproduction or commercial use allowed.

Software Engineering

Software engineering is the study and an application of engineering to the design, development, and maintenance of software - via Wikipedia

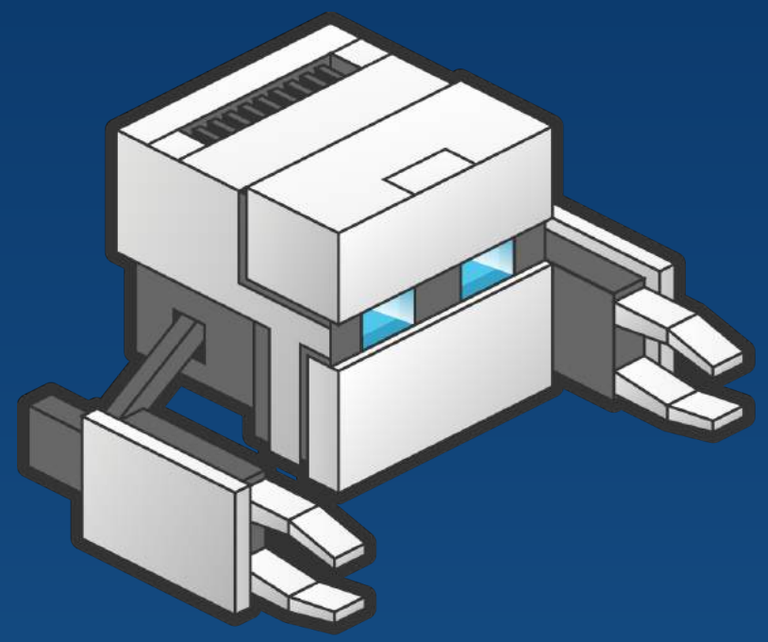


http://jspm.io

Software Engineering (cont.)

Solve problems in the real world through a medium - software. End user is human or machine.

software



https://travis-ci.org/ http://phonegap.com/

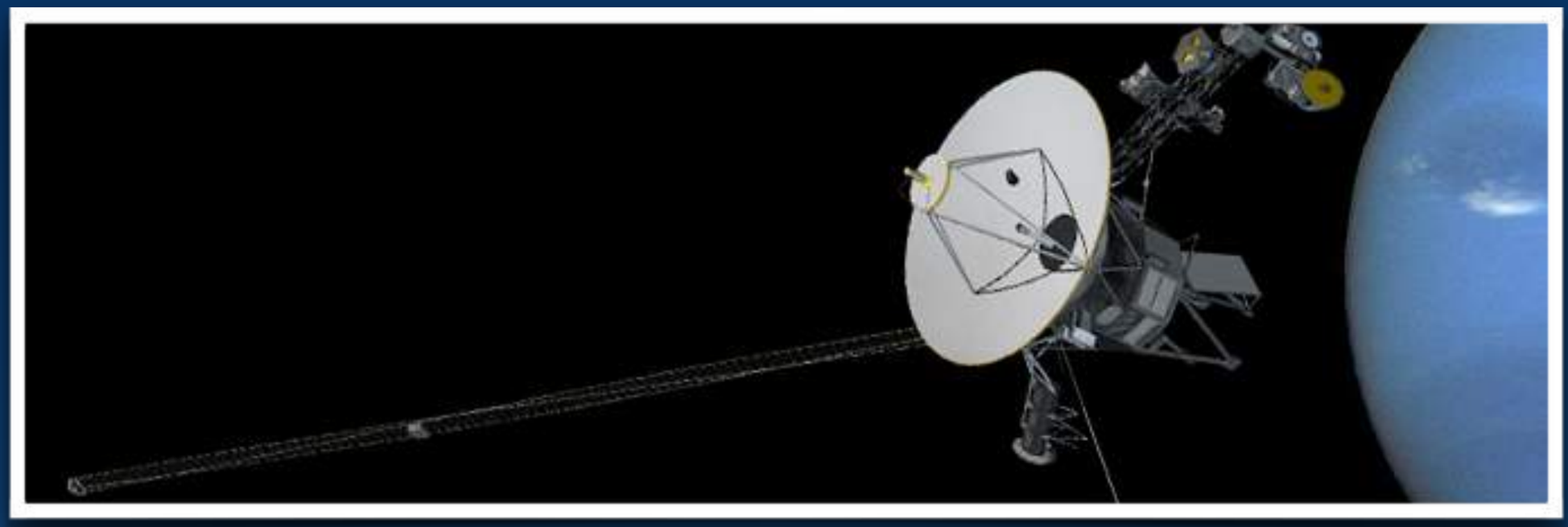
Software in Various Disciplines

Free SMS!

Technologies : Java, Objective-C, etc.

http://kakao.com

Software in Various Disciplines (cont.)

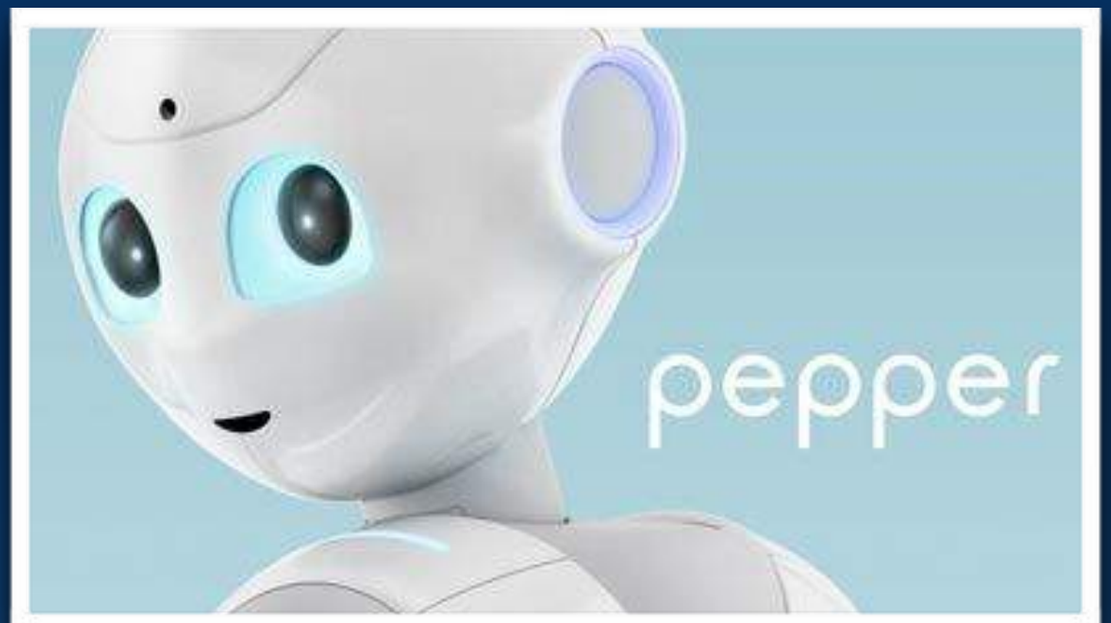


NASA Voyager : Explore more than solar system.

https://www.nasa.gov/mission_pages/voyager/index.html

Technologies : Assembly, etc.

Software in Various Disciplines (cont.)

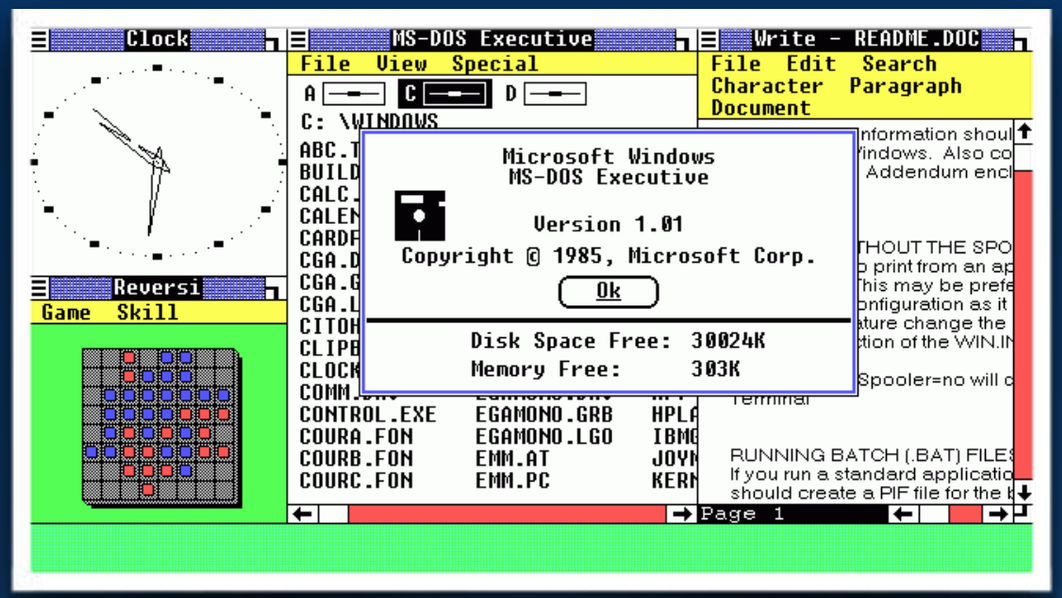


SoftBank Pepper : More freedom to aged people.

http://www.therobotreport.com/cache/uploads/Pepper-1_400_225_80.jpg
http://www.geeksuper.com/wp-content/uploads/2014/06/pepper-robot.jpg

Technologies : ROS (Robot Operating System), etc.

