

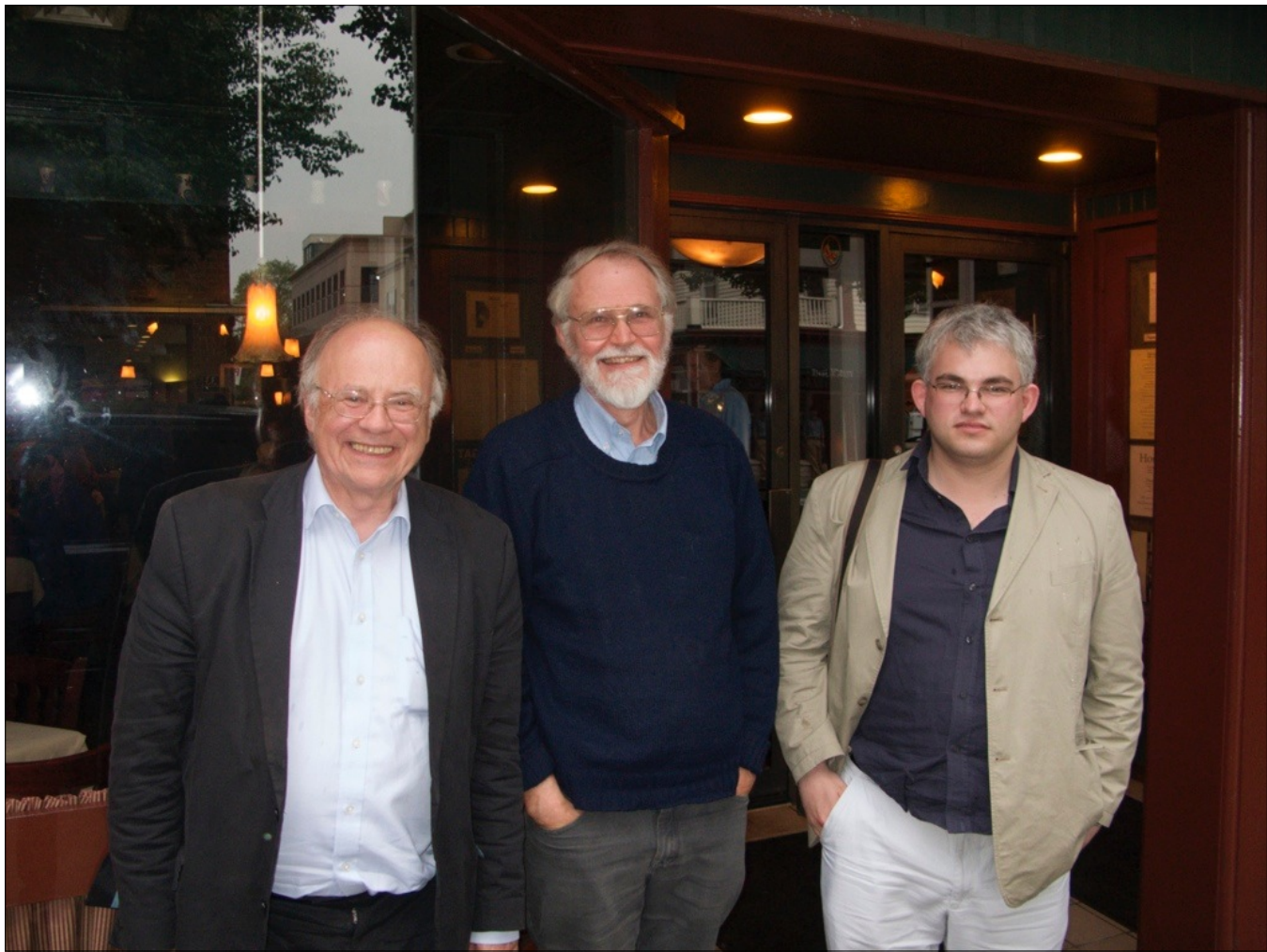
```
label DEFB "G51CSA", 10, 0  
ADR R0, label  
SWI 3  
MOV PC, R14
```

```
printf( "Programming" );
```

Steven R. Bagley



- G5IPRG — Programming (using C)
- G5ICSA — Computer Systems Architecture
- Co-requisites, will effectively be taught as co-courses



