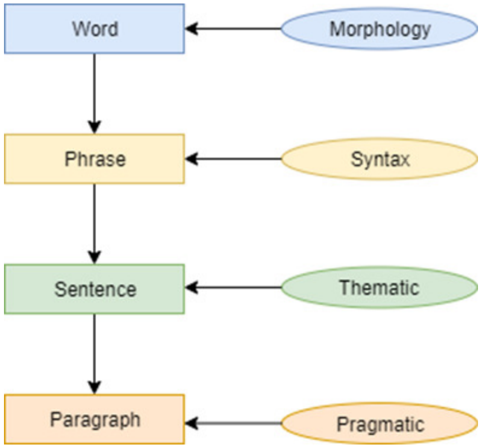
Theory of Programming Languages

Buddhitha Hettige
Department of Computer Engineering

1

Programming Languages

Formal Languages



Formal languages

- Attempts have been made by linguists in early 50's to define precisely
  - valid sentences
  - give structural descriptions of sentences
  - formal grammar
  - describe the rules of grammar in rigorous mathematical way

to describe English

Formal languages ...

- It was believed that, such description of natural languages would make language translation using computers easily.
- Noam Chomsky gave a mathematical model of a grammar in 1956.
- It turned out to be useful for computer languages but not for natural languages.
- Definition of **context-free grammar** by Chomsky was used to describe **Algorithmic Languages**

Programming language?

- A programming language is a set of rules that provides a way of telling a computer what operations to perform.
- A programming language is a set of rules for communicating an algorithm
- It provides a linguistic framework for describing computations

Programming language?

- A programming language is a notational system for describing computation in a machine-readable and human-readable form.
- A programming language is a tool for developing executable models for a class of problem domains.

Programming language?

- English is a natural language. It has words, symbols and grammatical rules.
- A programming language also has words, symbols and rules of grammar.
- The grammatical rules are called syntax.
- Each programming language has a different set of syntax rules

Programming language?

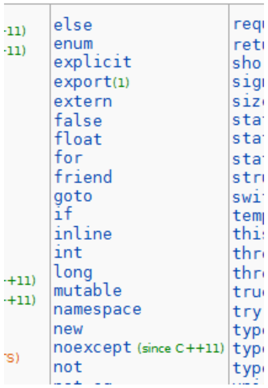
- Programming languages have evolved over time as better ways have been developed to design them.
- First programming languages were developed in the 1950s
- Since then thousands of languages have been developed
- Different programming languages are designed for different types of programs.

C++ Vs Natural Languages

- C++
  - Artificial Language
  - Consist of
    - Keywords
    - Syntax
    - Semantics
  - Translate through the Compilers
- Natural Language
  - Natural
  - Consist of
    - Words
    - Syntax
    - Semantics
  - Translate trough the Machine Translation systems or Human

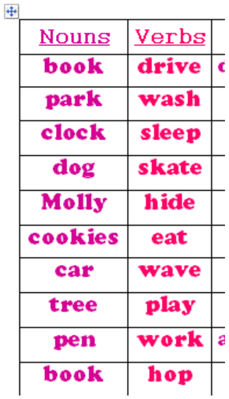
Key words/ words

- C++



-11)	else	req
-11)	enum	ret
	explicit	sho
	export(1)	sig
	extern	siz
	false	sta
	float	sta
	for	sta
	friend	str
	goto	swi
	if	tem
	inline	thi
	int	thr
-+11)	long	thr
-+11)	mutable	tru
	namespace	try
	new	typ
'S)	noexcept (since C++11)	typ
	not	typ

- Natural Language

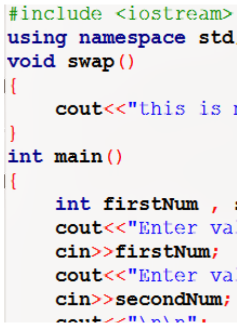


Nouns	Verbs	
book	drive	c
park	wash	
clock	sleep	
dog	skate	
Molly	hide	
cookies	eat	
car	wave	
tree	play	
pen	work	a
book	hop	

