

---

# Object-Oriented Programming (OOP) vs Procedural Programming (PP)

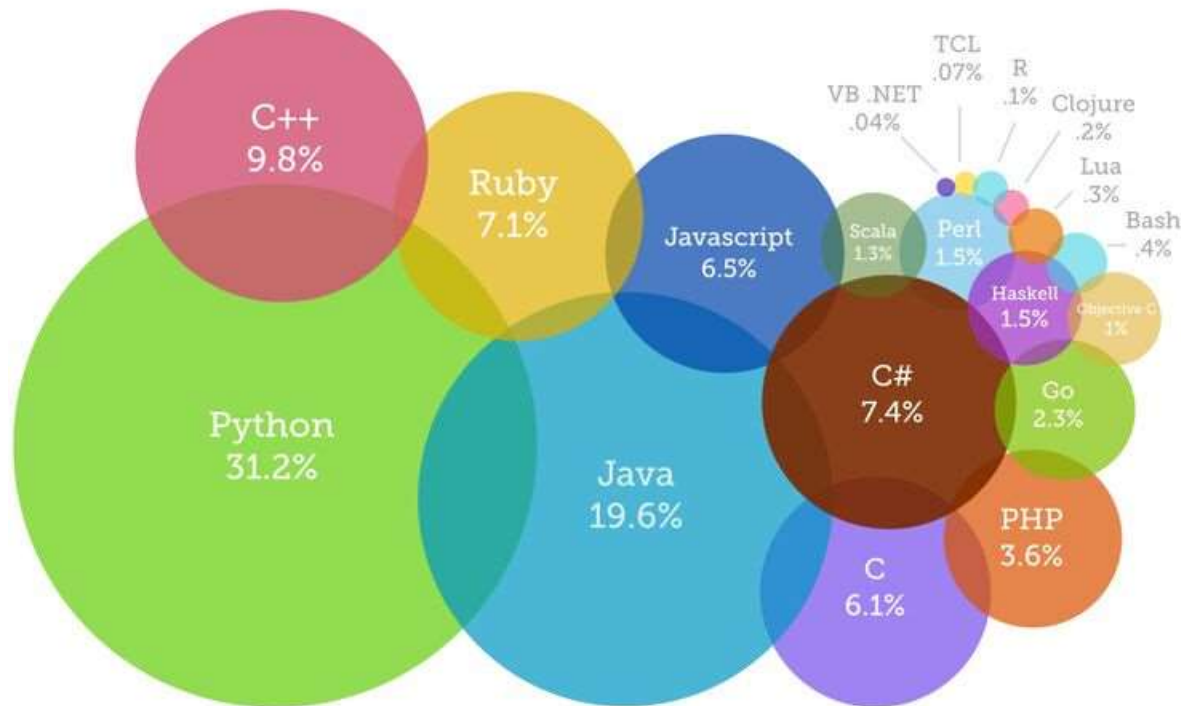
# Activity

---

- ◆ True or False
  - ◆ A C program is not an object-oriented program.
  - ◆ A C++ program is always an object-oriented program.
  - ◆ I can write an object-oriented program in C++.
- ◆ Name three OOP languages
- ◆ Name three PP languages

