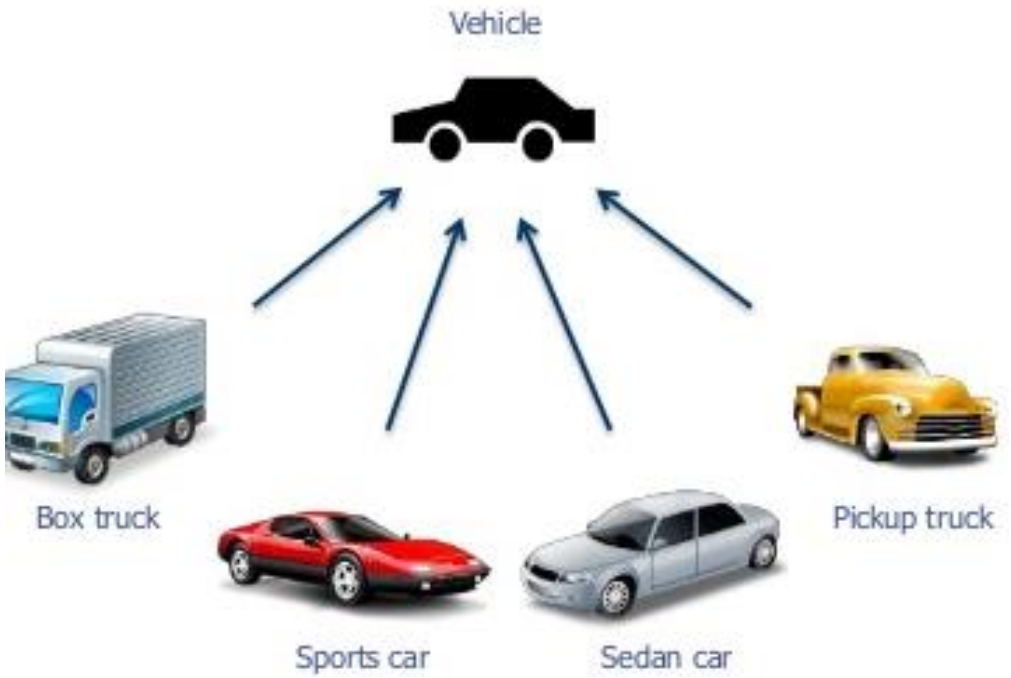
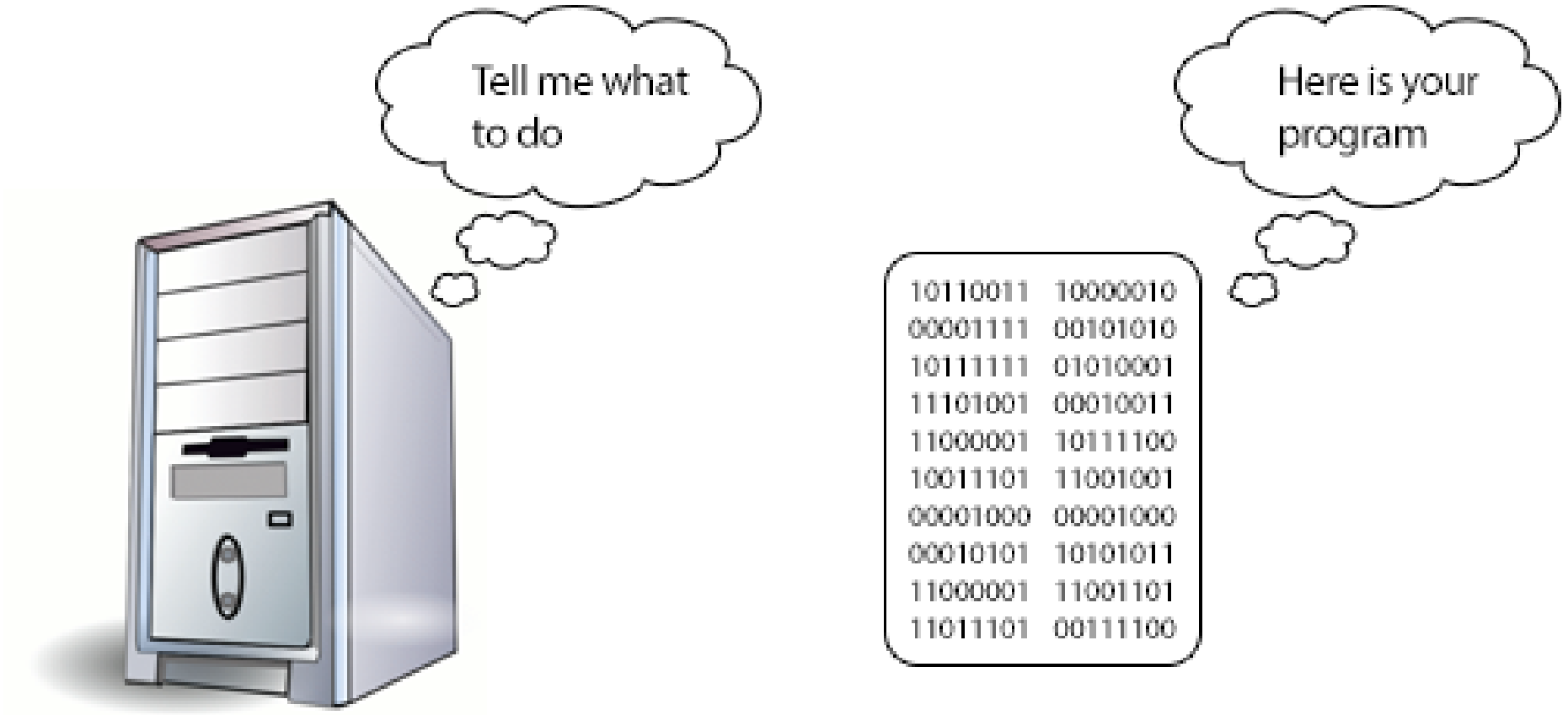
Introduction to Object Orientation