Software in Various Disciplines (cont.)



Microsoft Windows : An graphical operating system

https://en.wikipedia.org/wiki/Microsoft_Windows#/media/File:Windows_8.1_Start_screen.jpg

Technologies : C, C++, C#, Assembly, etc.

Required Techniques

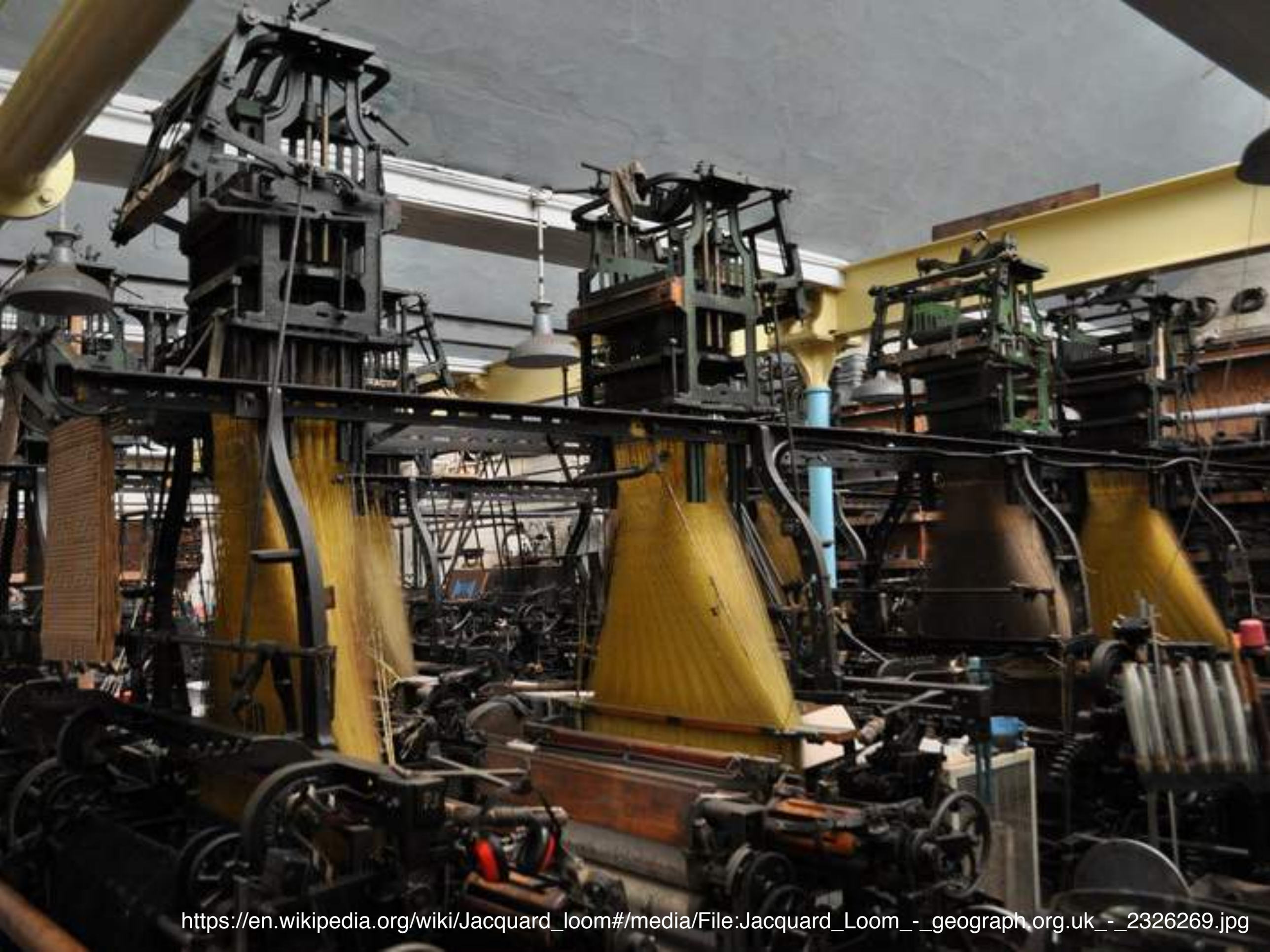
- Sense of discover real, critical problems
- An ability to implement a solution
- Managing group of people toward to achieve the goal

http://haml.info/

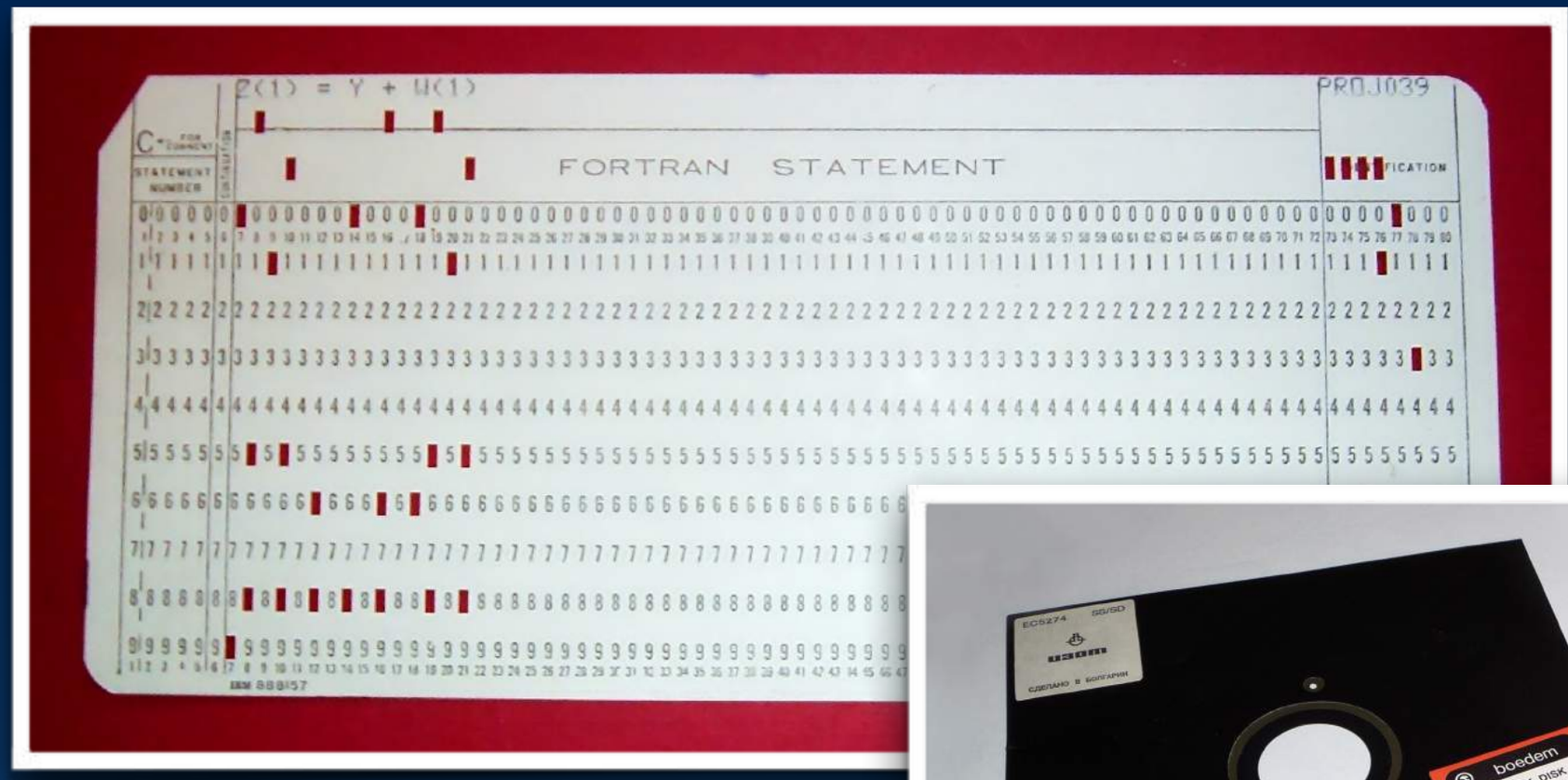
Programming Language

Programming language is a formal constructed language designed to communicate instructions to a machine - via Wikipedia

https://en.wikipedia.org/wiki/Textile#/media/File:Otavalo_Artisan_Market_-_Andes_Mountains_-_South_America_-_photograph_001.JPG