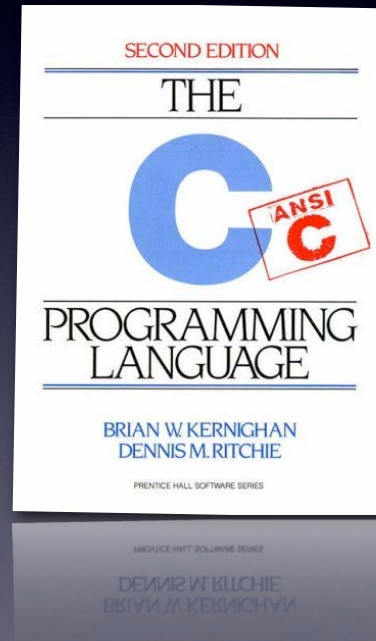




Steve Bagley
srb@cs.nott.ac.uk
C55

Brian Kernighan
Author of *The C
Programming Language*

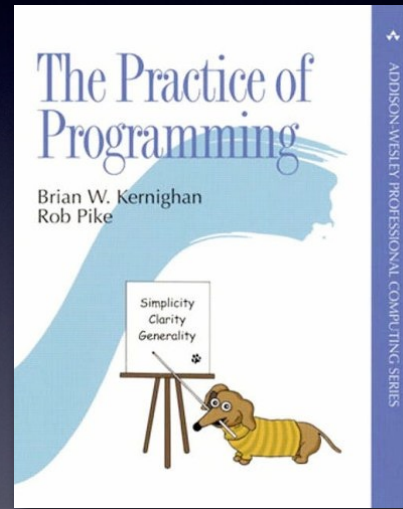
Book



- *The C Programming Language* by Brian Kernighan and Dennis Ritchie
- ISBN: 978-0131103627
- £20.17 from Amazon at the moment (Kindle)

If you only get one book, get the C Programming Language one
This one much better for getting a feel for how to program bigger systems

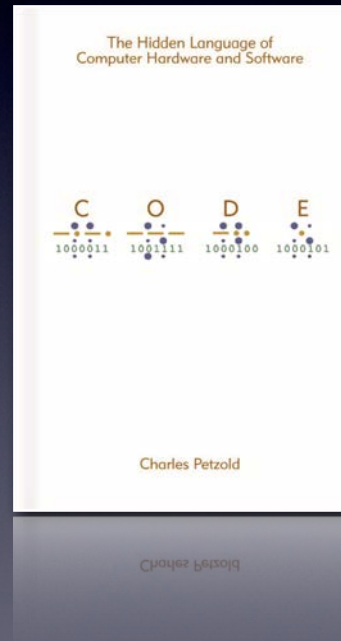
Book



- *The Practice of Programming* by Brian Kernighan and Rob Pike
- ISBN: 978-0201615869
- £19.48 from Amazon at the moment (Kindle)

If you only get one book, get the C Programming Language one
This one much better for getting a feel for how to program bigger systems

Book



- *Code*
by Charles Petzold
- ISBN: 978-0735611313
- £5.56 from Amazon at the moment (Kindle)

If you only get one book, get the C Programming Language one
This one much better for getting a feel for how to program bigger systems

NAND 2 TETRIS

- Will be making some use of the *nand2tetris* material
www.nand2tetris.org
- Worth looking at the free book chapters on there

Will set some of them as reading...

Notes

- On Moodle...
- Notes will also be on the G5IPRG website
<http://g51prg.cs.nott.ac.uk/>
- G5ICSA notes will also be at
<http://g51csa.cs.nott.ac.uk/>

Lectures

Monday 12:00 (LT2)

Thursday 09:00 (LT2)

Thursday 10:00 (LT2)

G5IPRG

Lectures

Monday 12:00 (LT2)

Thursday 09:00 (LT2)

Thursday 10:00 (LT2)

G5IPRG

Monday 10:00 (LT3)

Tuesday 09:00 (B52)

G5ICSA

Labs

Tuesday 14:00 — 16:00

Thursday 11:00 — 12:00

Friday 16:00 — 17:00

All in A32

Will start on Friday — another session to get experience with compiling software

Preceptorials

- 12 timetabled tutorial slots
- Sign up sheets for each slot
- Once full, its full
- *Please* let people with constrained timeslots sign up first

Will happen later this week -- watch your email for details...

Lecture Notes

- Notes will be placed online
- Purely what is presented on screen
- You are expected to attend lectures
- Take notes if we say things that aren't on the screen

G5 I PRG Assessment

- Purely by four pieces of coursework
- Week long exercises
- Coursework will be set on a Wednesday
- Due in the following Wednesday
- Start simple — ramp up in difficulty

Assessment Dates

Coursework One	8th — 16th October
Coursework Two	22nd — 29th October
Coursework Three	5th — 9th November
Coursework Four	<i>Over Christmas...</i>

Provisional Calendar...

Practice Exercises

- Also provide some 'fun' practice exercises
- Not assessed
- But can ask for help with them in labs...
- Strongly encouraged to do them

Practice... Practice... Practice...