Syntax

- C++

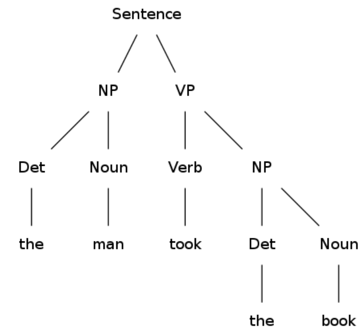
Rules for construction of valid statements, including, Order of words, Punctuation



```
#include <iostream>
using namespace std
void swap()
{
    cout<<"this is 
}
int main()
{
    int firstNum , 
    cout<<"Enter va
    cin>>firstNum;
    cout<<"Enter va
    cin>>secondNum;
```

- Natural Language

Grammar rules, subject, object, verbs etc.



Semantics

- C++

The set of rules that determines the meaning of instructions (what the computer will do) written in a programming language.

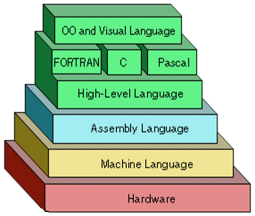
- Natural Language

Is the study of meaning

Programming language generations

This classification is used to indicate increasing power of programming styles

1. *First-generation programming languages*
2. Second-generation programming languages
3. Third-generation programming languages
4. Fourth-generation programming languages
5. Fifth-generation programming languages



OO and Visual Language

FORTRAN

C

Pascal

High-Level Language

Assembly Language

Machine Language

Hardware

First-generation programming language (1GL)

- Is a machine-level programming language
- Translator isn’t used to compile
- The instructions in 1GL are made of binary numbers, represented by 1s and 0s
- Advantage
  - The code can run very fast and very efficiently because the instructions are executed directly by the CPU
- Disadvantage
  - When an error occurs, the code is not as easy to fix

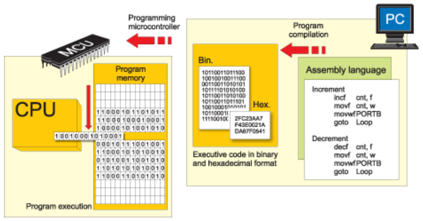
Human to Machine

Executable Machine code

- 0001001001000101 0001001001000101 0010010011101100 0010010011101100 10101101001... 10101101001...

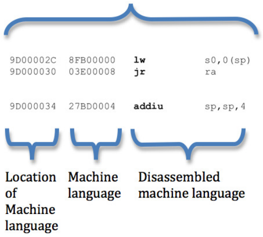
Second-generation programming language(2GL)

- Assembly language.
- Properties
  - The code can be read and written by a programmer
  - The language is specific to a particular processor family and environment
- Used in kernels and device drivers



```
Increment
        incf    cnt, f
        movf    cnt, w
        movwfPORTB
        goto    Loop

Decrement
        decf    cnt, f
        movf    cnt, w
        movwfPORTB
        goto    Loop
```

Code generated by assembler



```
9D00002C  8FB00000  lw      s0,0(sp)
9D000030  03E00008  jr      ra

9D000034  27BD0004  addiu   sp,sp,4
```

Location of Machine language

Machine language

Disassembled machine language

Human to Machine

High-level program

x = b*h/2; return x

Low-level program

```
LOAD r1,b        LOAD r1,b
LOAD r2,h        LOAD r2,h
MUL r1,r2        MUL r1,r2
DIV r1,#2        DIV r1,#2
```

Advantages

- It requires less memory and execution time;
- It allows hardware-specific complex jobs in an easier way;
- It is suitable for time-critical jobs;
- It is most suitable for writing interrupt service routines and other memory resident programs.