# Modern Programming Languages

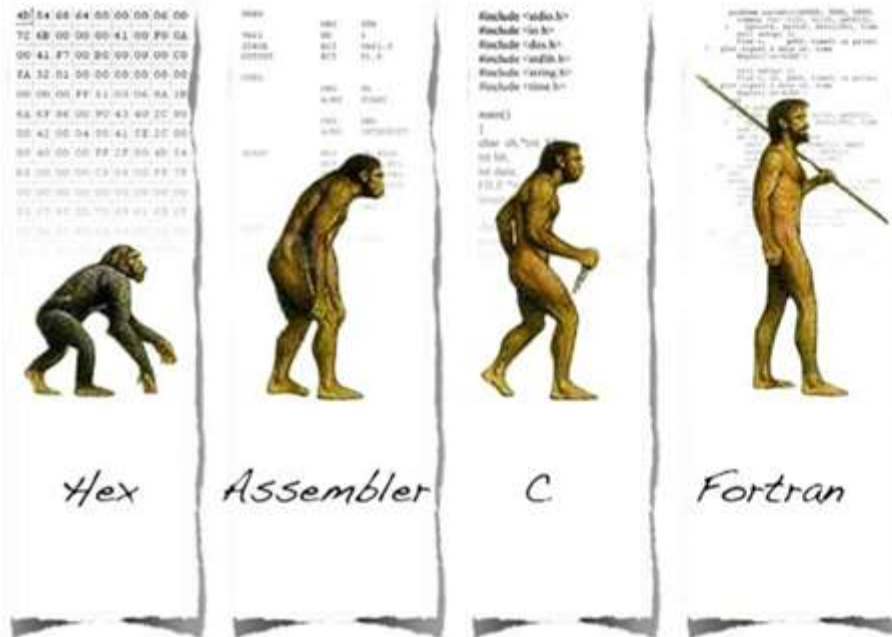
---



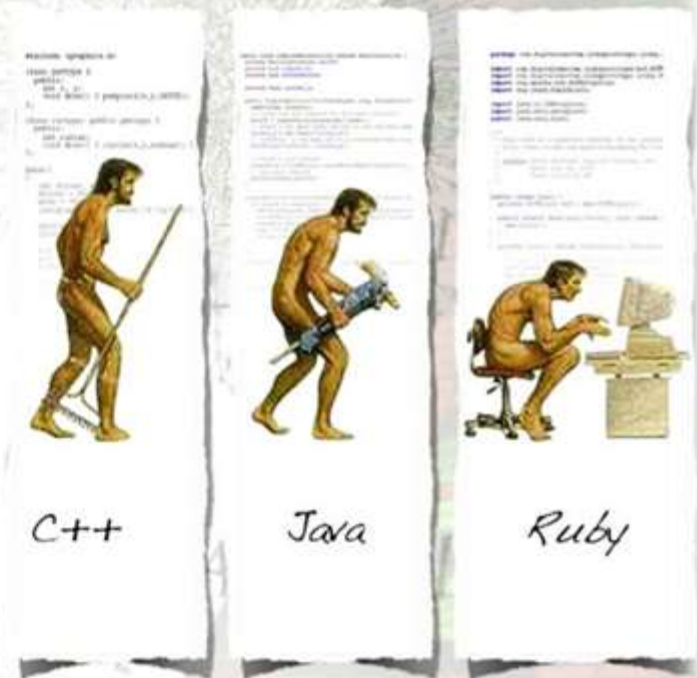
⑩ <https://www.devsaran.com/blog/10-best-programming-languages-2020-you-should-know>

# OOP Came After PP

# The Evolution Of Computer Programming Languages



# Procedural Programming (PP)



# Object-Oriented Programming (OOP)

# What is Procedural Programming

---

- ◆ *Procedural Programs:*
  - ◆ are made up of **procedures** (e.g., routines, subroutines, methods, or functions)
  - ◆ represent a list of instructions telling a computer what to do **step by step** and how to perform from the first code to the second code
- ◆ Traditional programming languages were procedural
  - ◆ C, Pascal, BASIC, FORTRAN

# Activity

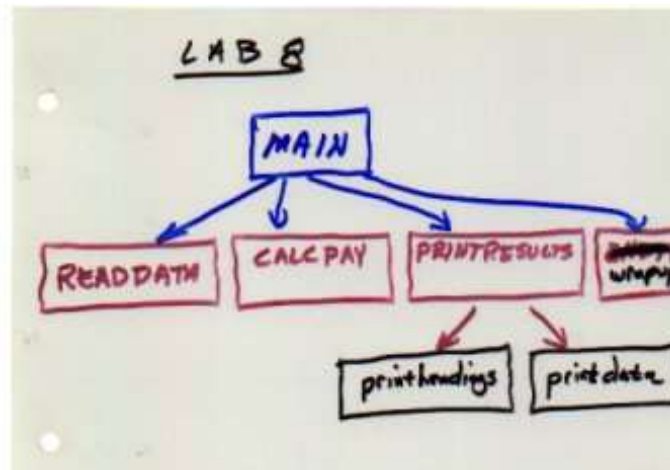
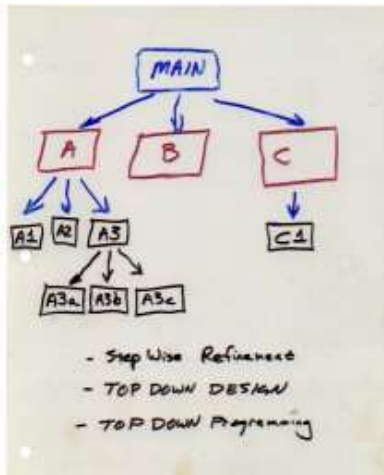
---

- ◆ Which program unit is an abstraction of “step by step” and “top-down” design?
- ◆ What we call this abstraction, i.e. what is the name of this abstraction?