G5 I CSA Assessment

- 50% Exam
- 50% Coursework
 - 50% Lab Exercises
 - 50% Assembly programming exercise

Introduction

- G5IPRG — Programming the computer to do what we want it to do
- G5ICSA — Understanding how the computer does what we want it to do

Abstractions

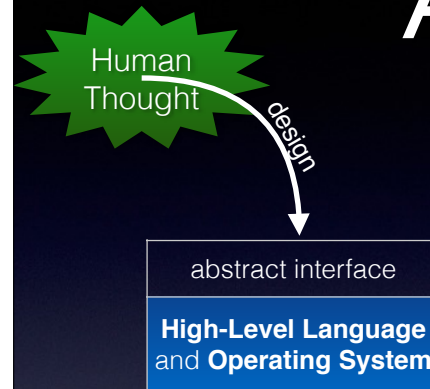
- Computers are built on abstractions
- Allow us to avoid thinking about the underlying implementation
- Most of the time...

Abstractions

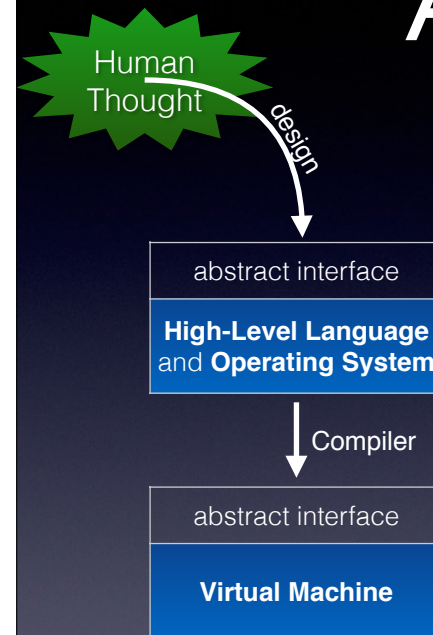
Abstractions



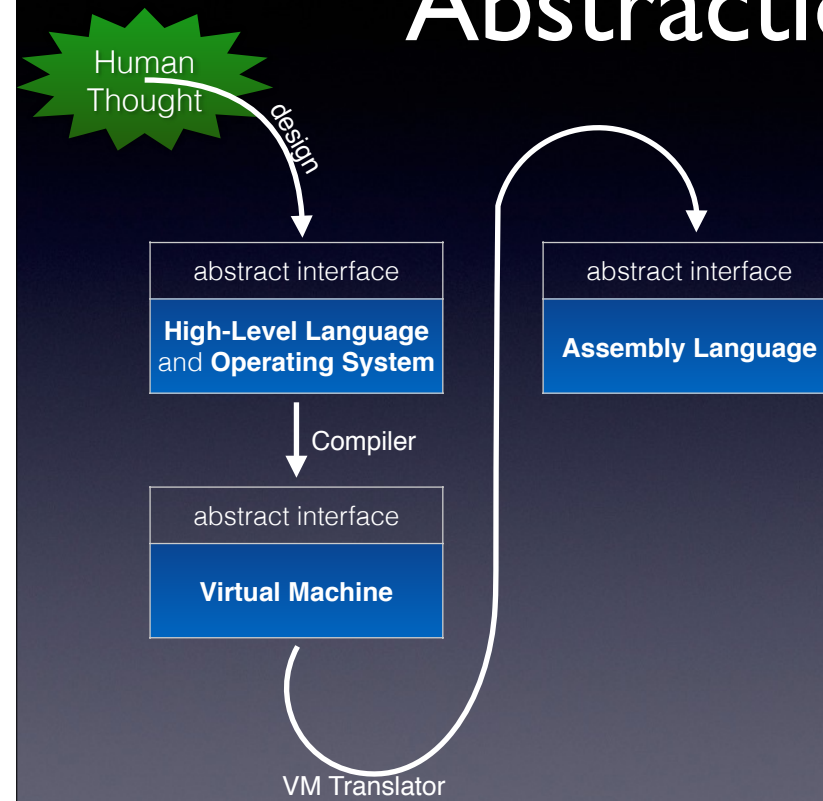