Syntax

- One statement per line
- Format
  - [label] mnemonic [operands] [;comment]

```
INC COUNT        ; Increment the memory variable COUNT

MOV TOTAL, 48    ; Transfer the value 48 in the
                 ; memory variable TOTAL

ADD AH, BH       ; Add the content of the
                 ; BH register into the AH register

AND MASK1, 128   ; Perform AND operation on the
                 ; variable MASK1 and 128

ADD MARKS, 10    ; Add 10 to the variable MARKS
MOV AL, 10       ; Transfer the value 10 to the AL register
```

Hello world program

```
section .text
   global _start     ;must be declared for linker (ld)

_start:             ;tells linker entry point
   mov  edx,len     ;message length
   mov  ecx,msg     ;message to write
   mov  ebx,1       ;file descriptor (stdout)
   mov  eax,4       ;system call number (sys_write)
   int  0x80        ;call kernel

   mov  eax,1       ;system call number (sys_exit)
   int  0x80        ;call kernel

section .data
msg db 'Hello, world!', 0xa  ;string to be printed
len equ $ - msg     ;length of the string
```

Instruction Set

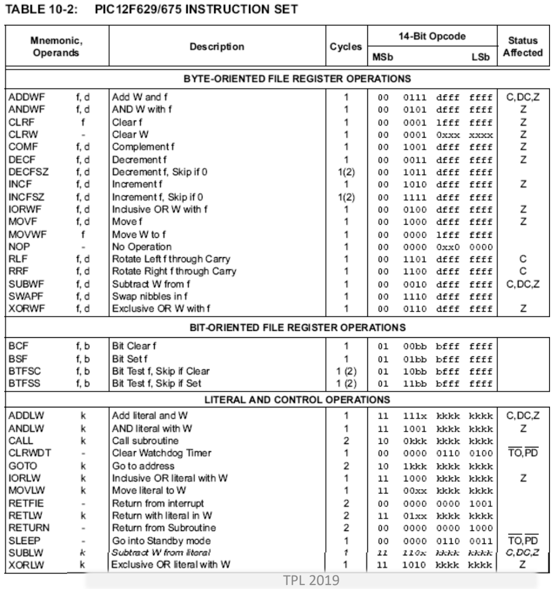
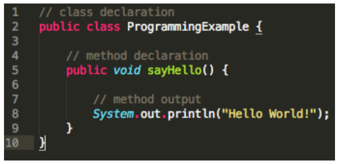