# Procedural Programming Design

## ◆ Top Down Design

- ◆ **Functional decomposition** – a problem (procedure) is systematically broken down into sub problems (sub procedures)
- ◆ Functional decomposition continues until a sub problem is straightforward enough to solve



# Sample Code in Pascal

---

```
program TestSwap;

{
  Procedure:  Swap
  Input:      v1, v2 : Integers
  Output:     v1, v2 : Integers
  Description: Swaps the values of passed in Integers
}

procedure Swap(var v1, v2 : Integer);
var temp : Integer;
begin
  temp:=v1;           {sets temp to value v1}
  v1:=v2;             {sets the value of v1 to v2}
  v2:=temp;           {sets the value of v2 to temp}
end;

procedure Main();
var
  v1,v2 : Integer;
begin
  Write('Enter a number ' );
  ReadLn(v1);          {Read user input and assign value to v1}
  Write('Enter another number ' );
  ReadLn(v2);          {Read user input and assign value to v2}
  WriteLn ('v1 is :',v1,' and v2 is :',v2);
  Swap(v1,v2);         {Calls Swap Procedure}
  WriteLn ('v1 is now :',v1,' and v2 is now :',v2);
end;

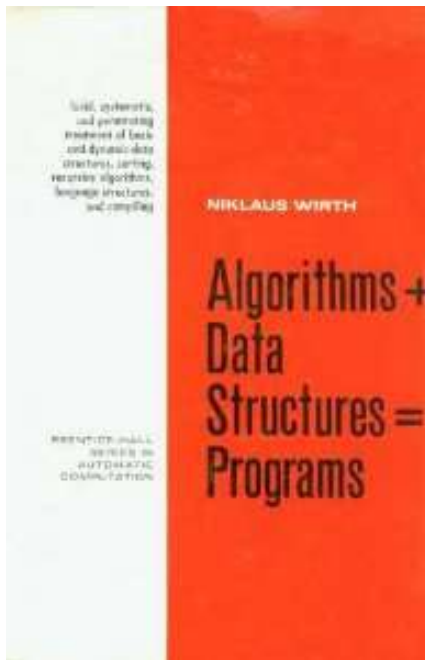
begin
  Main();
end.
```



# Procedural Programming Design

---

- ◆ *Algorithms + Data Structures = Programs*
  - ◆ Algorithms are implemented in **procedures**
  - ◆ Procedures take specific data structures as input and generate other types of data as output



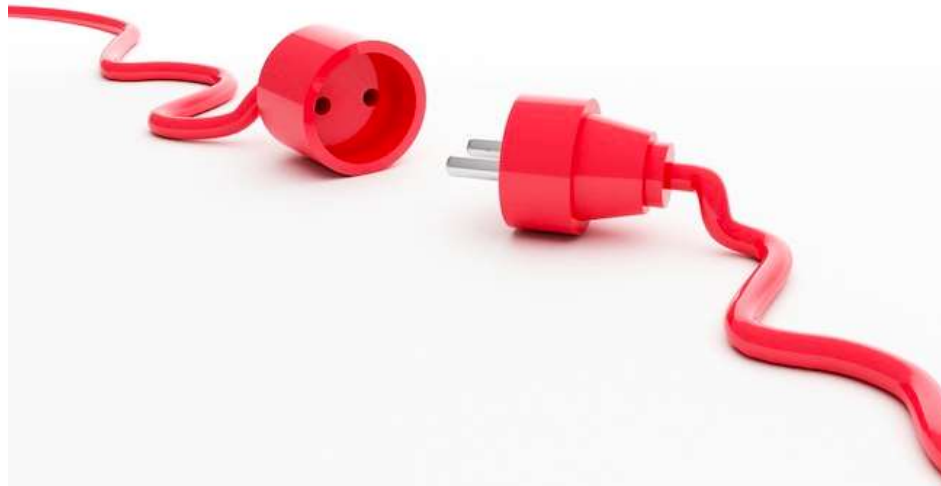
---

## ◆ Side note

- ◆ CS department's curriculum is designed based on PP
- ◆ **Activity:** CSI400 – did the instructor teach Java using objects first approach?