Abstractions



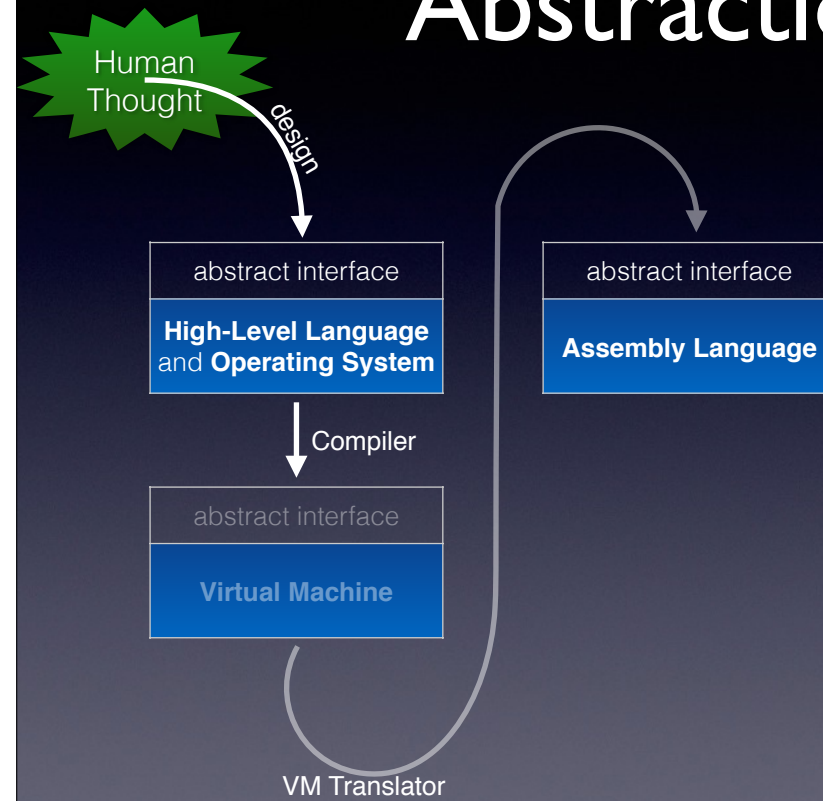
Abstractions



Abstractions



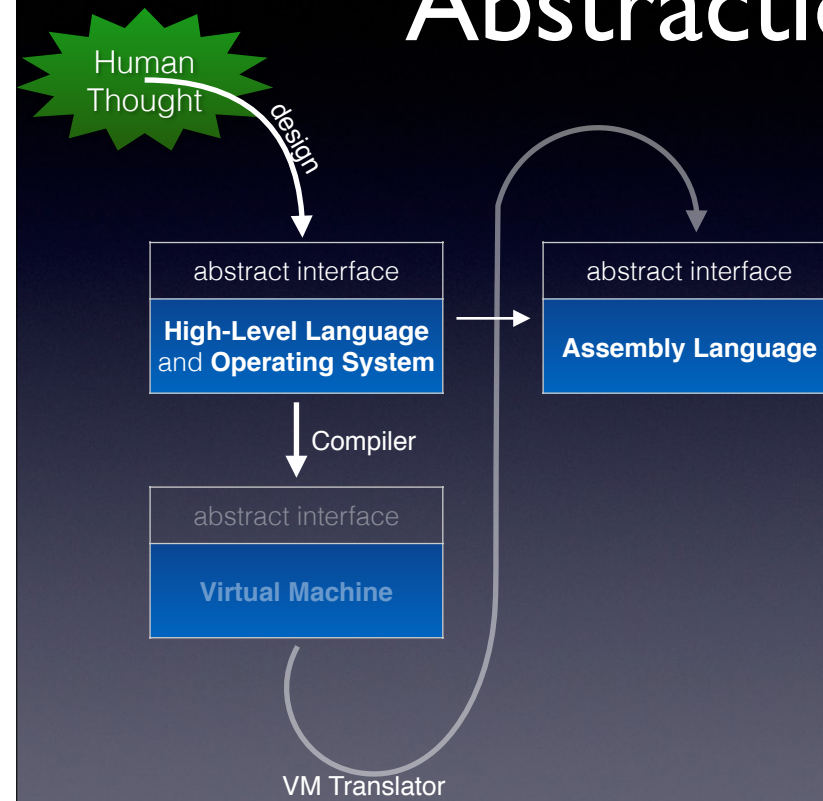
Abstractions



LD A, #65
RST 16
RET

D7
C9

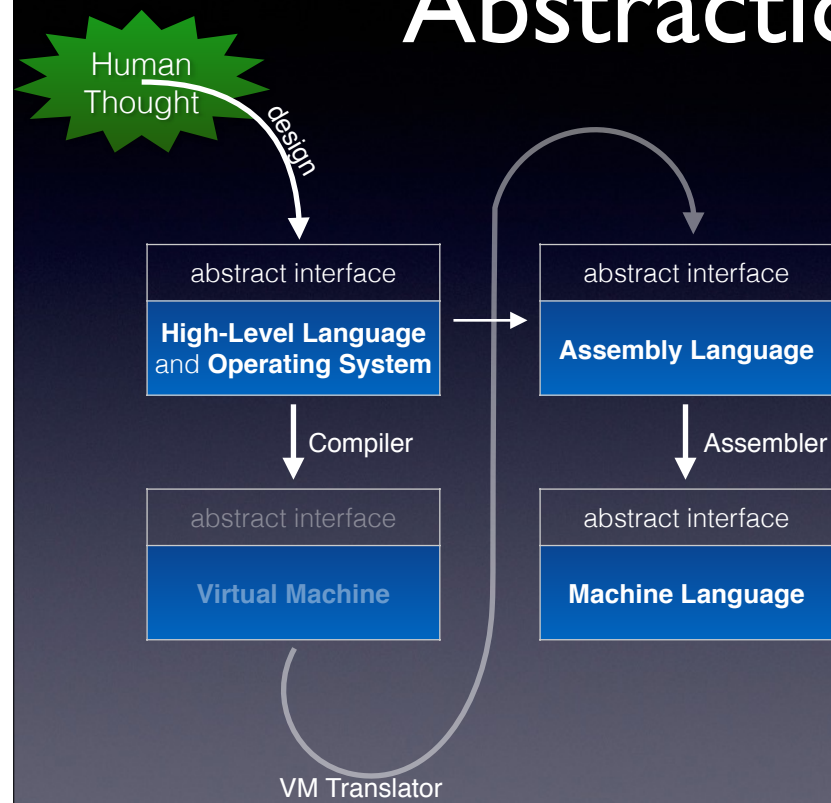
Abstractions



LD A, #65
RST 16
RET