TABLE 10-2: PIC12F629/675 INSTRUCTION SET

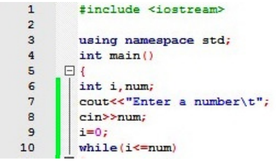
Mnemonic, Operands		Description	Cycles	14-Bit Opcode				Status Affected
				MSb			LSb	
BYTE-ORIENTED FILE REGISTER OPERATIONS								
ADDWF	f, d	Add W and f	1	00	0111	dfff	ffff	C,DC,Z
ANDWF	f, d	AND W with f	1	00	0101	dfff	ffff	Z
CLRF	f	Clear f	1	00	0001	lfff	ffff	Z
CLRW	-	Clear W	1	00	0001	0xxx	xxxx	Z
COMF	f, d	Complement f	1	00	1001	dfff	ffff	Z
DECF	f, d	Decrement f	1	00	0011	dfff	ffff	Z
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00	1011	dfff	ffff	
INCF	f, d	Increment f	1	00	1010	dfff	ffff	Z
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff	
IORWF	f, d	Inclusive OR W with f	1	00	0100	dfff	ffff	Z
MOVF	f, d	Move f	1	00	1000	dfff	ffff	Z
MOVWF	f	Move W to f	1	00	0000	lfff	ffff	
NOP	-	No Operation	1	00	0000	0xx0	0000	
RLF	f, d	Rotate Left f through Carry	1	00	1101	dfff	ffff	C
RRF	f, d	Rotate Right f through Carry	1	00	1100	dfff	ffff	C
SUBWF	f, d	Subtract W from f	1	00	0010	dfff	ffff	C,DC,Z
SWAPF	f, d	Swap nibbles in f	1	00	1110	dfff	ffff	
XORWF	f, d	Exclusive OR W with f	1	00	0110	dfff	ffff	Z
BIT-ORIENTED FILE REGISTER OPERATIONS								
BCF	f, b	Bit Clear f	1	01	00bb	bfff	ffff	
BSF	f, b	Bit Set f	1	01	01bb	bfff	ffff	
BTFSC	f, b	Bit Test f, Skip if Clear	1 (2)	01	10bb	bfff	ffff	
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)	01	11bb	bfff	ffff	
LITERAL AND CONTROL OPERATIONS								
ADDLW	k	Add literal and W	1	11	111x	kkkk	kkkk	C,DC,Z
ANDLW	k	AND literal with W	1	11	1001	kkkk	kkkk	Z
CALL	k	Call subroutine	2	10	0kkk	kkkk	kkkk	
CLRWDT	-	Clear Watchdog Timer	1	00	0000	0110	0100	$\overline{TO}$,$\overline{PD}$
GOTO	k	Go to address	2	10	1kkk	kkkk	kkkk	
IORLW	k	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z
MOVLW	k	Move literal to W	1	11	00xx	kkkk	kkkk	
RETFIE	-	Return from interrupt	2	00	0000	0000	1001	
RETLW	k	Return with literal in W	2	11	01xx	kkkk	kkkk	
RETURN	-	Return from Subroutine	2	00	0000	0000	1000	
SLEEP	-	Go into Standby mode	1	00	0000	0110	0011	$\overline{TO}$,$\overline{PD}$
SUBLW	k	*Subtract W from literal*	1	11	110x	kkkk	kkkk	C,DC,Z
XORLW	k	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z

Third-generation programming languages (3GL)

- Languages are more programmer-friendly
- Example
  - C, C++, C#, Java, BASIC and Pascal
- Support structured programming.
- Must be translated into machine language by a compiler or interpreter
- Advantages
  - Easier to read, write, and maintain

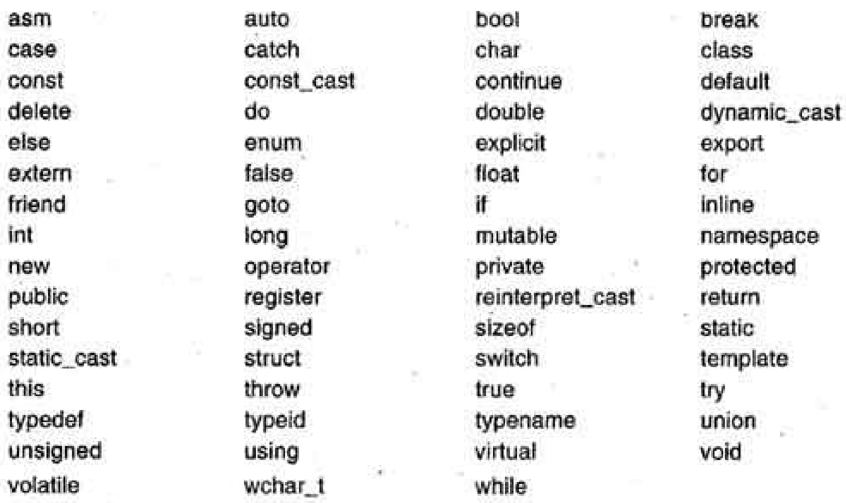


```
// class declaration
public class ProgrammingExample {

    // method declaration
    public void sayHello() {

        // method output
        System.out.println("Hello World!");
    }
}
```



```
#include <iostream>

using namespace std;
int main()
{
int i,num;
cout<<"Enter a number\t";
cin>>num;
i=0;
while(i<=num)
```