# Problems with PP - Expressiveness

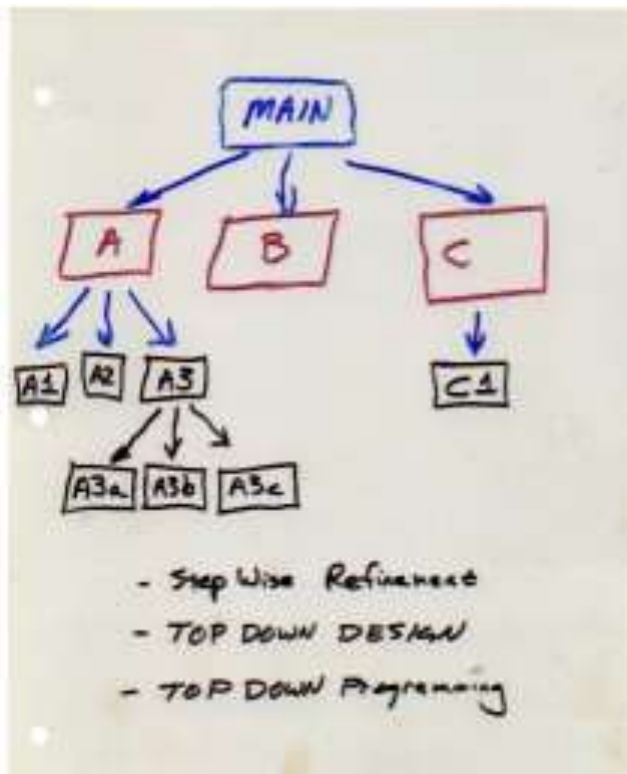
---



- ◆ Procedural languages are difficult to relate with the real world objects.
- ◆ Disconnected from real world problems.

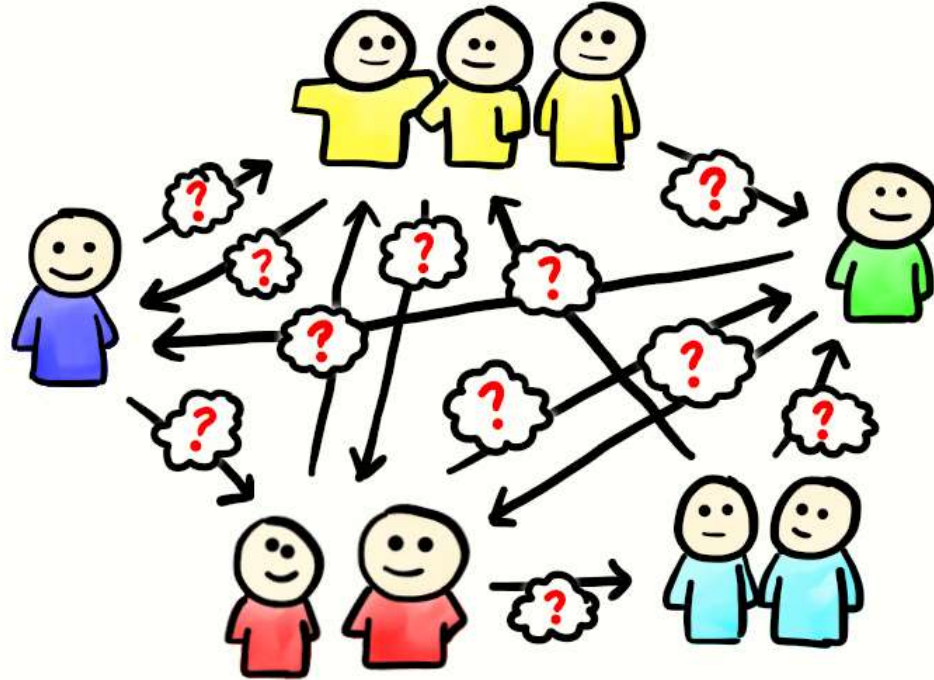
# Problems with PP - Extensibility

- ◆ Top-down design does not scale with top-level changes
- ◆ Changes in algorithm or data structures affect each other



# Problems with PP – Security

---



- ◆ Data used in procedural languages are exposed to the whole program (procedures)
- ◆ No security for the data

# Activity

---

- ◆ What are problems with PP?
- ◆ Do you agree with the above assessment of PP?