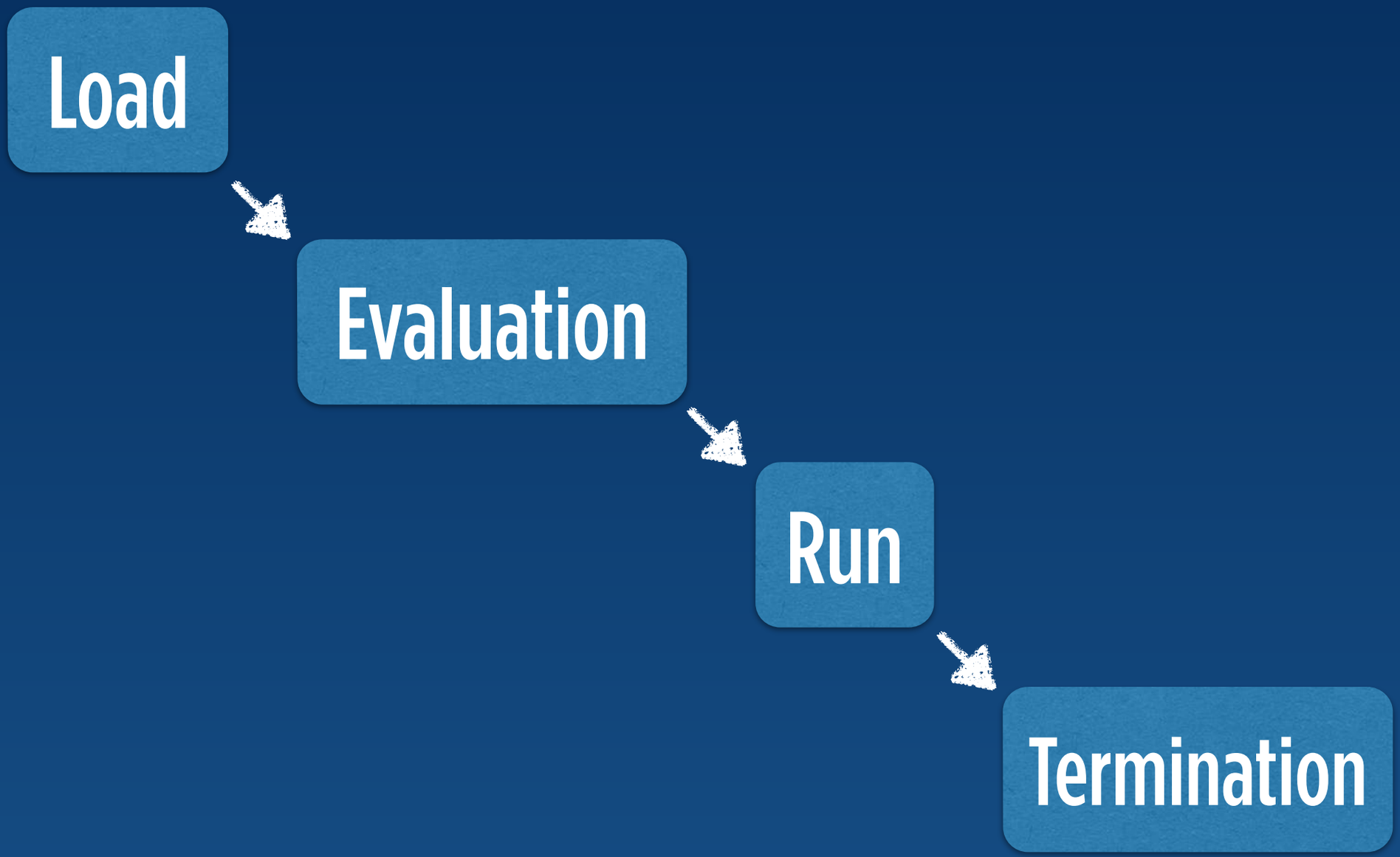
https://en.wikipedia.org/wiki/Jacquard_loom#/media/File:Jacquard_Loom_-_geograph.org.uk_-_2326269.jpg


Z(1) = Y + W(1)
PRO.1039
C
STATEMENT NUMBER
CONTINUATION
FORTRAN STATEMENT
IDENTIFICATION

EC5274 SS/SD
СДЕЛАНО В БОЛГАРИЯ
boedem
FLOPPY DISK
5,25" HD,DS
WRITE ENABLE
WRITE PROTECT

GT100

Flow of Execution

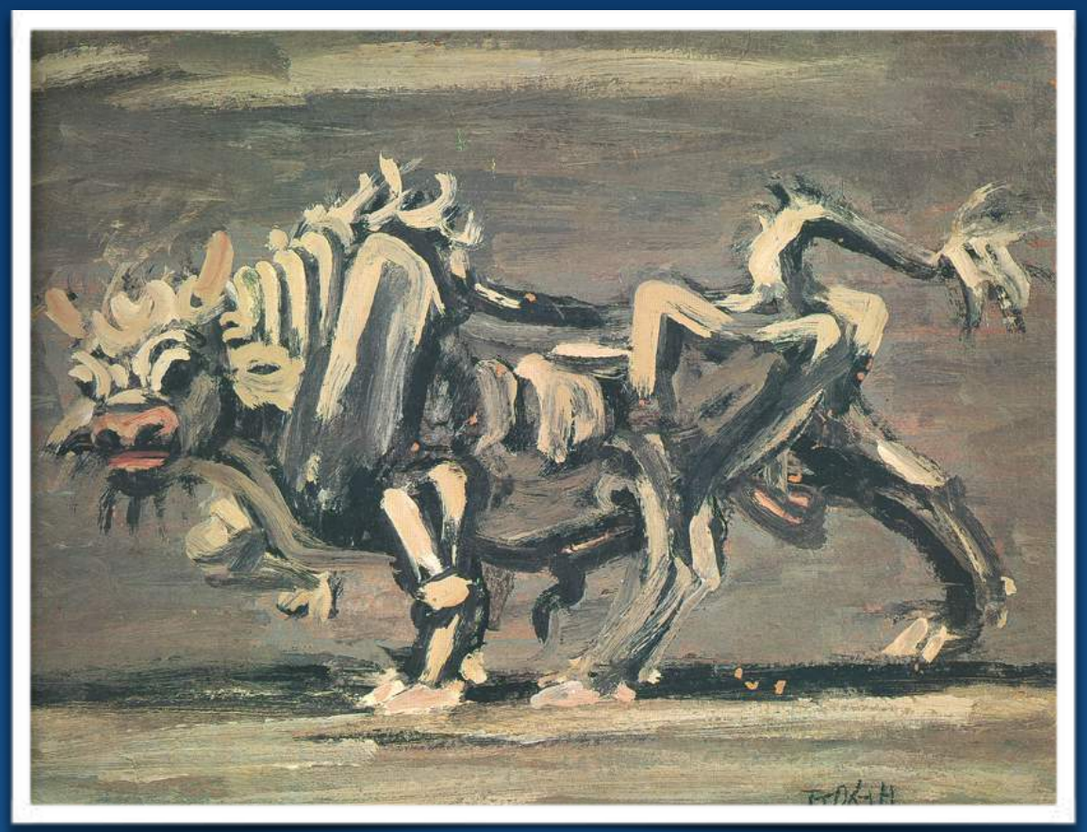


Exercise #1

Programing on Node.js REPL (Read-Eval-Print-Loop)