D7
C9

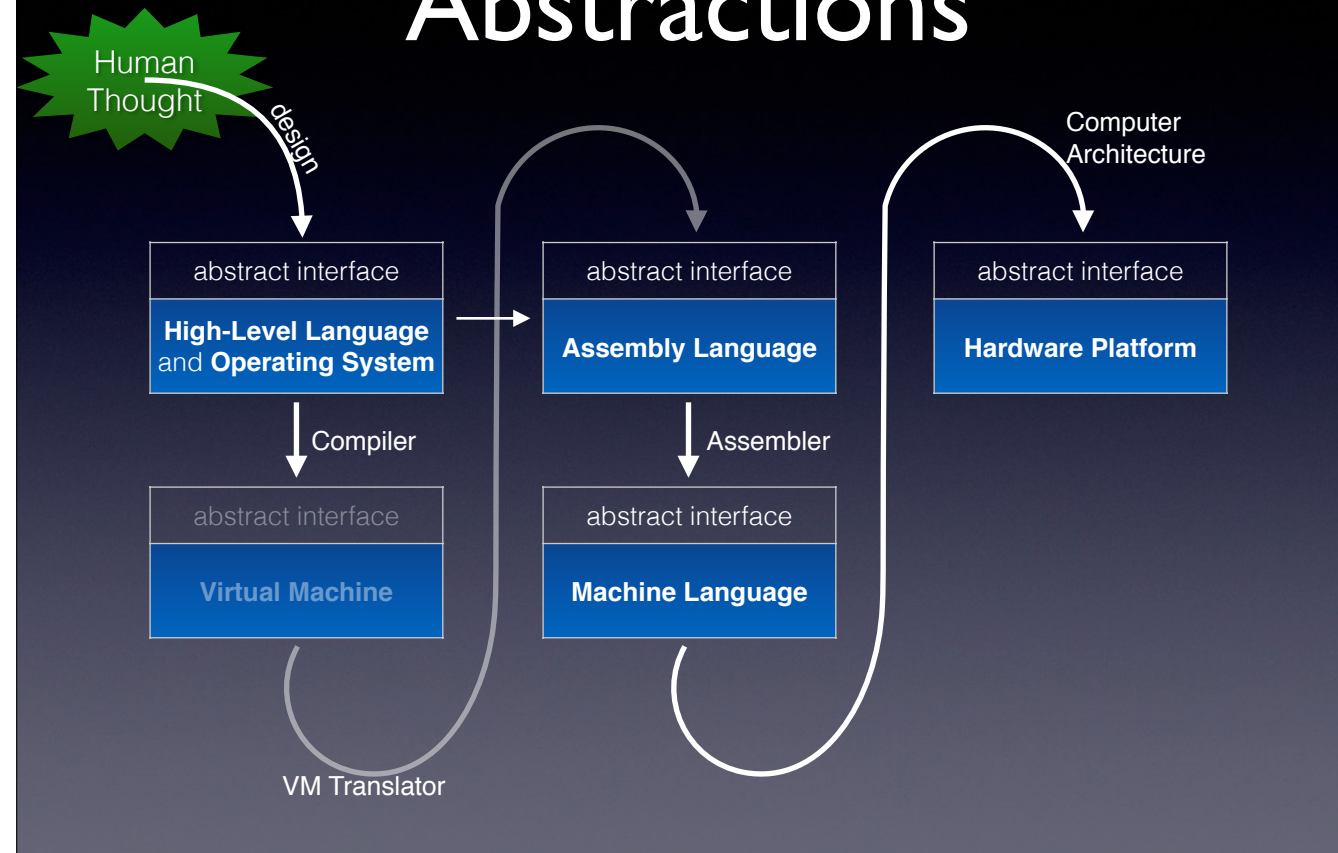
Abstractions



LD A, #65
RST 16
RET

D7
C9

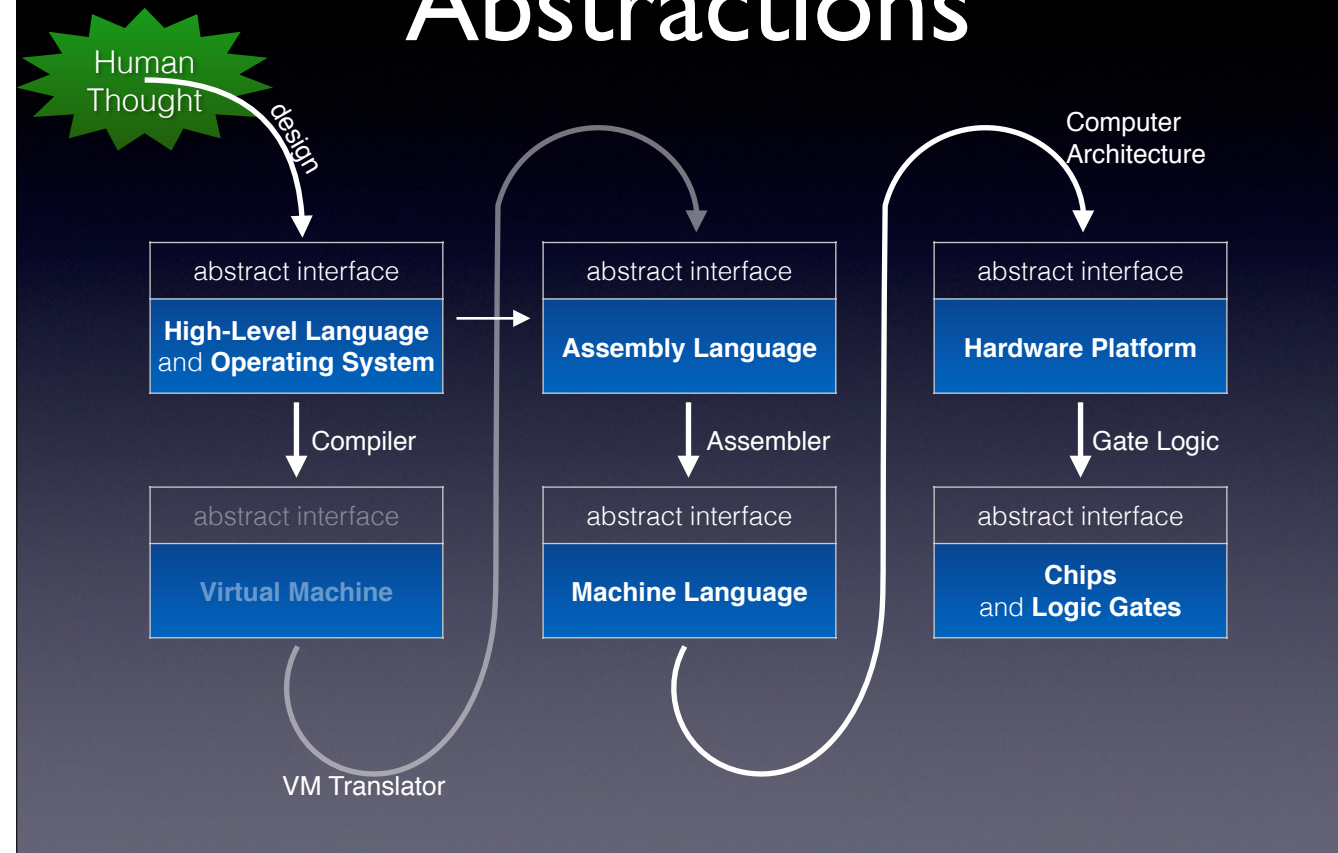
Abstractions



LD A, #65
RST 16
RET

D7
C9

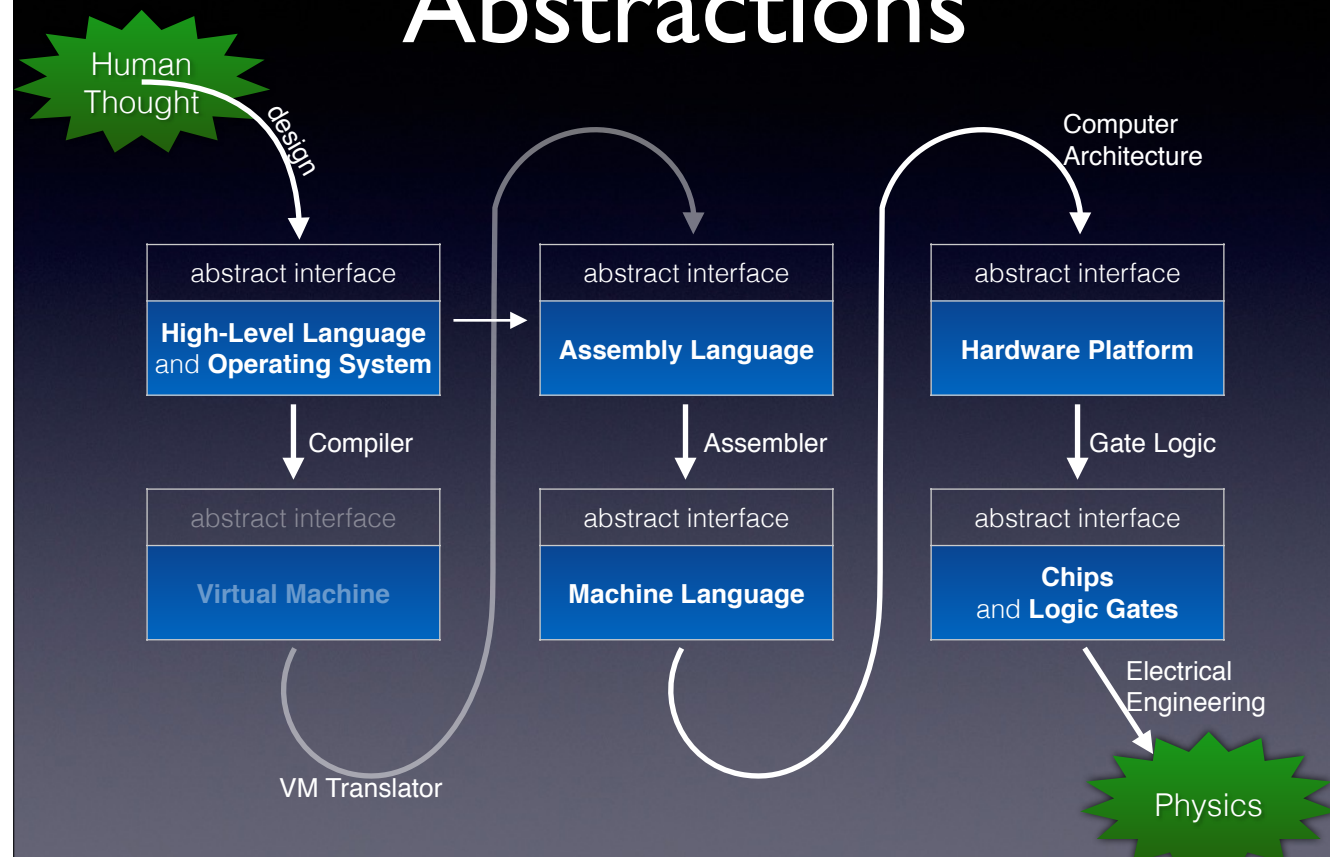
Abstractions



LD A, #65
RST 16
RET

D7
C9

Abstractions



LD A, #65
RST 16
RET

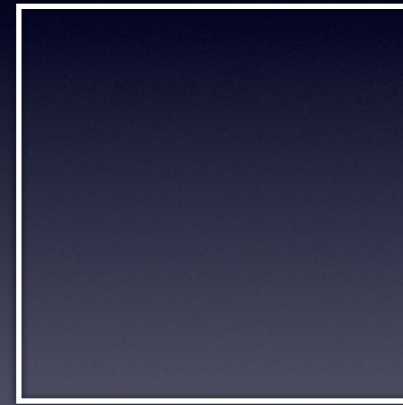
D7
C9

Introduction

- Programming is making the computer do what you want it to do
- Computer is very stupid...
- Describe what the computer needs to do
- Lets have a look at the options...

