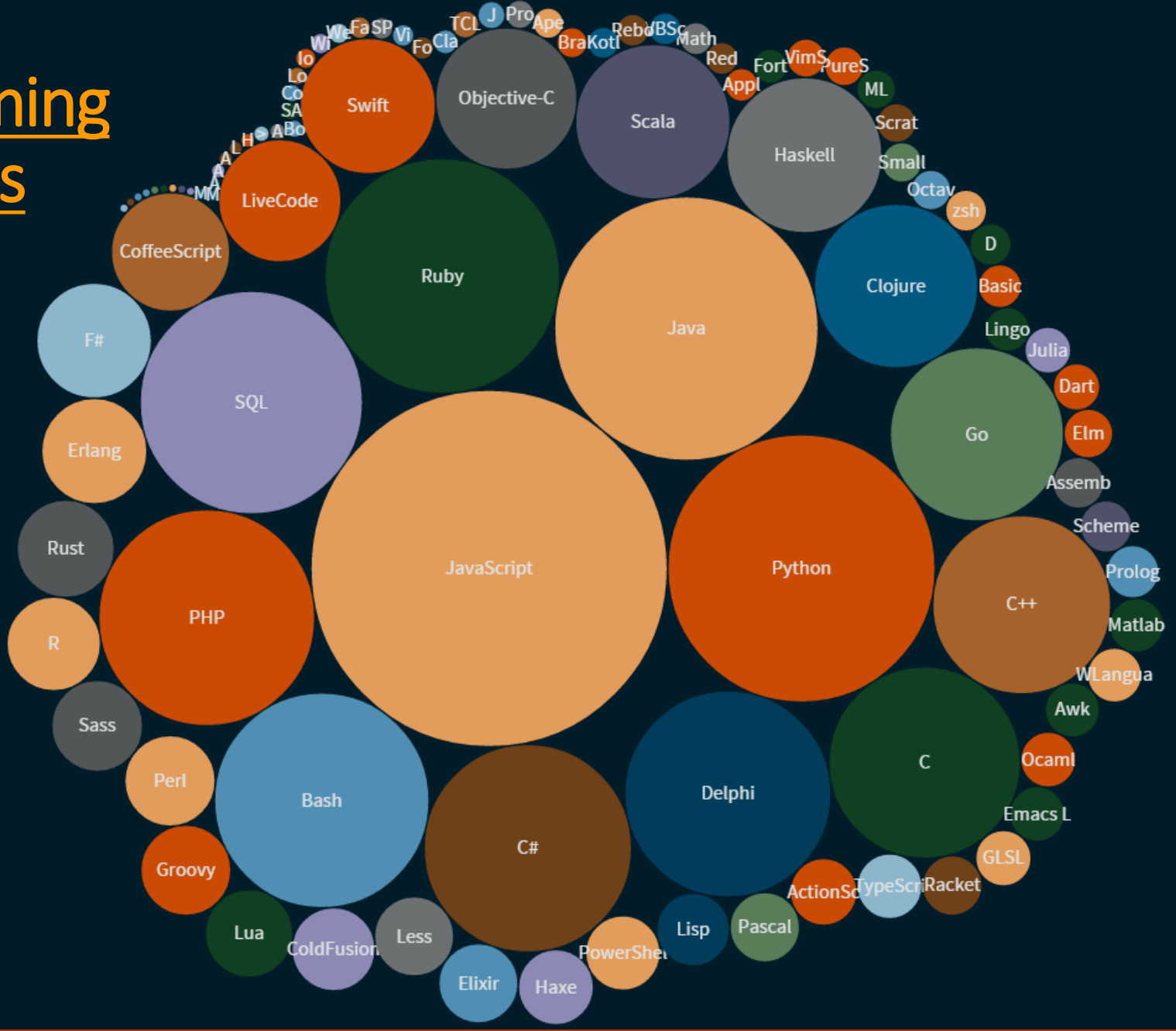
Programming Languages Pragmatics

INTRODUCTION

DR. ERIC CHOU

IEEE SENIOR MEMBER

Programming Languages

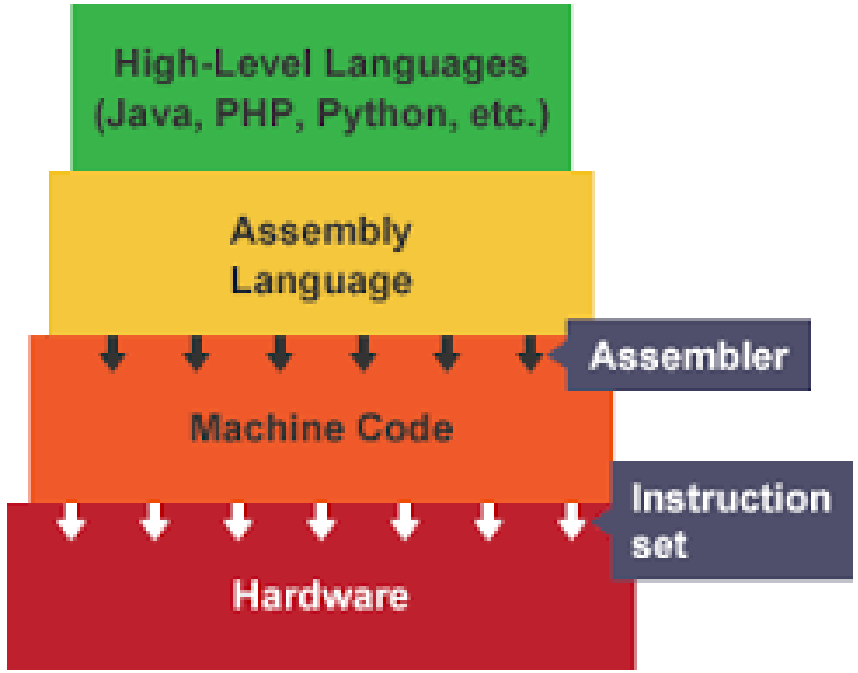


Machine Code

Unreadable

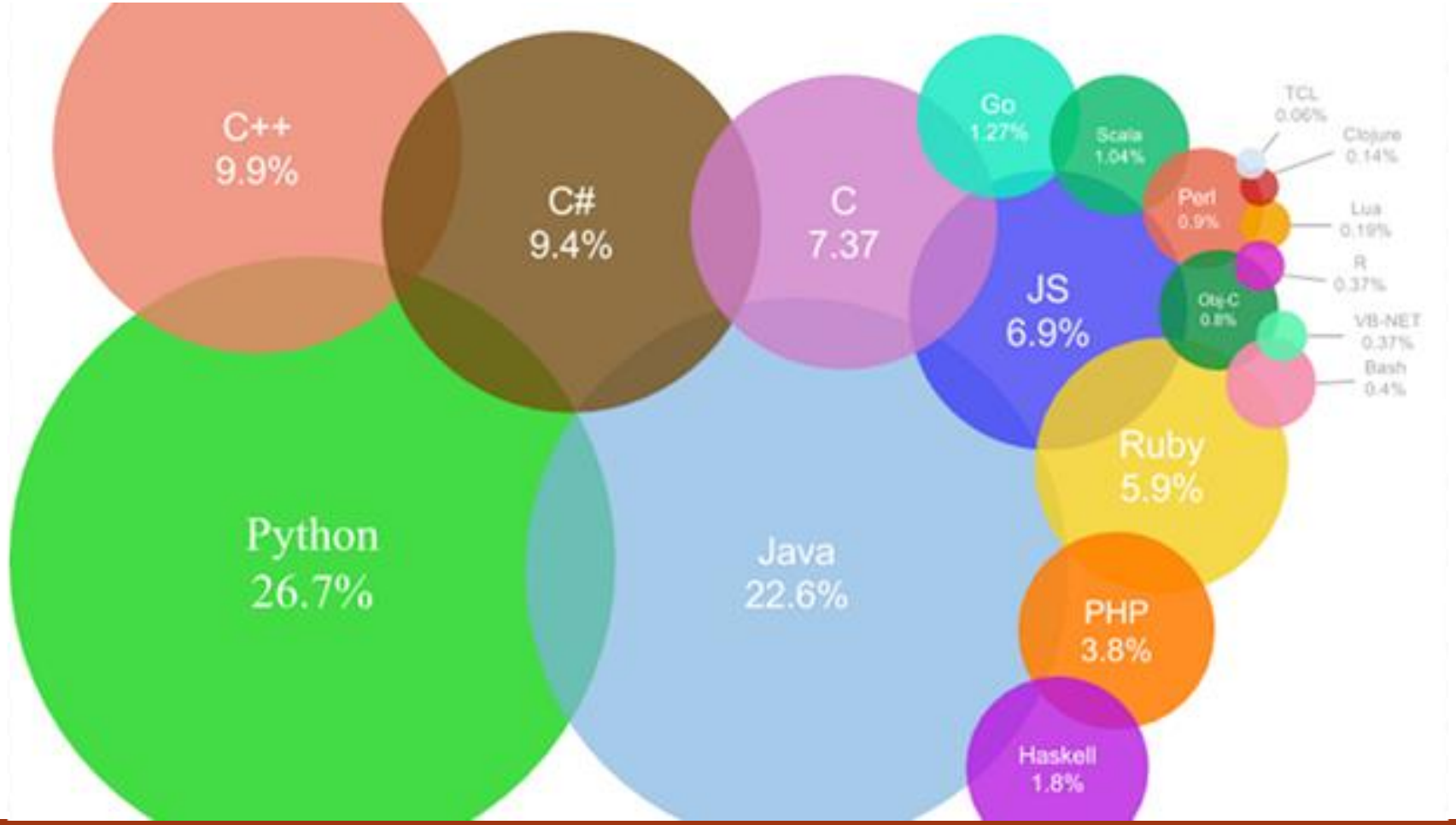
Assembly Language

GCD Code

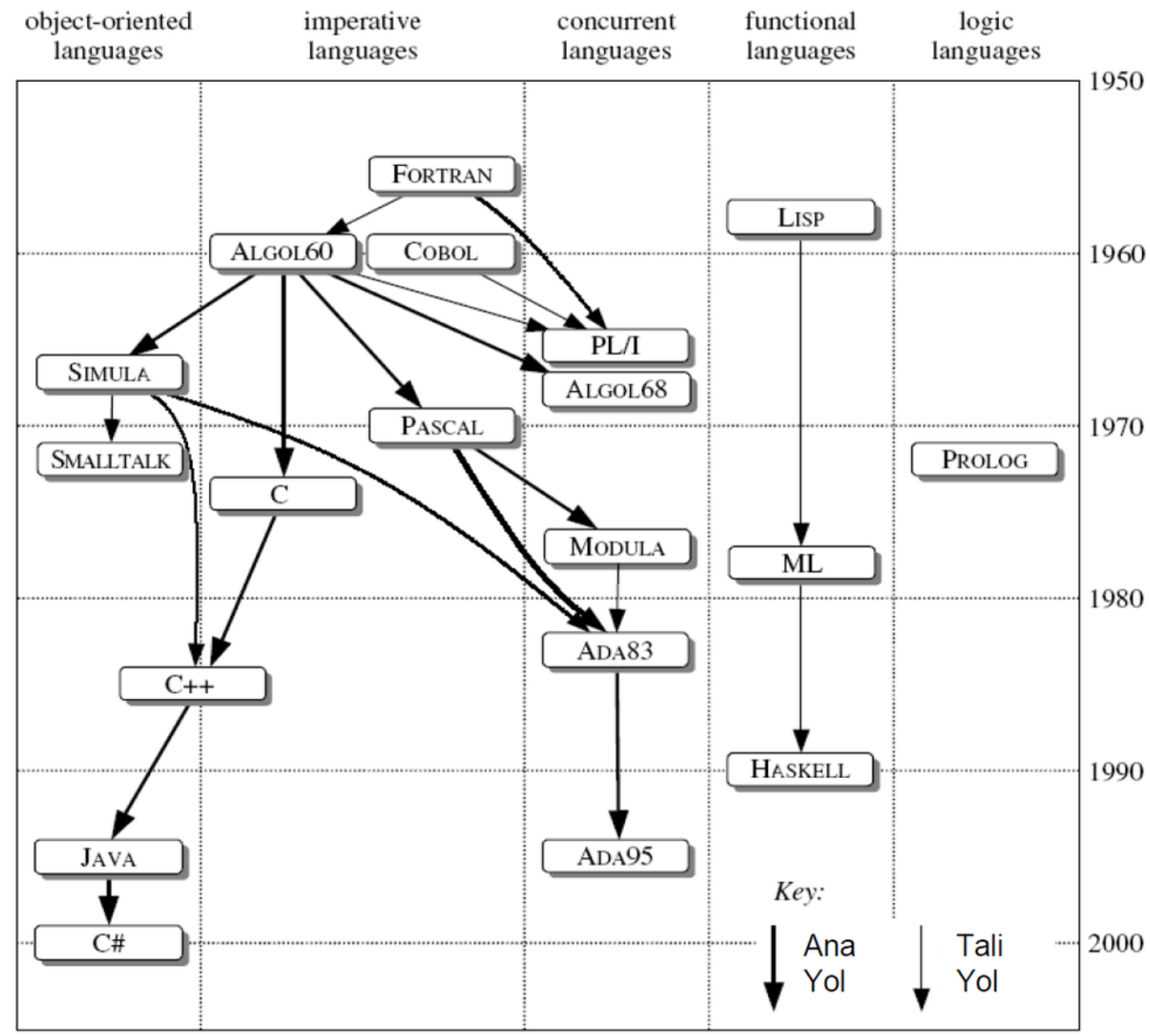



```
        pushl   %ebp                # \
        movl    %esp, %ebp          #  ) reserve space for local variables
        subl    $16, %esp           # /
        call    getint              # read
        movl    %eax, -8(%ebp)      # store i
        call    getint              # read
        movl    %eax, -12(%ebp)     # store j
A:      movl    -8(%ebp), %edi      # load i
        movl    -12(%ebp), %ebx     # load j
        cmpl    %ebx, %edi          # compare
        je      D                   # jump if i == j
        movl    -8(%ebp), %edi      # load i
        movl    -12(%ebp), %ebx     # load j
        cmpl    %ebx, %edi          # compare
        jle     B                   # jump if i < j
        movl    -8(%ebp), %edi      # load i
        movl    -12(%ebp), %ebx     # load j
        subl    %ebx, %edi          # i = i - j
        movl    %edi, -8(%ebp)      # store i
        jmp     C
B:      movl    -12(%ebp), %edi     # load j
        movl    -8(%ebp), %ebx      # load i
        subl    %ebx, %edi          # j = j - i
        movl    %edi, -12(%ebp)     # store j
C:      jmp     A
D:      movl    -8(%ebp), %ebx      # load i
        push    %ebx                # push i (pass to putint)
        call    putint              # write
        addl    $4, %esp            # pop i
        leave                       # deallocate space for local variables
        mov     $0, %eax            # exit status for program
        ret                         # return to operating system
```