Abstraction

Technique for managing complexity of computer systems - via Wikipedia

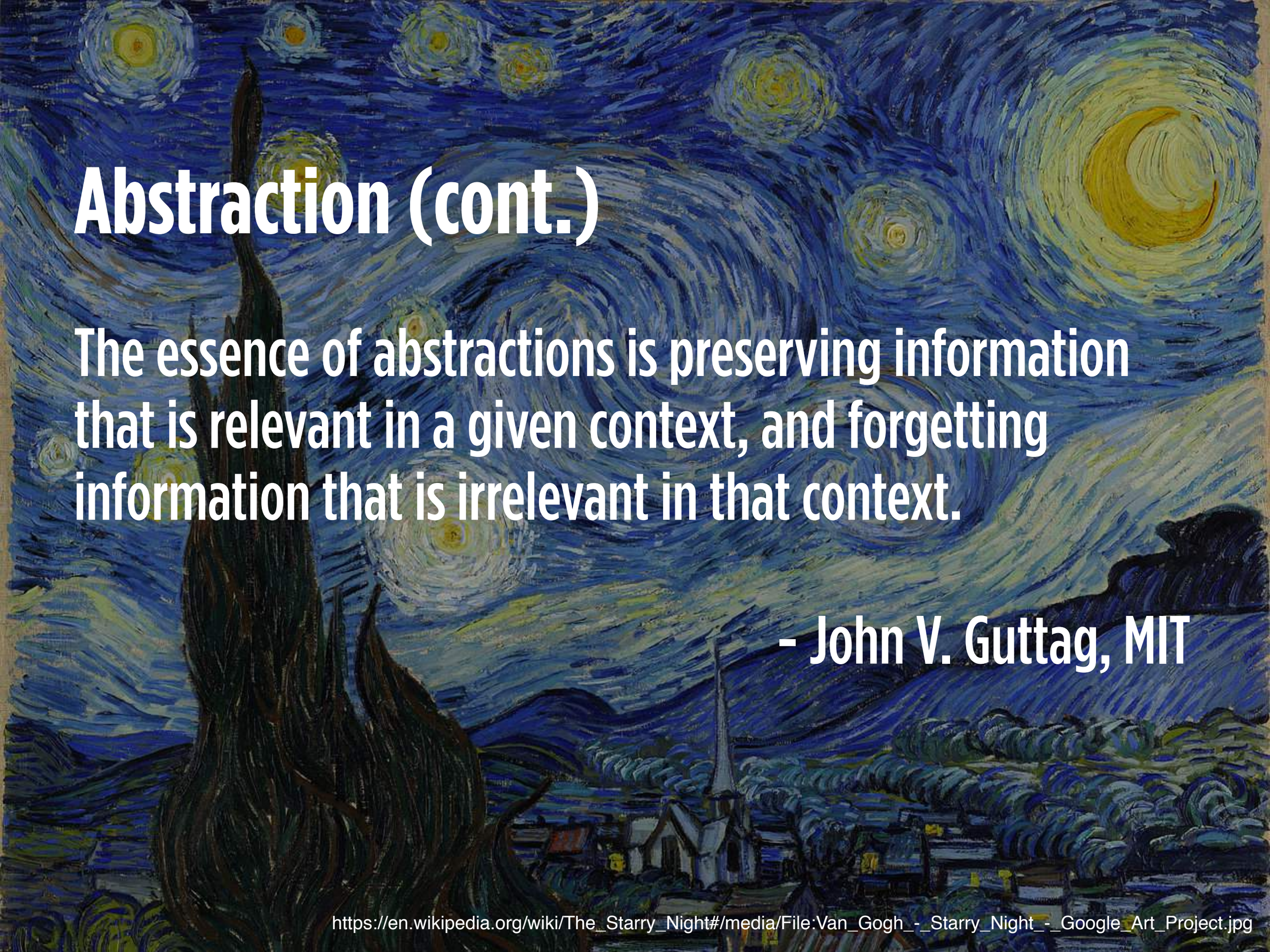


https://en.wikipedia.org/wiki/Lee_Jung-seob#/media/File:White_Ox_(1954)_-_Lee_Jung_Seob.jpg

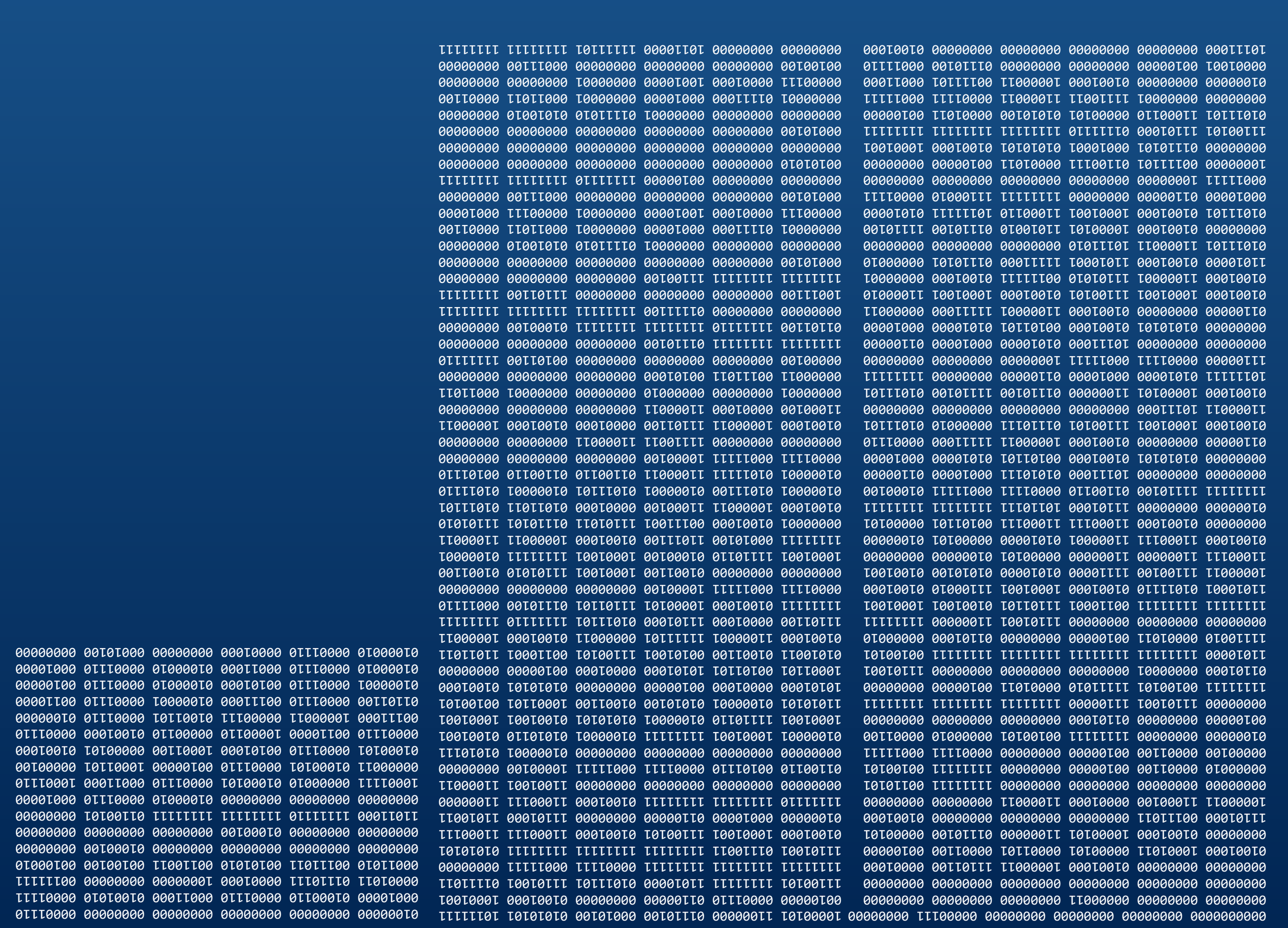
Abstraction (cont.)

The essence of abstractions is preserving information that is relevant in a given context, and forgetting information that is irrelevant in that context.

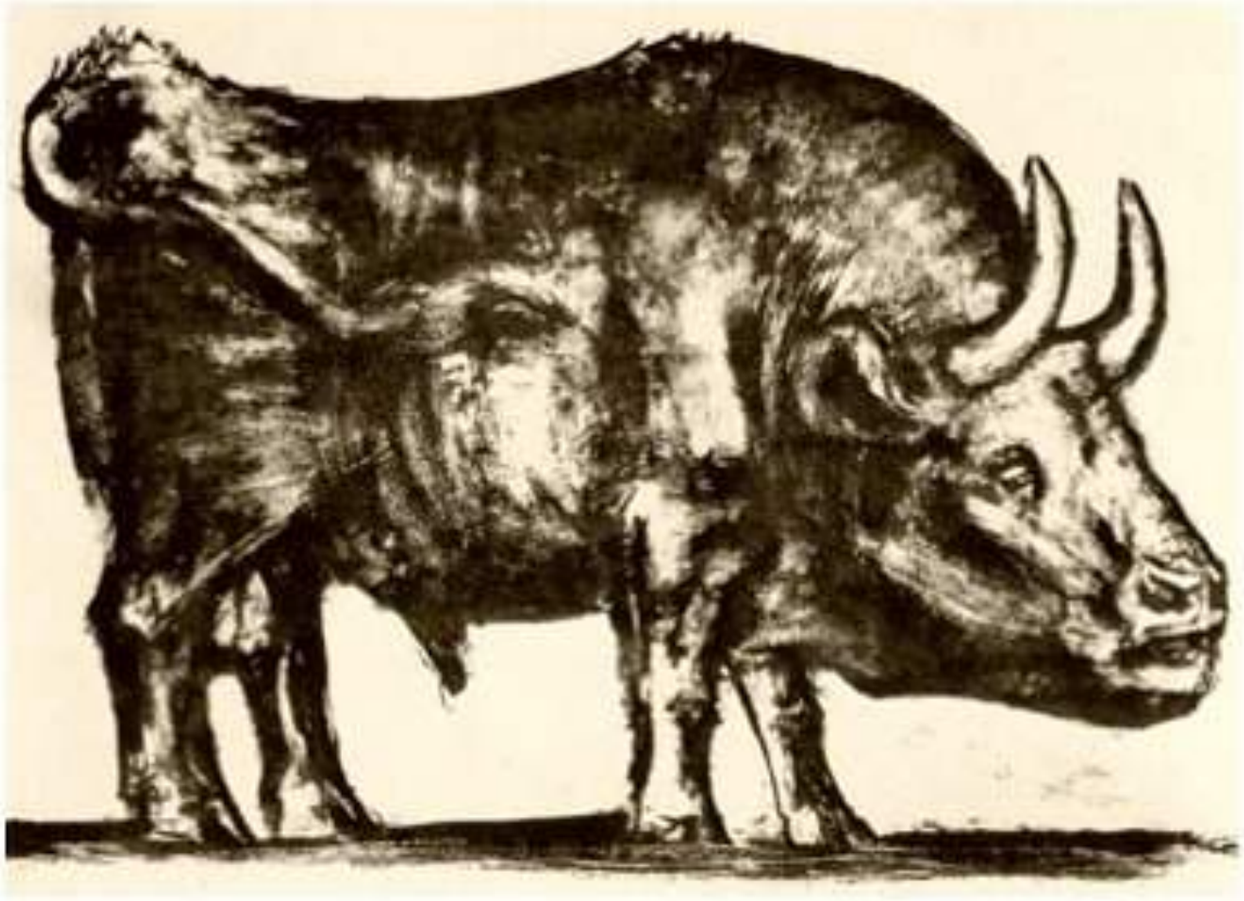
– John V. Guttag, MIT



https://en.wikipedia.org/wiki/The_Starry_Night#/media/File:Van_Gogh_-_Starry_Night_-_Google_Art_Project.jpg



```
global _start

section .text
_start:
   mov  eax, 4
   mov  ebx, 1
   mov  ecx, msg
   mov  edx, msg.len
   int  0x80

   mov  eax, 1
   mov  ebx, 0
   int  0x80

section .data
msg: db "Hello, world!", 10
.len:   equ  $ - msg
```



Pablo Picasso - ‘Bull - plate 2’
(December 12, 1945)

Assembly

Programmer need to handle atomic CPU instructions.

