

# CSE240 – Introduction to Programming Languages (online)

Lecture 02:  
History of Programming Languages, Paradigms, and Features

**Javier Gonzalez-Sanchez**

javiergs@asu.edu  
javiergs.engineering.asu.edu  
Office Hours: By appointment

1940

## Machine language

A program is a sequence of bytes (binary numbers) stored in memory. The CPU interprets the program in the specified order.

```
00000001 00000011  
00000001 00000101  
00000010 00000010  
00000011 00000111
```

1947

## Assembly Language

**Use of letters and numbers** (hexadecimal or decimal)

led to programming becoming more sophisticated  
(e.g., Intel assembly, MIPS, and similar to a bytecode  
for Java).

Requires a program (**assembler**) to translate assembly  
language to machine code

LOD 3

LOD 5

OPR 2

STO 7

00000001 00000011

00000001 00000101

00000010 00000010

00000011 00000111

1952

## High-level languages

Higher level programming languages, such as Autocode, FORTRAN, COBOL, LISP, Scheme, and C.

- **Variables** of different types (real, integer, array)
- **Procedures or functions**
- **Control structures**

Requires a program (**compiler**) to translate to assembly language and a program (assembler) to translate it to machine code

x = 3 + 5;

LOD 3

LOD 5

OPR 2

STO 7

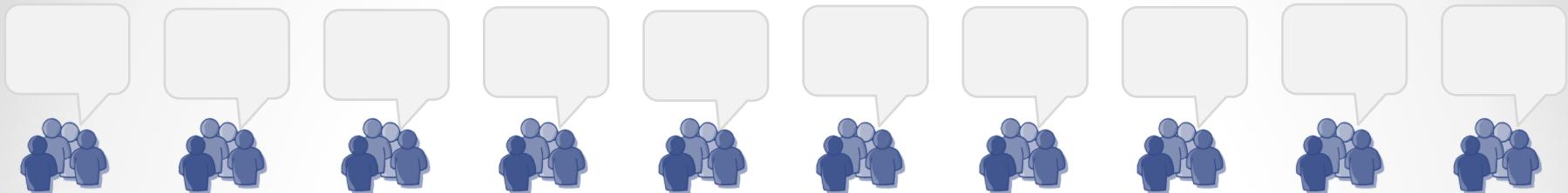
00000001 00000011

00000001 00000101

00000010 00000010

00000011 00000111

# Paradigms



```
x = 2 + 3;  
  
do {}  
while(true);  
  
int x = 5;
```

```
(= X (SUM 2 3))  
  
( repeat 0 10)  
  
(5)
```

```
x ← 2 + 3.  
  
for 1 to 10  
  
x := (number) 5.
```

What is  
the result of  
two plus three?