Most Popular Programming Languages 2016



Timeline


object-oriented languages
imperative languages
concurrent languages
functional languages
logic languages
1950
1960
1970
1980
1990
2000
FORTRAN
ALGOL60
COBOL
LISP
PL/I
SIMULA
ALGOL68
PASCAL
SMALLTALK
PROLOG
C
MODULA
ML
ADA83
C++
HASKELL
JAVA
ADA95
C#
Key:
Ana Yol
Tali Yol

Why are there so many programming languages?

1. evolution -- we've learned better ways of doing things over time
2. socio-economic factors: proprietary interests, commercial advantage
3. orientation toward special purposes
4. orientation toward special hardware
5. http://cdn.oreillystatic.com/news/graphics/prog_lang_poster.pdf

What makes a language successful?

1. easy to learn (BASIC, Pascal, LOGO, Scheme)
2. easy to express things, easy use once fluent, "powerful" (C, Common Lisp, APL, Algol-68, Perl)
3. easy to implement (BASIC, Forth, Python)
4. possible to compile to very good (fast/small) code (Fortran, C)
5. backing of a powerful sponsor (COBOL, PL/1, Ada, Visual Basic, C#)
6. wide dissemination at minimal cost (Pascal, Turing, Java, Javascript)

Why do we have programming languages? What is a language for?

1. way of thinking -- way of expressing algorithms
2. languages from the user's point of view
3. abstraction of virtual machine -- way of specifying what you want
4. the hardware to do without getting down into the bits
5. languages from the implementor's point of view

The Art of Language Design

- Evolution
- Special Purposes
- Personal Preference
- Expressive Power
- Ease of Use for Novice
- Ease of Implementation
- Standardization
- Open Source
- Excellent Compiler
- Economics/Patronage/Inertia