Lot’s of boilerplate with control flow, set, get and others compare to high level languages like C.

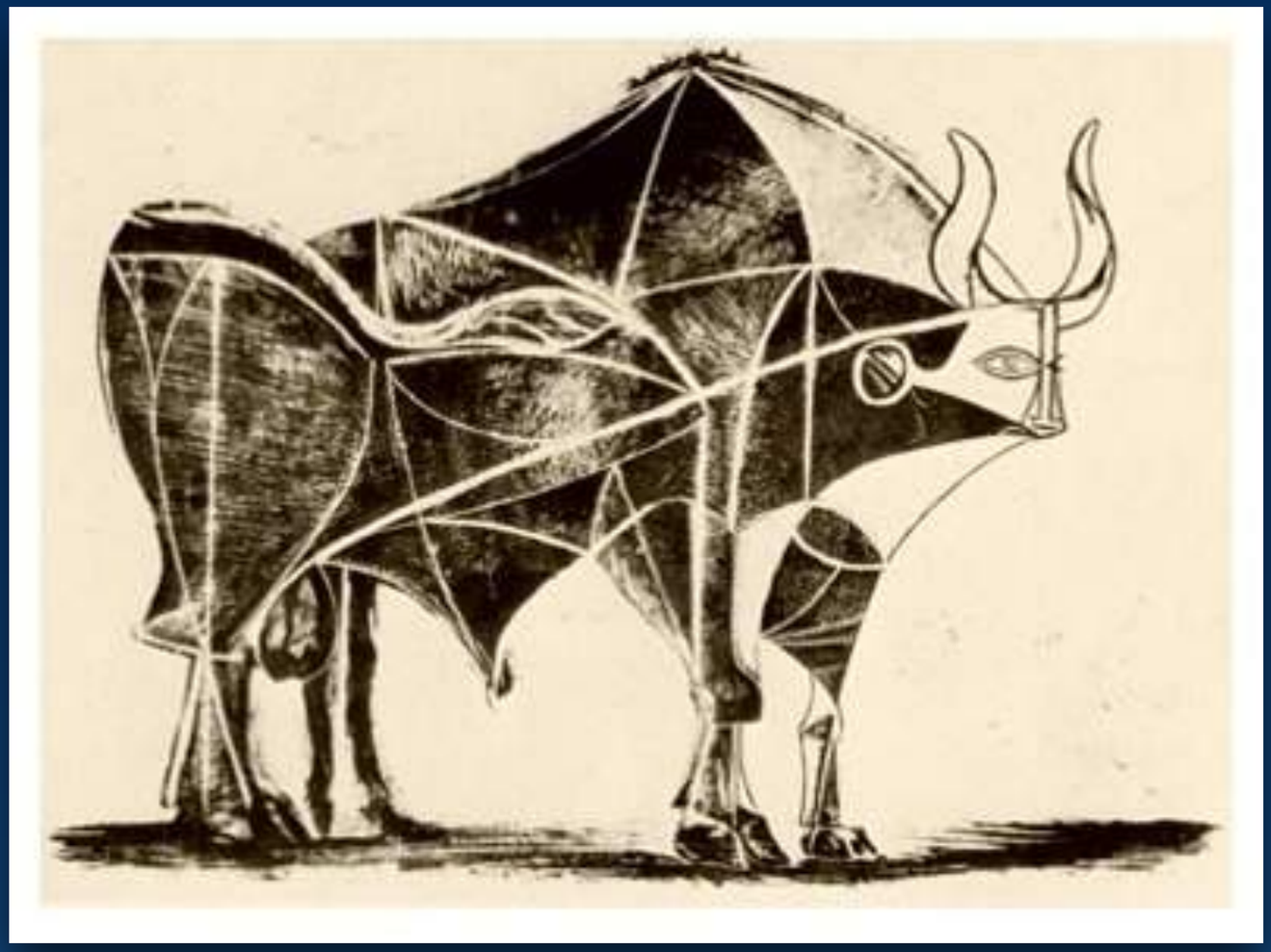
```c
#include <stdio.h>

int main () {
   printf("Hello World");
}
```

C

Need to handle memory addresses and their usages.

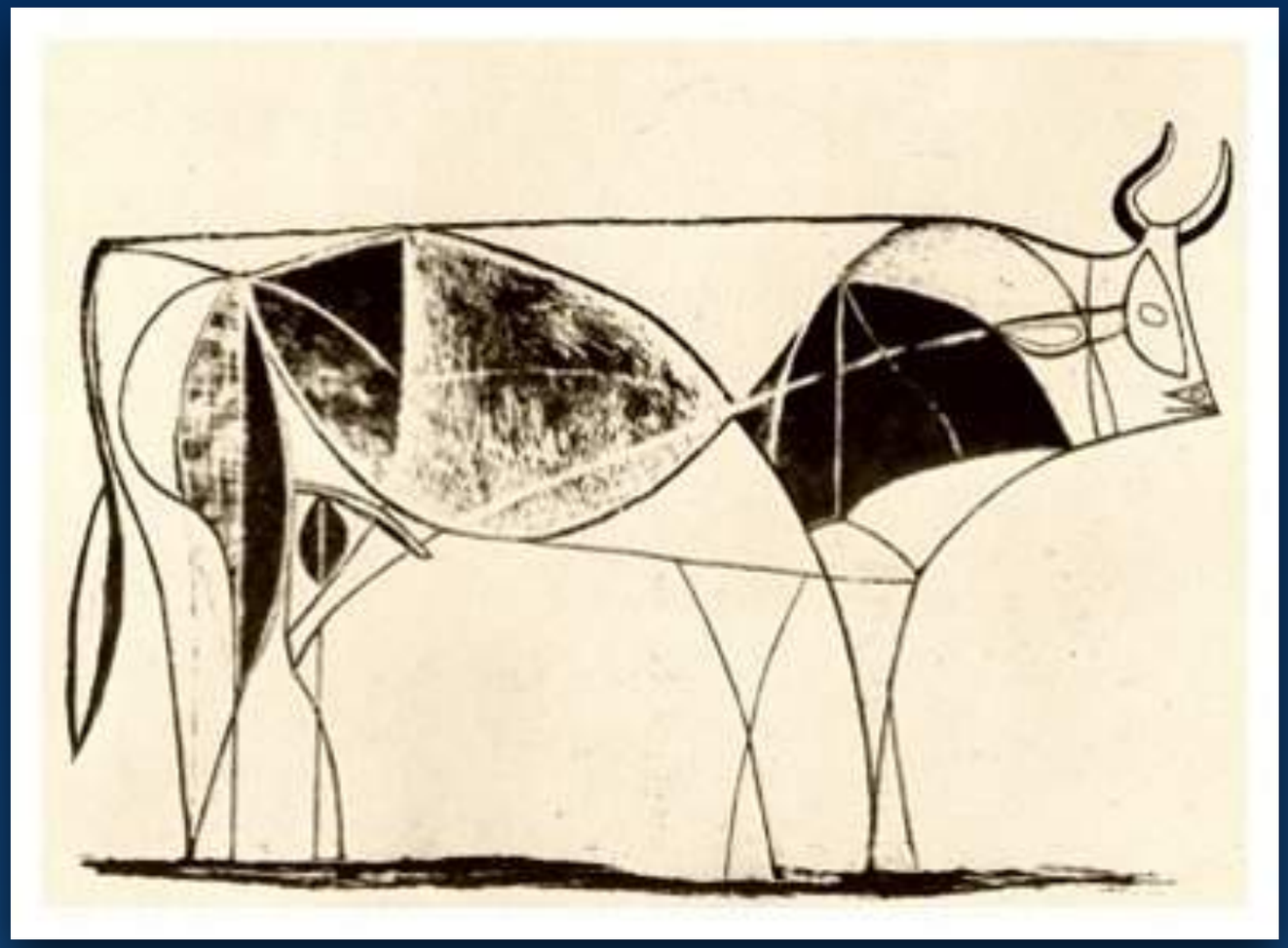
Significantly abstractor then assembly.

Still boilerplates are exsiting compare to more high level languages – like python.