# Activity

---

- ◆ What are problems with PP?
  - ◆ Problems with PP
    - ◆ Expressiveness
    - ◆ Extensibility
    - ◆ Security
- ◆ Do you agree with the above assessment of PP?

# Motivation for OOP

---

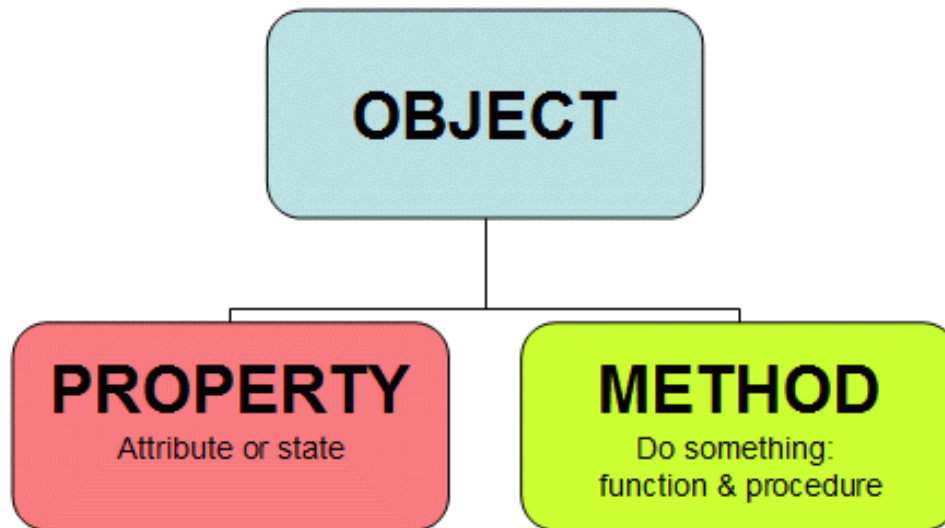
- ◆ Real world connection
  - ◆ We build software to solve real world problems.
  - ◆ People have real world problems.
  - ◆ *Therefore, we build software for **people**.*
- ◆ Extensible
  - ◆ Good software not only solves immediate problems, but it can be maintained and modified to address the inevitable **changes** that the customer will want.
- ◆ Secure
  - ◆ Exposing data to everywhere prevents the software from being the extensible good software. More importantly, it is not **secure**.



# OOP

---

- ◆ A class is a combination of **state (properties)** and **behavior (methods)**
- ◆ An object is an instance of a class