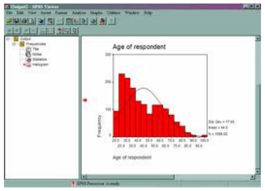
C++ Keywords

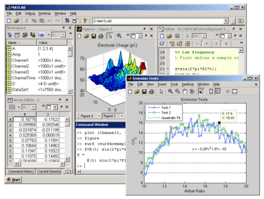


asm	auto	bool	break
case	catch	char	class
const	const_cast	continue	default
delete	do	double	dynamic_cast
else	enum	explicit	export
extern	false	float	for
friend	goto	if	inline
int	long	mutable	namespace
new	operator	private	protected
public	register	reinterpret_cast	return
short	signed	sizeof	static
static_cast	struct	switch	template
this	throw	true	try
typedef	typeid	typename	union
unsigned	using	virtual	void
volatile	wchar_t	while	

Fourth-generation programming languages(4GL)

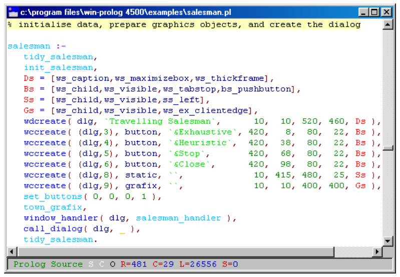
- Designed to reduce programming effort
- Consist of
  - Set of libraries
  - CRUD generators
  - Report generators
  - DBMS
  - Visual design tool and integration API
- Different types of 4GLs
  - Table-driven (codeless) programming
    - PowerBuilder
  - Data management
    - SAS, SPSS
  - Report-generator programming languages
    - Oracle Developer Suite





Fifth-generation programming language(5GL)

- Based on **solving problems** using constraints given to the program, rather than using an algorithm written by a programmer
- Use mainly in Artificial Intelligence research
- Example
  - Prolog, OPS5, and Mercury



```
% initialise data, prepare graphics objects, and create the dialog

salesman :-
   tidy_salesman,
   init_salesman,
   Ds = [ws_caption,ws_maximizebox,ws_thickframe],
   Bs = [ws_child,ws_visible,ws_tabstop,bs_pushbutton],
   Ss = [ws_child,ws_visible,ss_left],
   Gs = [ws_child,ws_visible,ws_ex_clientedge],
   wdcreate( dlg, `Travelling Salesman`,     10,  10, 520, 460, Ds ),
   wccreate( (dlg,3), button, `&Exhaustive`, 420,   8,  80,  22, Bs ),
   wccreate( (dlg,4), button, `&Heuristic`,  420,  38,  80,  22, Bs ),
   wccreate( (dlg,5), button, `&Stop`,       420,  68,  80,  22, Bs ),
   wccreate( (dlg,6), button, `&Close`,      420,  98,  80,  22, Bs ),
   wccreate( (dlg,8), static, ``,             10, 415, 480,  25, Ss ),
   wccreate( (dlg,9), grafix, ``,             10,  10, 400, 400, Gs ),
   set_buttons( 0, 0, 0, 1 ),
   town_grafix,
   window_handler( dlg, salesman_handler ),
   call_dialog( dlg, _ ),
   tidy_salesman.
```

```
Prolog Source S C O R=481 C=29 L=26556 S=0
```

Example

```
course(csu2280, as101, 76).
course(csu2280, as102, 56).
course(csu2280, as103, 45).
course(csu2279, as101, 78).
course(csu2279, as102, 29).

printList([]).
printList([H|T]) :- write(H),nl,printList(T).

stuList(Cou) :-     write('stu listRule Starting...'),nl,
                          setof(ID, Mark^ course(Cou,ID,Mark), List),
                          printList(List).
```

High Level Languages

- A high level language (4GL) that requires fewer instructions to accomplish a task than a third generation language.
  - Used with databases
  - Query languages
  - Report generators
  - Forms designers
  - Application generators

High Level Languages

- Declarative languages
- Functional(?): Lisp, Scheme, SML
- Also called applicative
- Everything is a function
- Logic: Prolog ◦ Based on mathematical logic Rule- or Constraint-based

Natural Language programs

- Though no clear definition at present, natural language programs generally can be interpreted and executed by the computer with no other action by the user than stating their question directly.
- However, at present capabilities of natural language programs are limited .