Pablo Picasso – ‘Bull – plate 5’
(December 24, 1945)

```
print "Hello World"
```



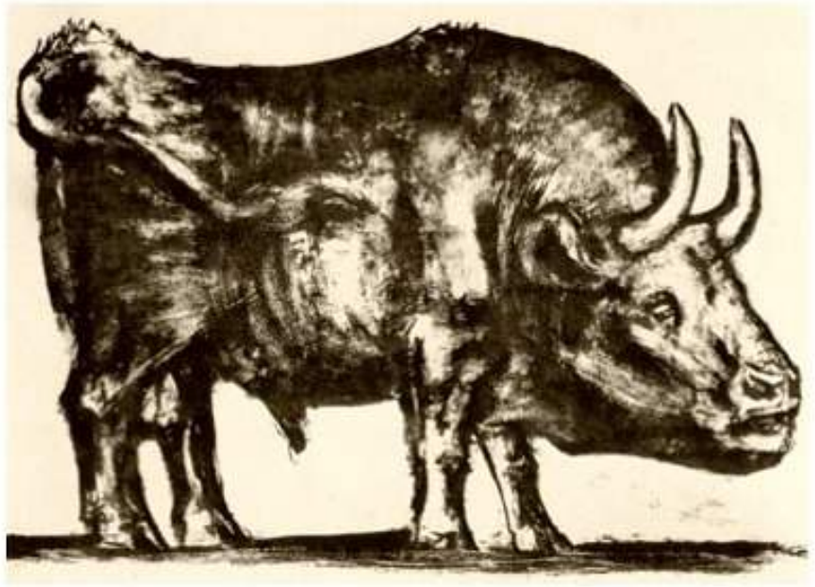
Pablo Picasso - ‘Bull - plate 8’
(January 1, 1946)

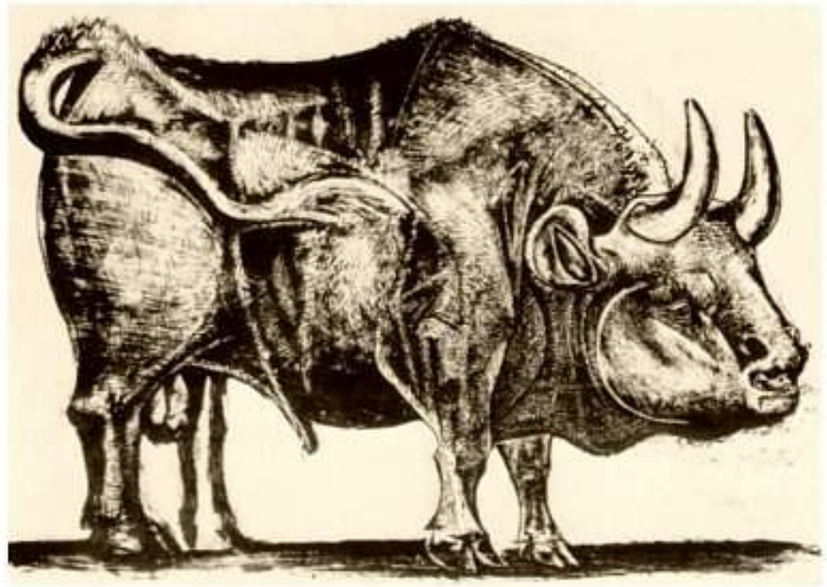
Python

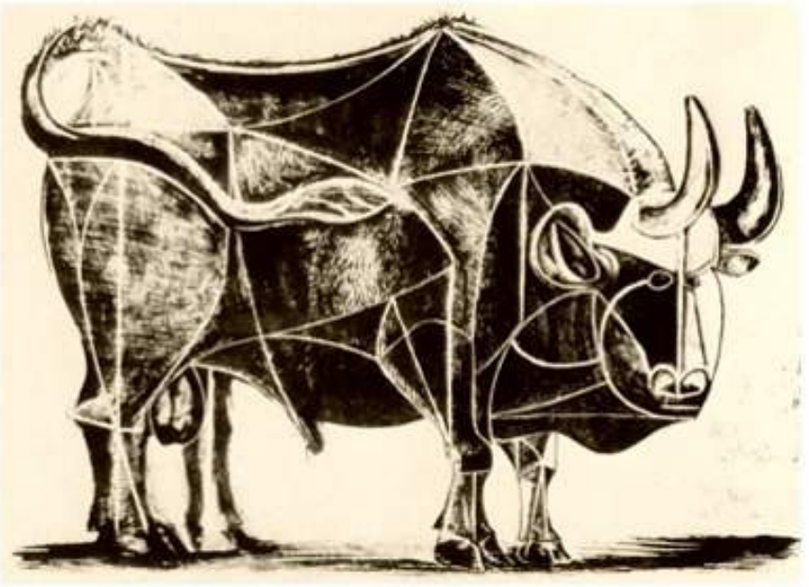
Programmer dosen’t need to handle raw memory addresses even ‘free’ them.

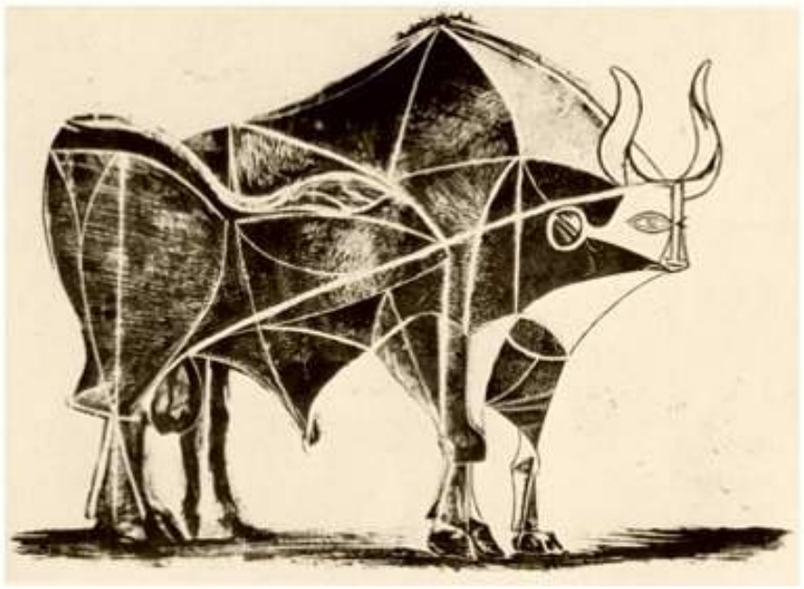
They don’t have to use basic boilerplate code.

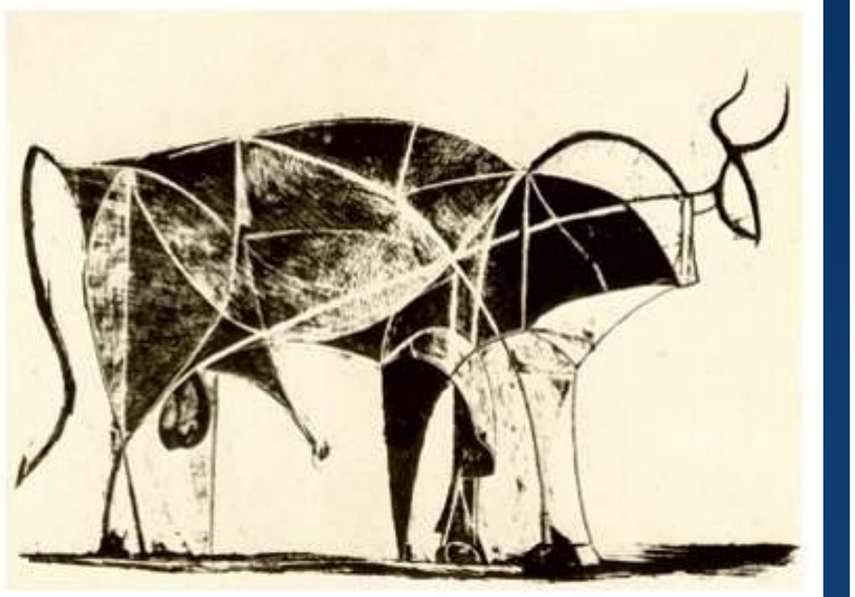
Some of boilerplates are still exisiting for using framework.