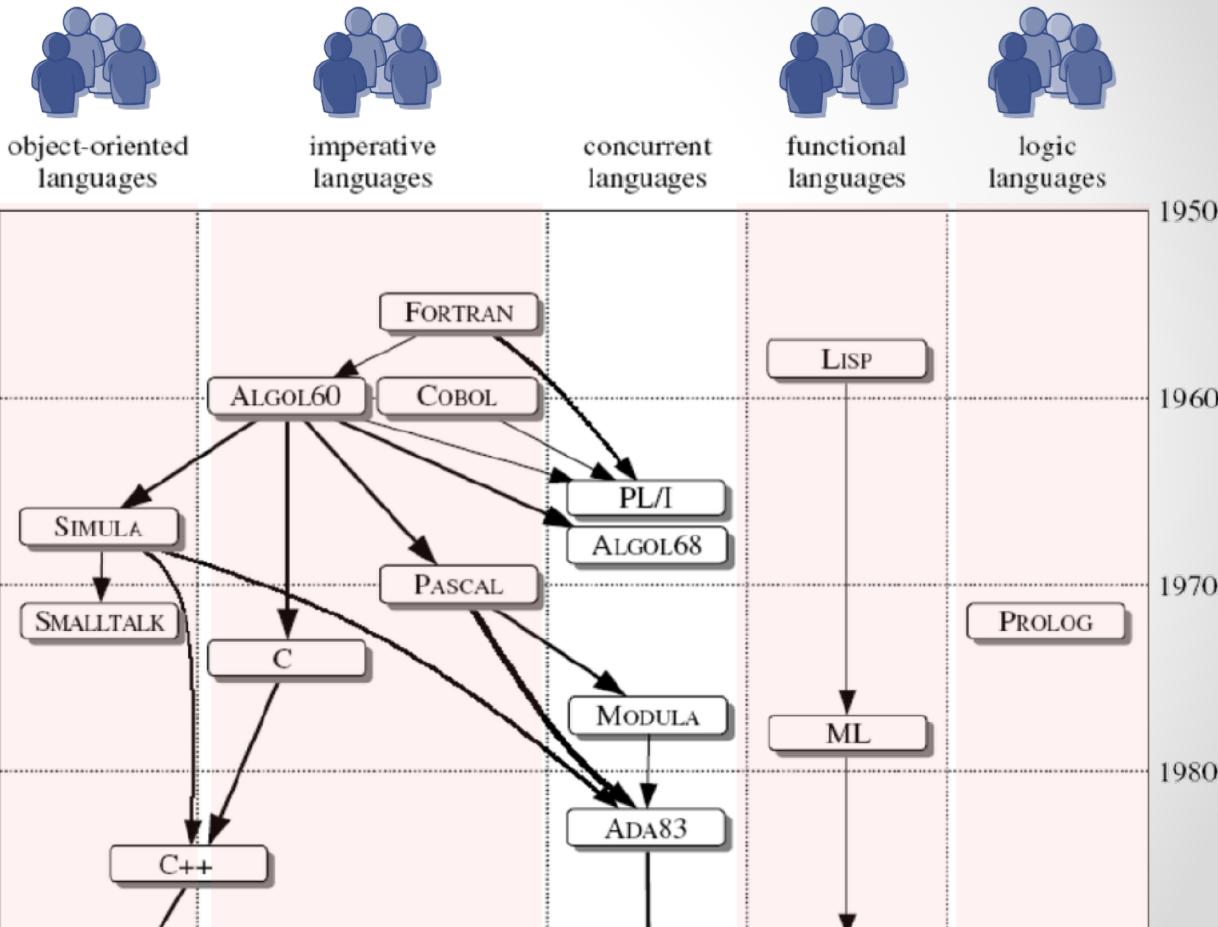


## Paradigm:

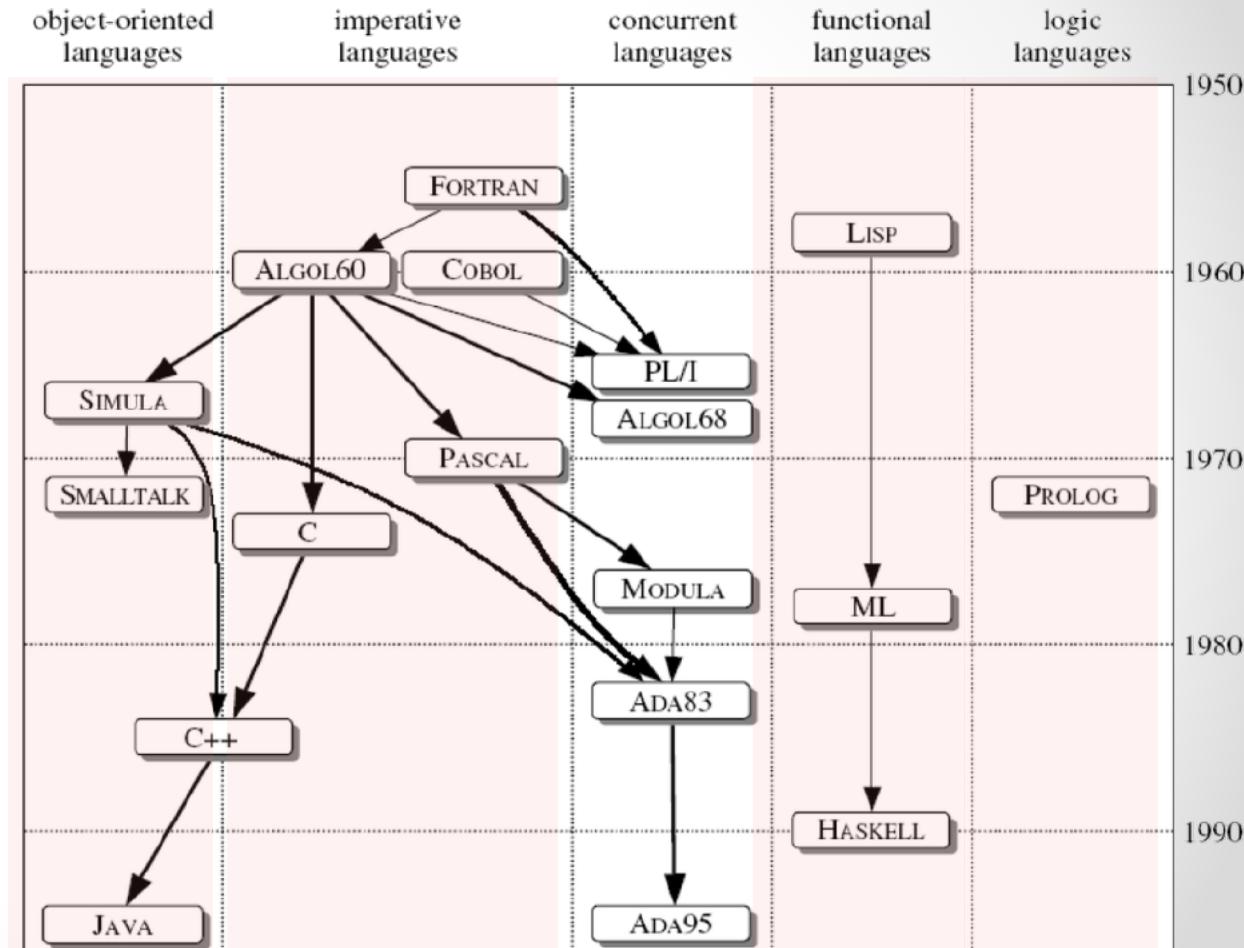
Basic assumptions, ways of thinking, and methodology  
that are commonly accepted  
by members of a group or community.