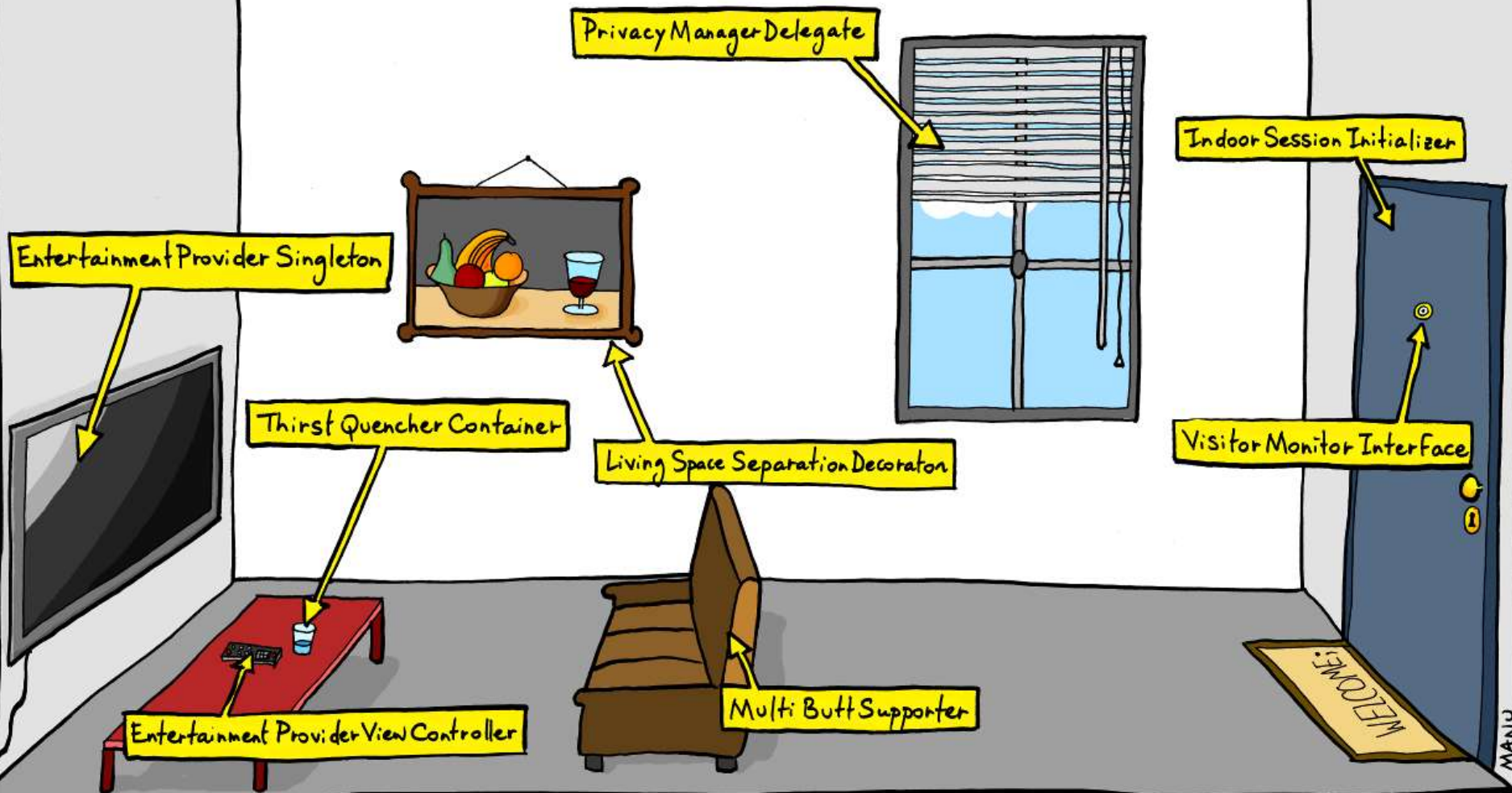


# OOP Connects Real World Objects

- ◆ An inheritance solution requires subclasses to represent each type of view



# THE WORLD SEEN BY AN "OBJECT-ORIENTED" PROGRAMMER.



---

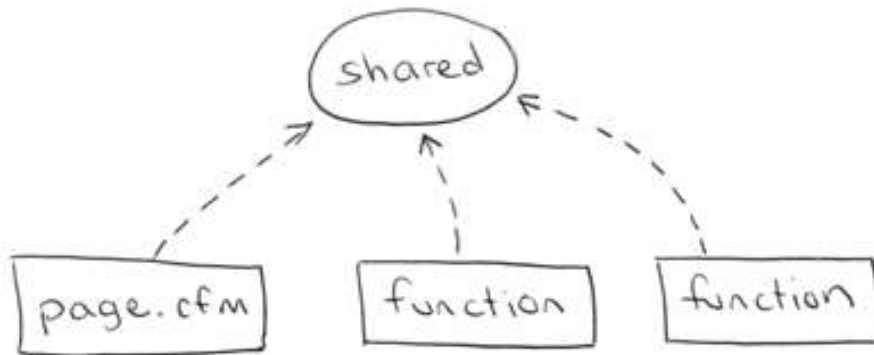
## ◆ Activity

- ◆ What are classes?
- ◆ What are objects?
- ◆ What kind of abstraction that OOP promotes?

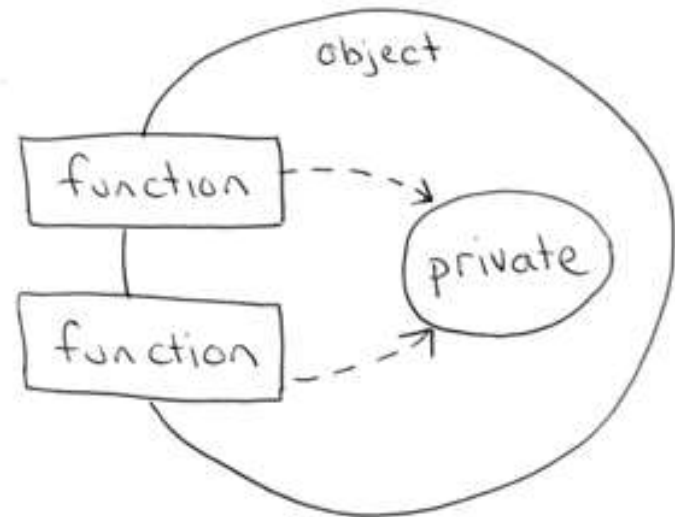
# OOP Encapsulates Data

---

- ◆ The ability of an object to **hide** its data and methods from the rest of the world