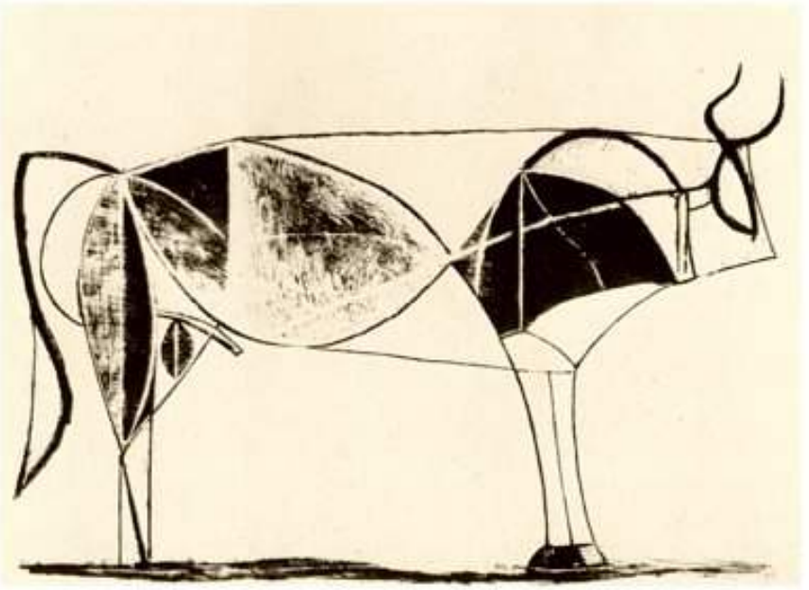


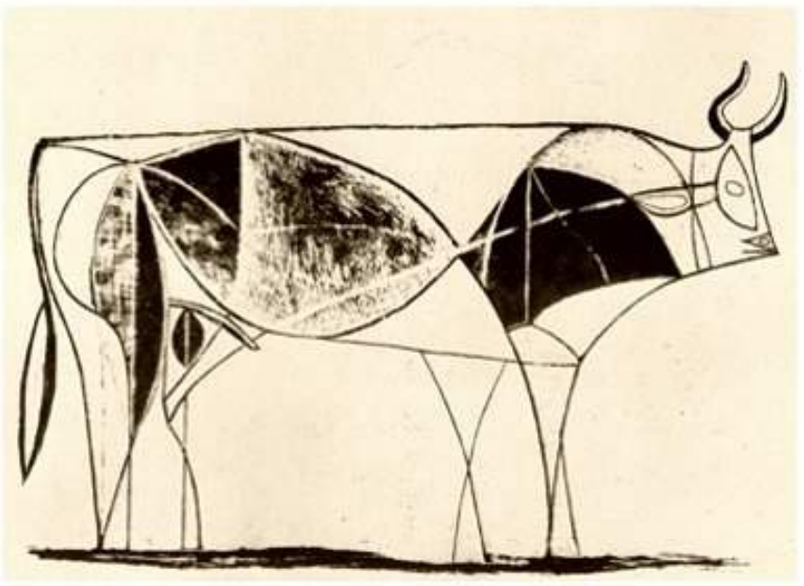


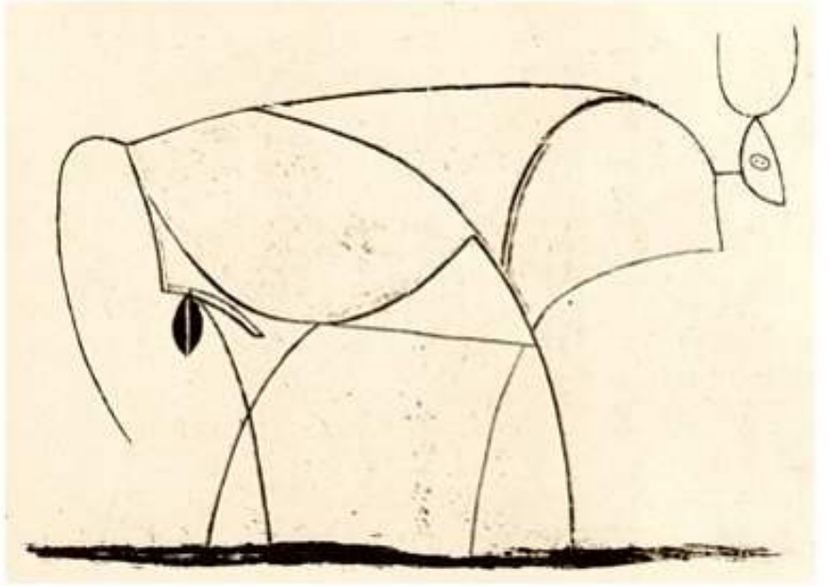










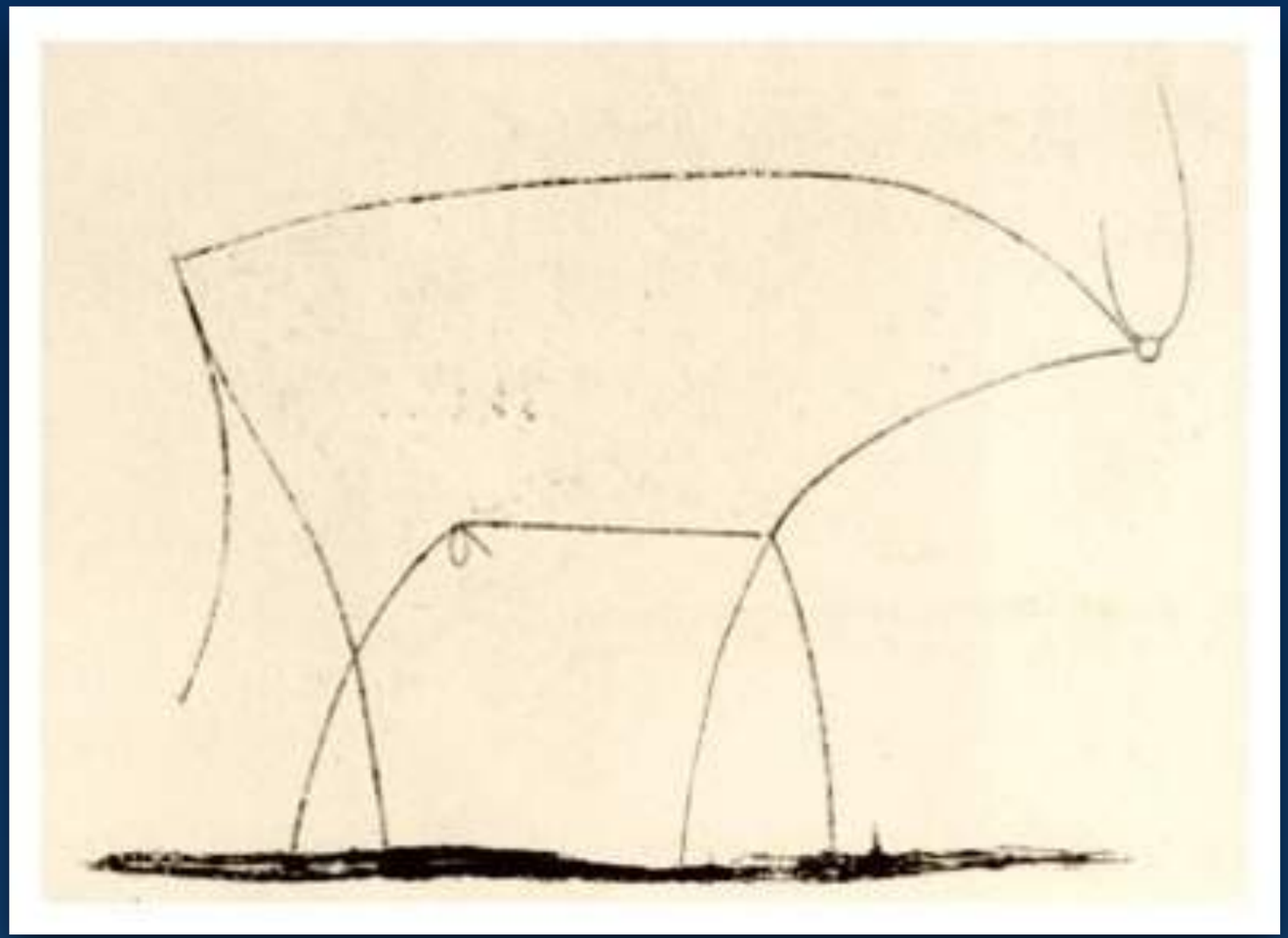


?

Future?

Transformation of General Purpose Languages to Domain Specific Language.

Computer's Prediction of programmer's mind.