# Paradigms



# Paradigms



# The Four Main Paradigms and their Features

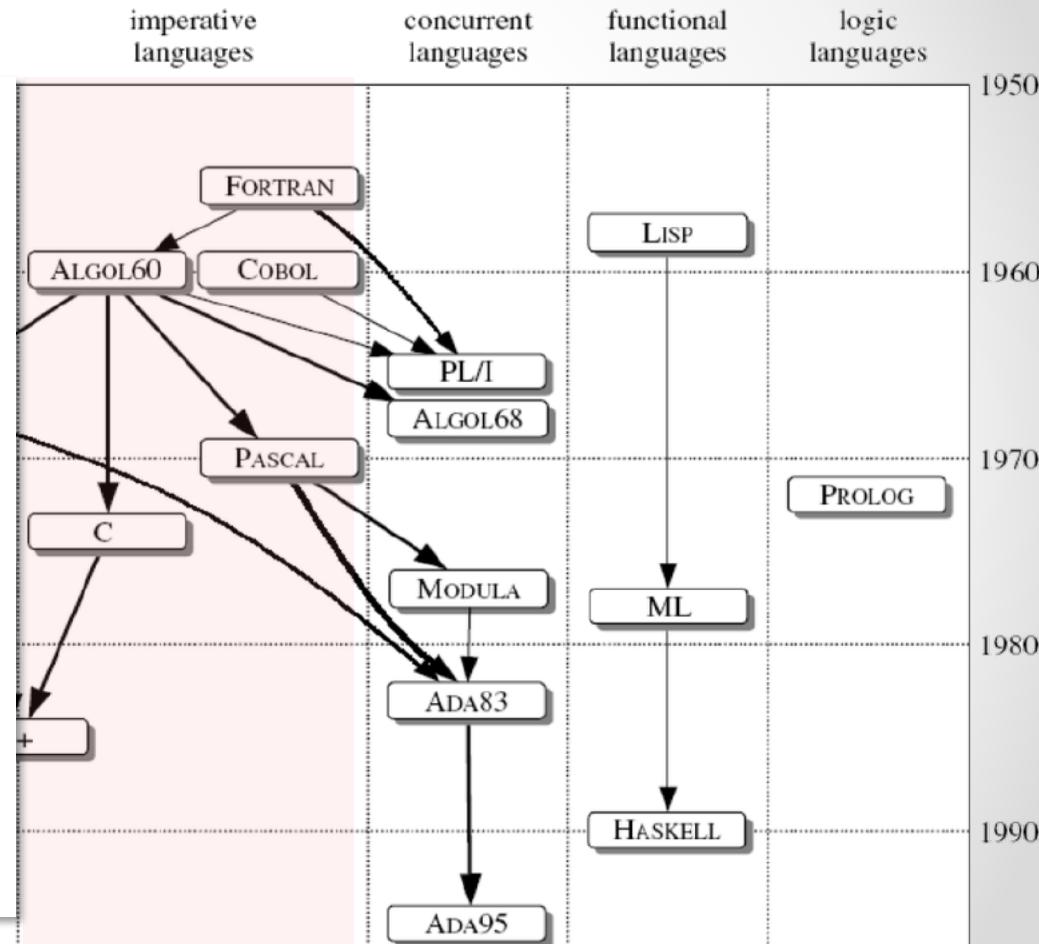
## Imperative or Procedural Paradigm

### Key idea:

Fully specified and fully controlled manipulation of named data in a step-wise fashion.

### Features:

- Divide the program into reasonable sized pieces named functions or procedures or modules or subroutines...
- Local and global variables
- Data structures
- Conditional and loop statements