PP shares the data globally



OOP encapsulates data privately

# OOP Supports Better Modularity

---

- ◆ The source code for a class can be written and maintained independently of the source code for other classes
- ◆ Once created, an object can be easily passed around inside the system





# OOP Improves Reuse and Extensibility

---

- ◆ If a class already exists, you can use objects from that class in your program.
- ◆ Object oriented programming languages allow classes to inherit commonly used state and behavior from other classes



# Using Java Doesn't Mean Using OOP

---

- ◆ Using OOP means:
  - ◆ Data encapsulation
  - ◆ Information Hiding
  - ◆ Model problems using objects
  - ◆ Polymorphism





# Disadvantages of OOP

---



Object-oriented programming is an exceptionally bad idea which could only have originated in California.

(Edsger Dijkstra)

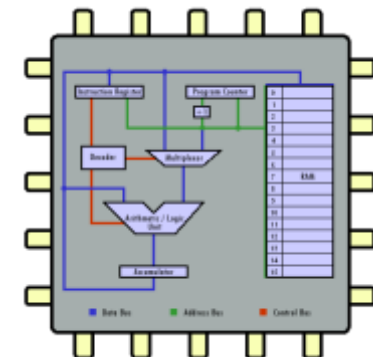
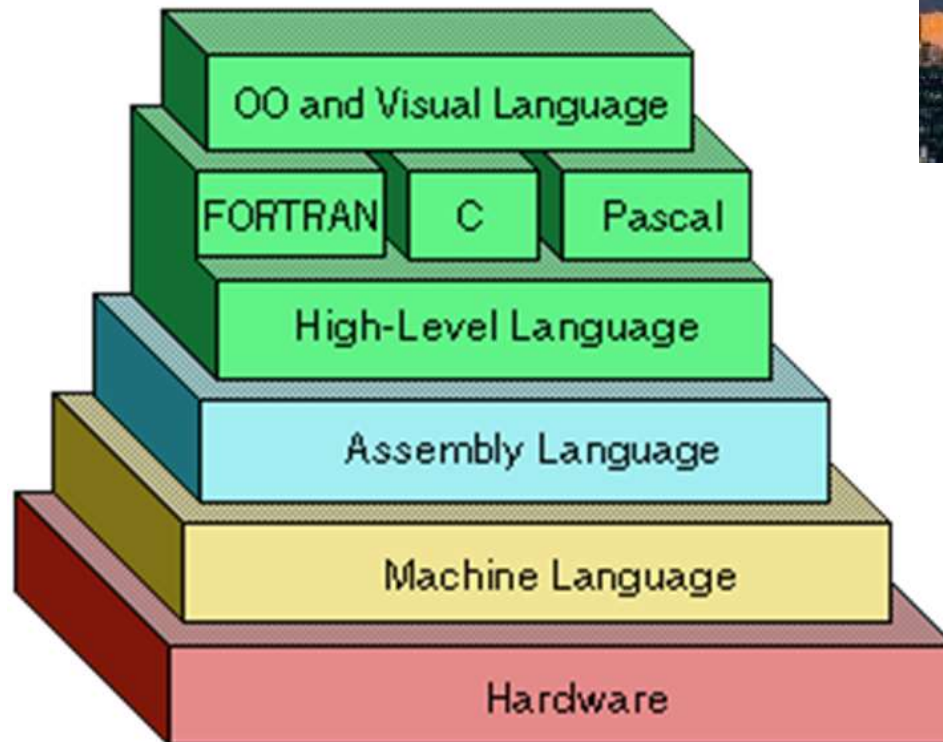
izquotes.com

# Disadvantages of OOP

---

- ◆ Not every problem can be considered as objects
  - ◆ e.g., Procedure is not an object
- ◆ Steep learning curve
  - ◆ PP (Top-down) is more straightforward for us to think
- ◆ Large program size
- ◆ Slower programs / Less efficient

# Level of Abstraction



# Activity

---

- ◆ Did this talk convince everyone that OOP is our current and future trend?
- ◆ True or False
  - ◆ At least this talk makes me love OOP more.

*Count my votes for OOP?*

- 
- ◆ Let's have more OOPs



- ◆ End of Lecture PP vs. OOP