Programming paradigms

- Imperative Programming (procedural programming ?) (C)
- Object-Oriented Programming (C++)
- Logic/Declarative Programming (Prolog)
- Functional/Applicative Programming (Lisp)

Two broad groups of programming languages

Traditional programming languages

Sequences of instructions first, second and some third generation languages

Object-oriented languages

Objects are created rather than sequences of instructions

Some third generation, and fourth and fifth generation languages are examples for OOLs

Features

FORTRAN - FORmula TRANslation.

- Developed at IBM in the mid-1950s.
- Designed for scientific and mathematical applications by scientists and engineers.

COBOL - COmmon Business Oriented Language.

- Developed in 1959.
- Designed to be common to many different computers.
- Typically used for business applications.

BASIC - Beginner's All-purpose Symbolic Instruction Code.

- Developed at Dartmouth College in mid 1960s.
- Developed as a simple language for students to write programs with which they could interact through terminals.

C

- Developed by Bell Laboratories in the early 1970s.
- Provides control and efficiency of assembly language while having third generation language features.
- Often used for system programs.
- UNIX is written in C.

Simula

- First object-oriented language ◦
- Developed by Ole Johan Dahl in the 1960s.

Smalltalk

- First purely object-oriented language.
- Developed by Xerox in mid-1970s. ◦
- Still in use on some computers.

C++

- It is C language with additional features.
- Widely used for developing system and application software.
- Graphical user interfaces can be developed easily with visual programming tools.

JAVA

- An object-oriented language similar to C++ that eliminates lots of C++'s problematic features
- Allows a web page developer to create programs for applications, called applets that can be used through a browser.
- Objective of JAVA developers is that it be machine, platform and operating system independent.

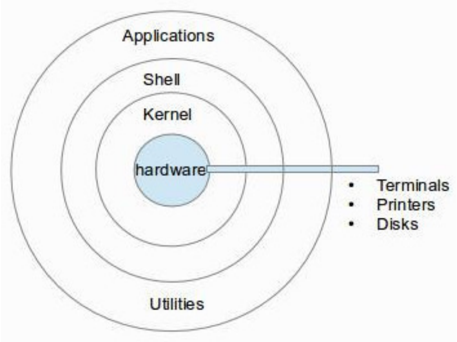
Scripting Languages

Scripting Languages

- JavaScript and VBScript
- Php and ASP
- Perl and Python

Command Languages

- sh, csh, bash (shell programming – hardware-kernal – shell – user)



Text processing Languages

- LaTex, PostScript
- HTML ◦ Hyper Text Markup Language.
- Used on the Internet and the World Wide Web (WWW).
- Web page developer puts brief codes called tags in the page to indicate how the page should be formatted.

About Programming languages

- When it comes to mechanics of the task i.e. The activity of programming, learning use a programming language is in many ways like learning to speak a human language
- In both kind of languages one has to learn new vocabulary, syntax and semantics (new words, sentence structure and meaning)}
- Both kind of language require considerable practice to gain proficiency.

About Programming languages

- Computer languages lack ambiguity and vagueness (uncertainty/indefiniteness)
- In English sentences such as "*Take a pinch of salt*" (How much is a pinch?) or "*Republicans grill IRS Chief over lost emails*" or "*look at the dog with one eye*" or "*I saw a man with a binoculars*"
- In a programming language a sentence either means one thing or it means nothing

About Programming languages

- **Formerly:** Run-time performance ◦ (Computers were more expensive than programmers)
- **Now:** Life cycle (human) cost is more important - Ease of designing, coding
  - – Debugging
  - – Maintenance
  - – Reusability

Characteristics (attributes) of programming languages

- **Writability:** The quality of a language that enables a programmer to use it to express a computation clearly, correctly, concisely, and quickly.
- **Readability:** The quality of a language that enables a programmer to understand and comprehend the nature of a computation easily and accurately.