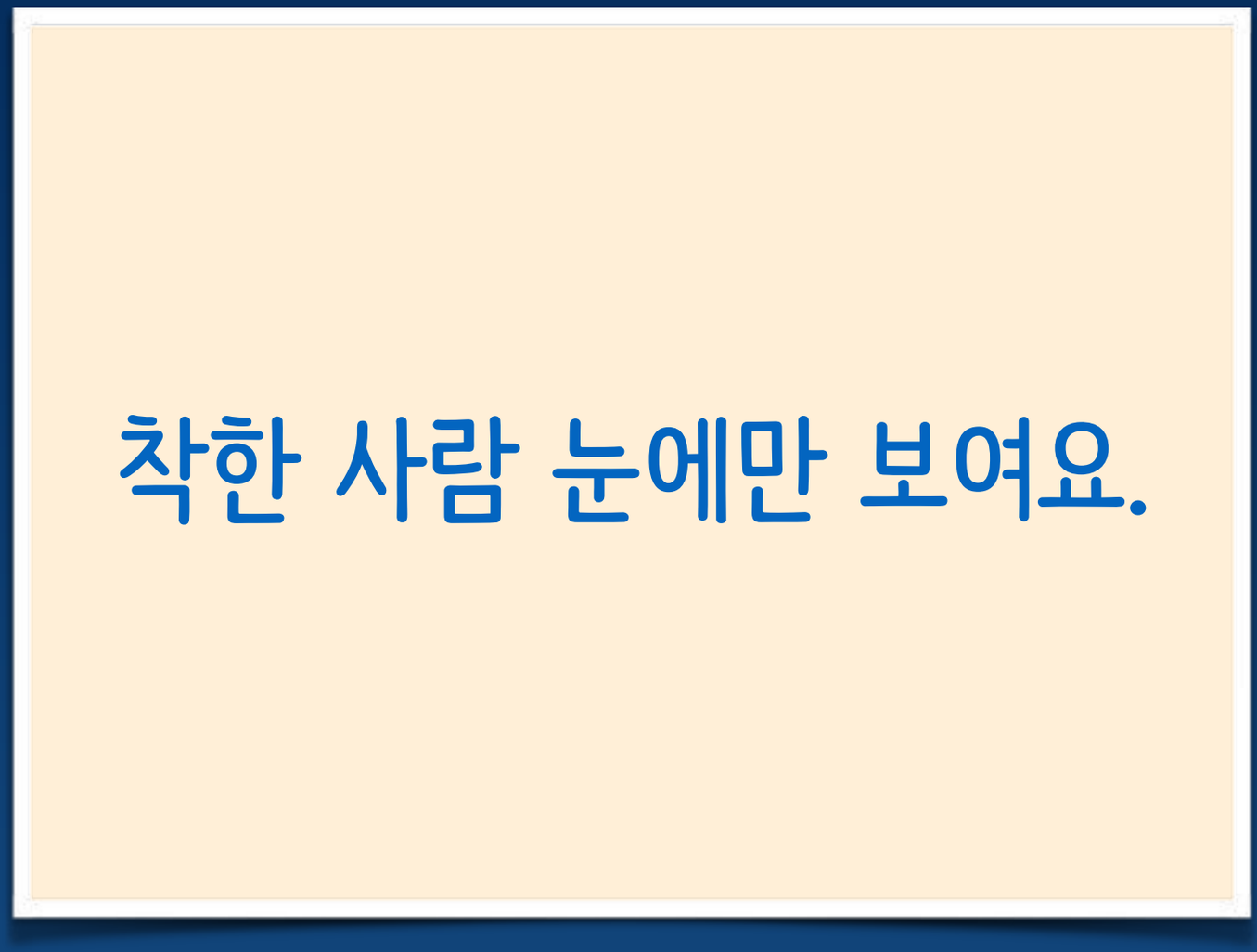


Pablo Picasso - 'Bull - plate 11'
(January 17, 1946)

Future?

Programming Language Corresponding to Natural Language.

Who Know?
(Lots of Possibilities)



Hiun Kim - 'Bull - plate 14'
(July 7, 2015)

Dual Edged Sword

Detailed Instruction

Fast

lower advantages

Weight of Abstraction

Productivity

higher advantages

Atomic Debuggablility

Design

Domain Specific

Value of Abstraction

Wasting programmer time is the true inefficiency, not wasting machine time. This will become ever more clear as computers get faster.

...

There's good waste, and bad waste. I'm interested in good waste—the kind where, by spending more, we can get simpler designs.

– Paul Graham, Y Combinator

Machine Language

```
001 01010101 01001000 10001001
110 11111111 11111111 11111111
101 0                 00000
011 1                 11111
000 1                 11000
000 0                 11110
000 0                 01000
100 00010100 01010101 10111111
000 00000000 01001000 10001001
000 01011101 11101001 01111011
111 00001111 00011111 00000000
111 11111111 11111111 01010101
101 01001000 11000111 11000111
000 00000000 11101000 11010011
111 01001000 11000111 11000000
000 00000000 11001001 11000011
111 00011111 10000100 00000000
000 00000000 01000001 01010111
111 01000001 01010110 01001001
001 01010101 01001001 10001001
100 01001100 10001101 00100101
000 00000000 01010101 01001000
000 00001000 00100000 00000000
001 11100101 00110001 11011011
101 00000011 01001000 10000011
000 01011101 11111110 11111111
101 11101101 01110100 00011110
100 00000000 00000000 00000000
100 10001001 11101010 01001100
100 10001001 11111111 01000001
100 01001000 10000011 11000011
001 11101011 01110101 11101010
100 00001000 01011011 01011101
001 01011101 01000001 01011110
011 01100110 01100110 00101110
100 00000000 00000000 00000000
011 11000011 00000000 00000000
100 00001000 01001000 10000011
011 00000000 00000000 00000000
010 00000000 00000001 00011011
000 00000000 00000000 00000000
000 00000000 00101100 11111110
100 00000000 00000000 00000000
```

Assembly

```
global _start

section .text
_start:
      mov   eax, 4
      mov   ebx, 1
      mov   ecx, msg
      mov   edx, msg.len
      int   0x80

      mov   eax, 1
      mov   ebx, 0
      int   0x80

section .data
msg:  db    "Hello, world!", 10
.len: equ   $ - msg
```

C

```
#include <stdio.h>

int main () {
      printf("Hello World\n");
}
```

Python

```
print "Hello World\n"
```

Industry Usages

< C >

Embedded System, OS Kernel, Safety-critical system (ex. railway, automotive), etc.

< C++ >

Windows App, Game, Server (in financial companies), Audio / Image Processing, Device Driver, System Software (ex. Database), etc.

< Java >

Web Application Server, Desktop App, Large Data Processing, Android App, etc.

< Python >

Web Application Server, A.I related stuff, Software Testing / Deployment, etc.

< JavaScript >

Web Frontend, Web Application Server, etc.

(Internet Company Only, Technology stack at the beginnig of.)

Industry Usages (cont.)

~ 1995 Amazon (C++) PayPal (C++) Viaweb (Lisp) eBay (C++) C++

1996 ~ 2000 Google (Java, Python) Naver (Java) LinkedIn (Java) Java

2001 ~ 2005 Youtube (Python) Facebook (PHP) Flickr (PHP) Tumblr (PHP) PHP

2006 ~ 2010 Twitter (Ruby [ROR]) Disqus (Python [Django]) Dropbox (Python)

2011 ~ 2015 Kakao Talk (Ruby [ROR]) Instagram, Pinterest (Python [Django]) Python, Ruby

JavaScript?

One Perspective

Every moment in business happens only once. The next Bill Gates will not build an operating system. The next Larry Page or Sergey Brin won't make a search engine. And the next Mark Zuckerberg won't create a social network. If you are copying these guys, you aren't learning from them.

– Peter Thiel, Palantir Technologies

Filling Gap Between Real World

But, how can create practical software in this weel?

Filling Gap Between Real World (cont.)

< Native Modules >

Enables us to use file system, send netwoek packet, get memory information, etc.

https://en.wikipedia.org/wiki/Ethernet#/media/File:Ethernet_Connection.jpg

http://partition.radified.com/JPEGs/hard_disk_drive.jpg

Filling Gap Between Real World (cont.)

< System Softwares >

Database allowed us to store data in structured way. Cache helps us to serve more content faster. Web Server make possible to controll inbound traffics, etc.

Filling Gap Between Real World (cont.)

< Frameworks >

Driving us to more productive. Helps to follow best practices. More importantly make us to fosuing on things that truly matter.

Filling Gap Between Real World (cont.)

< Libraries >

Helps us to prevent to inventing a new wheel.

Exercise #2

Simple Web Application with Node.js and ExpressJS

The Future

?

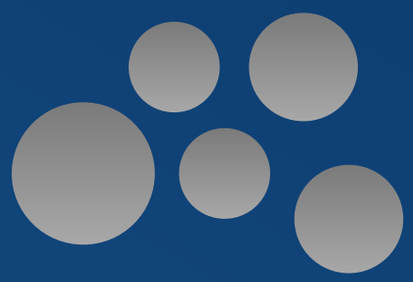
```
printf "Hello World"
```

```
#include <stdio.h>

int main () {
      printf("Hello World");
}
```

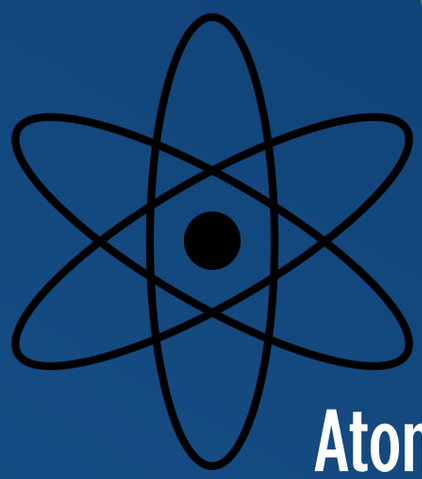
```
00000011 01000101 00001110 00100000 10001101 00000100
01000101 00001110 00101000 10001100 00000101 01001000
00001110 00110000 10000110 00000110 01001000 00001110
00111000 10000011 00000111 01001101 00001110 01000000
01101100 00001110 00111000 01000001 00001110 00110000
```

Software

Transister



Silicon



Atom

The Inheritances, the responsibilities.

http://www.jennyhouse.info/board/data/file/seoul/741763354_38906cff_3243.jpg

Thank You!

Questions or Opinions - hiun@divtag.sejong.edu

Slides are available - www.hiunkim.com