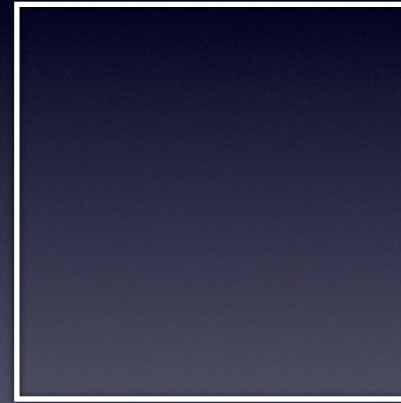
```
LD A, #65  
RST 16  
RET
```


Drawing a square



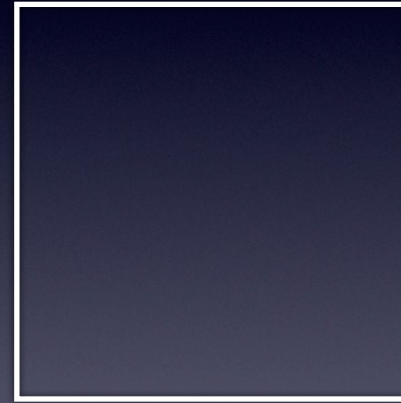
Drawing a square

- **Declarative**
Describe the properties



Drawing a square

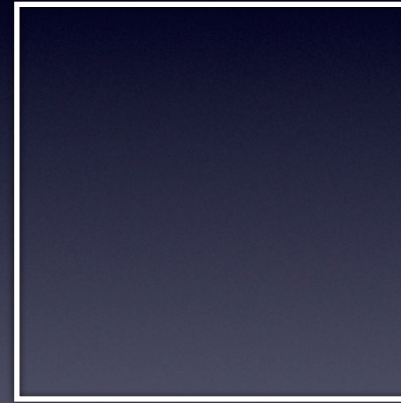
- **Declarative**
Describe the properties



```
box width 1i height 1i
```

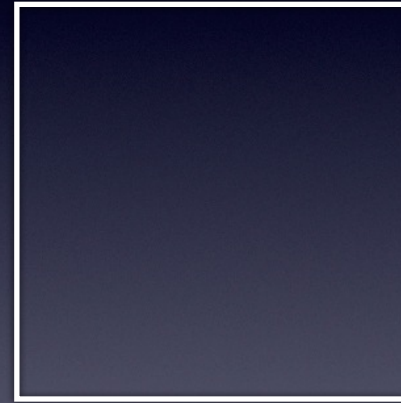
Drawing a square

- **Declarative**
Describe the properties
- **Imperative**
Describe how to build it



Drawing a square

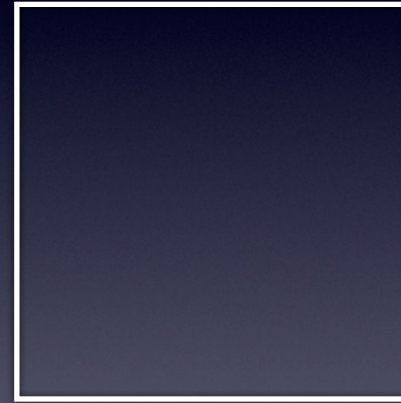
- **Declarative**
Describe the properties
- **Imperative**
Describe how to build it



```
0 0 moveto 100 0 lineto  
100 100 lineto ...
```


Drawing a square

- **Declarative**
Describe the properties
- **Imperative**
Describe how to build it
- Computers are naturally imperative
- But can use declarative descriptions



Breakdown

- Making the computer do what you want it to do
- Two parts
 - Breaking the problem down into small exact steps
 - Telling the computer how to perform those steps

High-Level Languages

- Computers only understand machine code
- Humans (generally) don't
- Although we will make you in CSA
- Write computer programs in a High-Level Language, such as C
- Get the computer to convert (compile) it to machine code

But what is a program?

- A program is a sequence of instructions
- Computer will execute them one by one
- Each instruction does one very simple thing
- Instructions act on some data
- Together they build up to do complex tasks

Program = Algorithms + Data Structures

But what is a program?

- A program is a sequence of instructions
- Computer will execute them one by one
- Each instruction does one very simple thing
- Instructions act on some data
- Together they build up to do complex tasks

Algorithm: how the instructions manipulate the data...

Types of Instructions

- Mathematical operations
- Variables — load and store data
- Conditionals — do this only if some condition is true
- Loops — repeat doing this
- Procedures — Define blocks of instructions for reuse

Counting Characters

Alice was beginning to get very tired of sitting by her sister on the bank, and of having nothing to do: once or twice she had peeped into the book her sister was reading, but it had no pictures or conversations in it, "and what is the use of a book," thought Alice "without pictures or conversation?"

Counting Characters

- Initialise a counter to zero
- Start at the first character
- Add one to counter
- Move to the next character
- Repeat until no more characters
- Counter contains the number of characters