Characteristics (attributes) of programming languages ...

- **Orthogonality:** The quality of a language that features provided have as few restrictions as possible and be combinable in any meaningful way.
- **Reliability:** The quality of a language that assures a program will not behave in unexpected or disastrous ways during execution.
- **Maintainability:** The quality of a language that eases errors can be found and corrected and new features added.

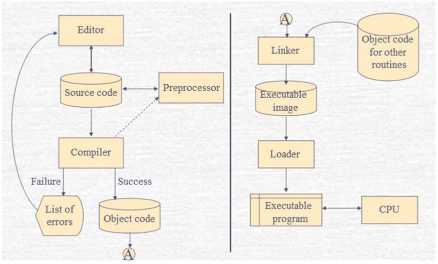
Characteristics (attributes) of programming languages ...

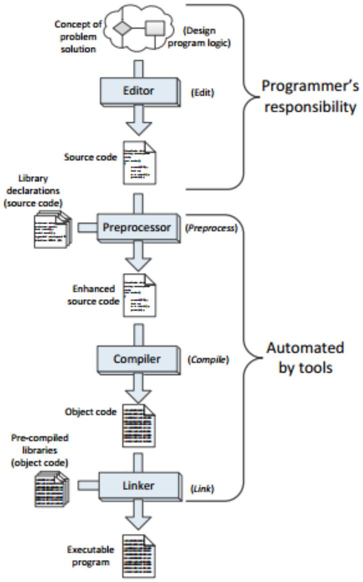
- **Generality:** The quality of a language that avoids special cases in the availability or use of constructs and by combining closely related constructs into a single more general one.
- **Uniformity:** The quality of a language that similar features should look similar and behave similar.
- **Extensibility:** The quality of a language that provides some general mechanism for the user to add new constructs to a language.

Characteristics (attributes) of programming languages ...

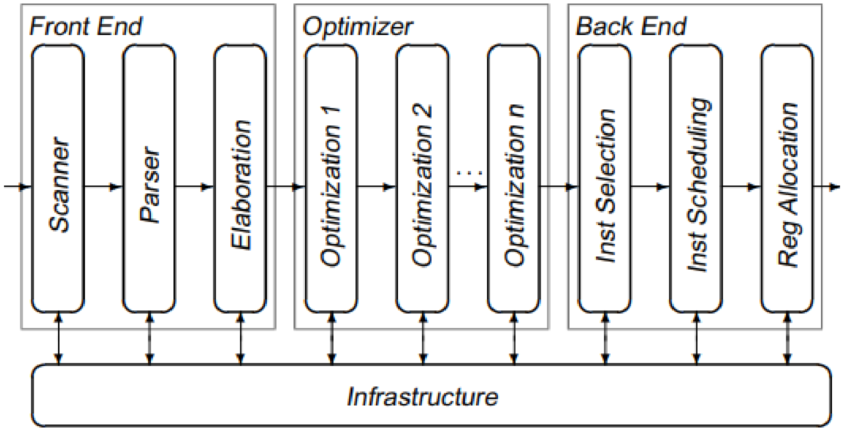
- **Standardability:** The quality of a language that allows programs written to be transported from one computer to another without significant change in language structure.
- **Implementability:** The quality of a language that provides a translator or interpreter can be written. This can address to complexity of the language definition.

Programming





Compiler



Activity/Assignment

Write 5 computer programs in five different programming languages to find the roots of the quadric equation $ax^2+bx+c=0$ where x represents a variable or an unknown, and a, b, and c are constants. Your program should run until user ask to exit.

Hint: The following equation shows the solutions of the quadratic equation

$$x = \frac{-b+\sqrt{b^2-4ac}}{2a} \quad \text{and} \quad x = \frac{-b-\sqrt{b^2-4ac}}{2a}$$