# The Four Main Paradigms and their Features

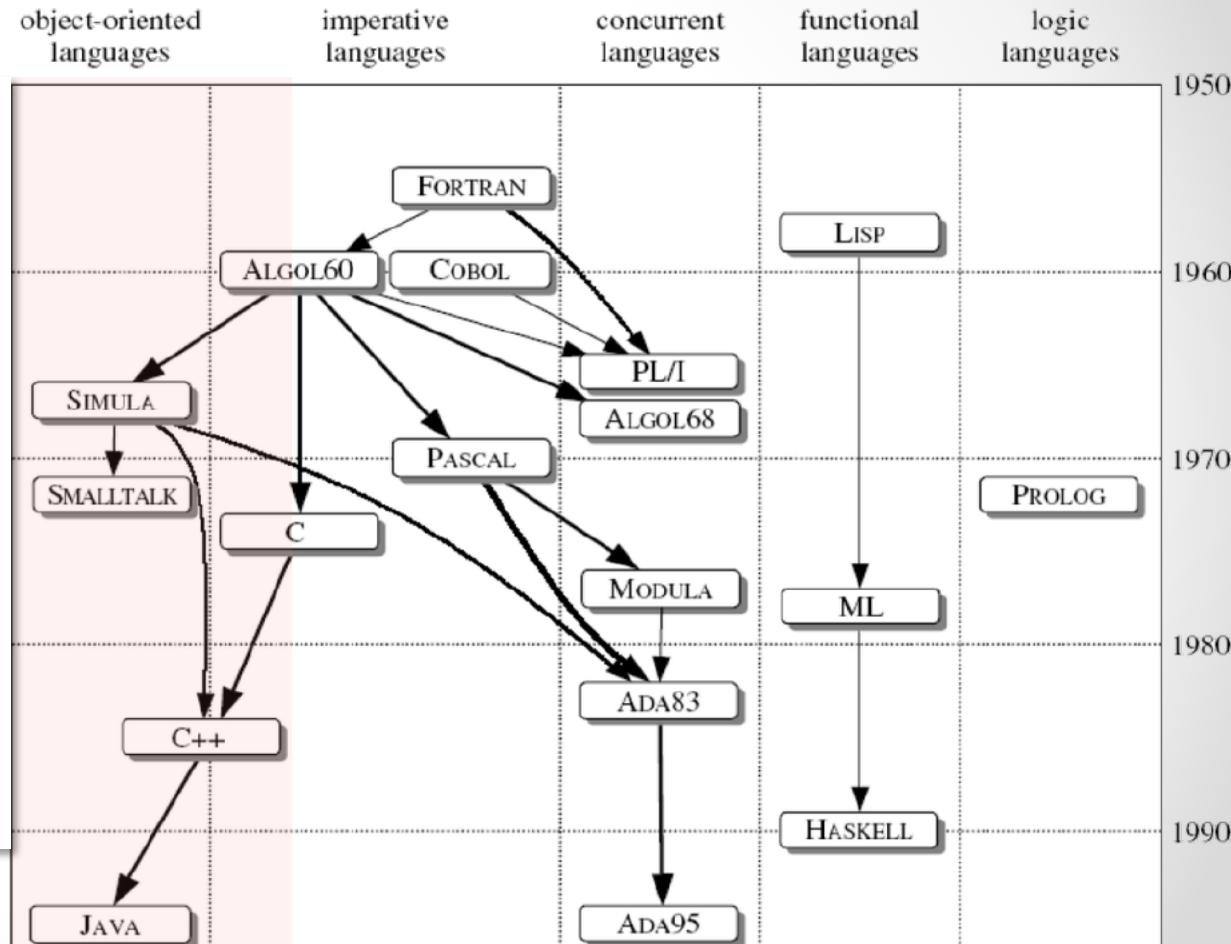
## Object-oriented Paradigm

**Key idea:**

encapsulation of data and functionality of the program in objects.

**Features:**

- Inheritance and polymorphism
- Class vs Instance variables (static)
- Public, private, protected



## Functional Paradigm

### Key idea:

Focus on higher level of **abstraction**  
(free from programming details)

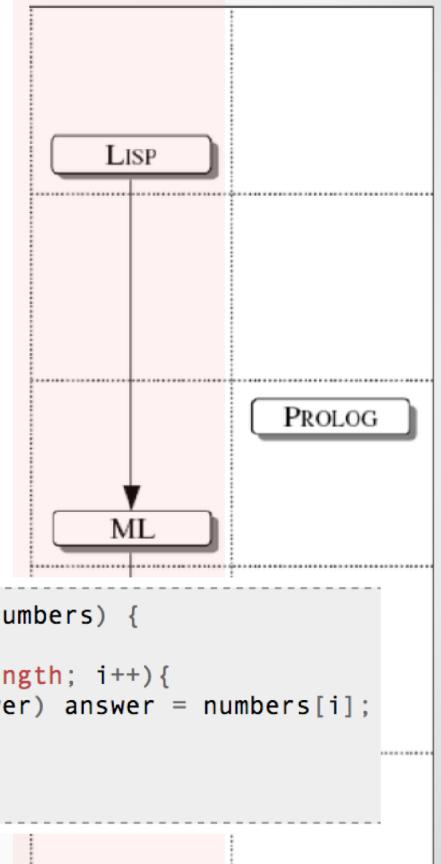
### Features:

- simpler semantics
- closer to mathematical functions

```
(defun get-max-value (list)
  (let ((ans (first list)))
    (do ((i 1 (1+ i)))
        ((>= i (length list)) ans)
        (when (> (nth i list) ans)
          (setf ans (nth i list)))))
```

```
public static int getMaxValue(int[] numbers) {
    int answer = numbers[0];
    for (int i = 1; i < numbers.length; i++){
        if (numbers[i] > answer) answer = numbers[i];
    }
    return answer;
}
```

functional  
languages      logic  
languages



# The Four Main Paradigms and their Features

## Logic/Declarative Paradigm

### Key idea:

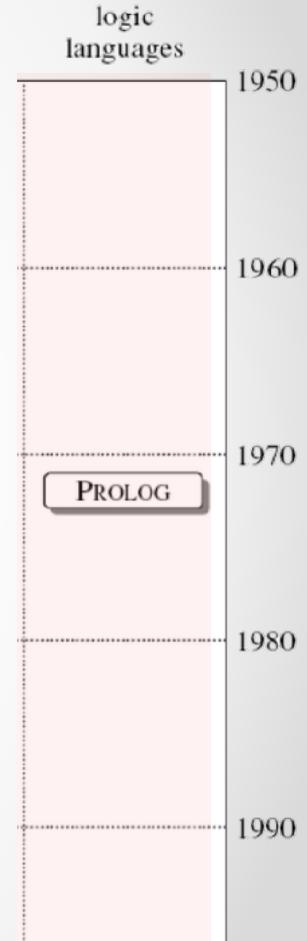
A program is a set of **facts** about objects,

**rules** about objects, and **questions** about the relations between objects.

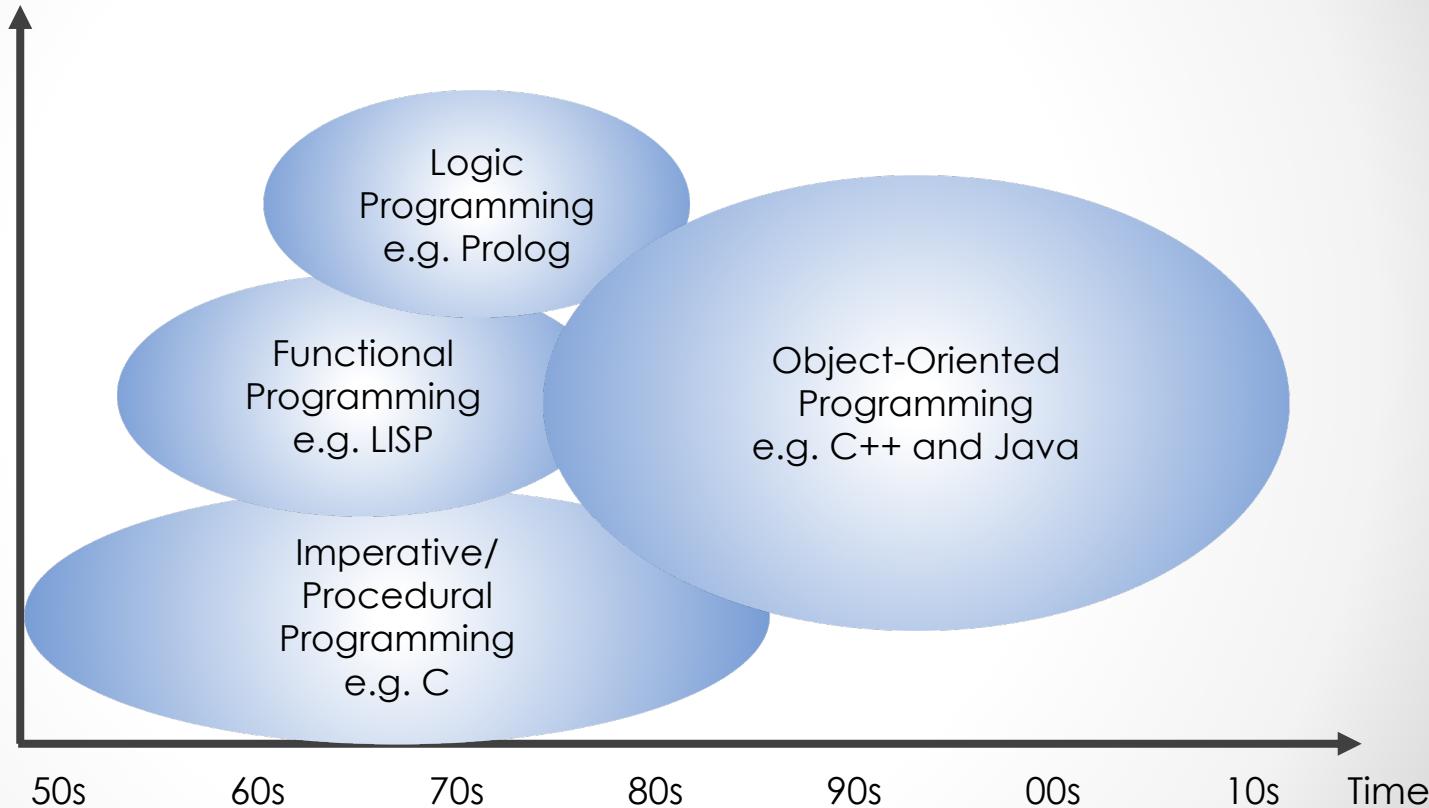
### Features:

- get rid of programming altogether.

```
male(james1).  
male(charles1).  
  
female(catherine).  
female(elizabeth).  
parent(charles1, james1).  
parent(elizabeth, james1).  
parent(catherine, charles1).  
  
% Was George I  
% the parent of Charles I?  
parent(charles1, george1).  
  
% Who was Charles I's parent?  
parent(charles1, X).  
  
% Who were the children of  
% Charles I?  
parent(X, charles1).
```



## Abstraction Level – Hiding complexity



Introduce yourselves (use the Discussion Board)

Read the Syllabus of the course

Read Textbook Chapter 1 (section 1.1)

Solve Quiz 1



## CSE2400 – Introduction to Programming Languages (online)

Javier Gonzalez-Sanchez

[javiergs@asu.edu](mailto:javiergs@asu.edu)

Fall 2017

**Disclaimer.** These slides can only be used as study material for the class CSE240 at ASU. They cannot be distributed or used for another purpose.