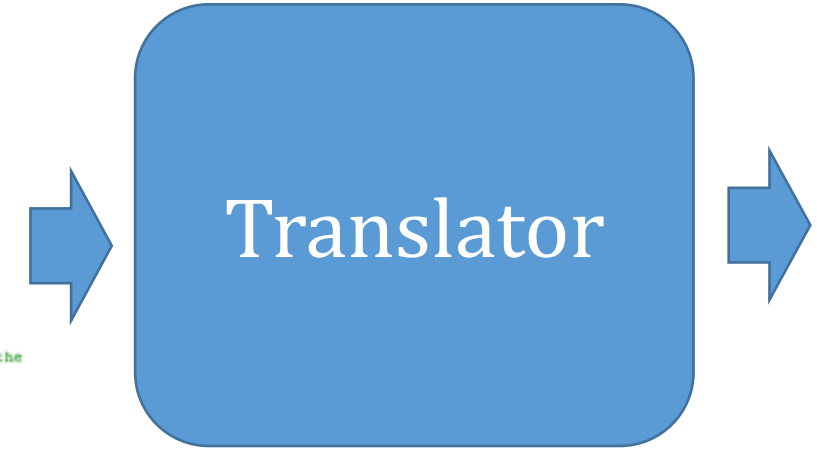
Computer Program

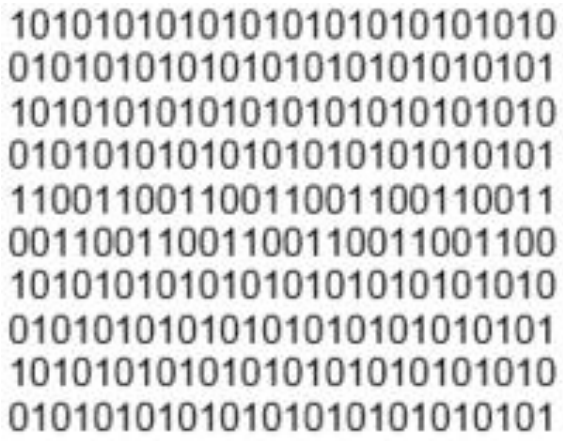


Programming Language

- It's really hard to write programs using the binary machine language
- Write a program in a more readable "Programming Language" and translate that programming language into binary machine language (on Windows, .exe file)

```
'//Parses <ref> tags in wikitext, code by Mike Billington
Private Function ParseRefs(text As String) As String
  Dim a() As String, i As Long, tmpRefs As String, tmpText As String
  a() = Split(text, "<ref>")
  For i = 0 To UBound(a())
    If i = 0 Then
      tmpText = tmpText & a(i)
    Else
      If InStr(a(i), "</ref>") = 0 Then
        tmpText = tmpText & "<big><font color='red'><b>&ltRef&gt not closed" & _
          "</b></font></big>" & a(i)
      Else
        tmpRefs = tmpRefs & "#" & getL(a(i), "</ref>") & vbCrLf
        tmpText = tmpText & "<sup>[" & i & "]</sup>" & getR(a(i), "</ref>")
      End If
    End If
  Next i
  tmpText = Replace(tmpText, "<references/>", tmpRefs) '// Yes, I know that isn't the
  tmpText = Replace(tmpText, "<references />", tmpRefs) '// "right" way....
  tmpText = Replace(tmpText, "<references>", tmpRefs)
  ParseRefs = tmpText
End Function
```



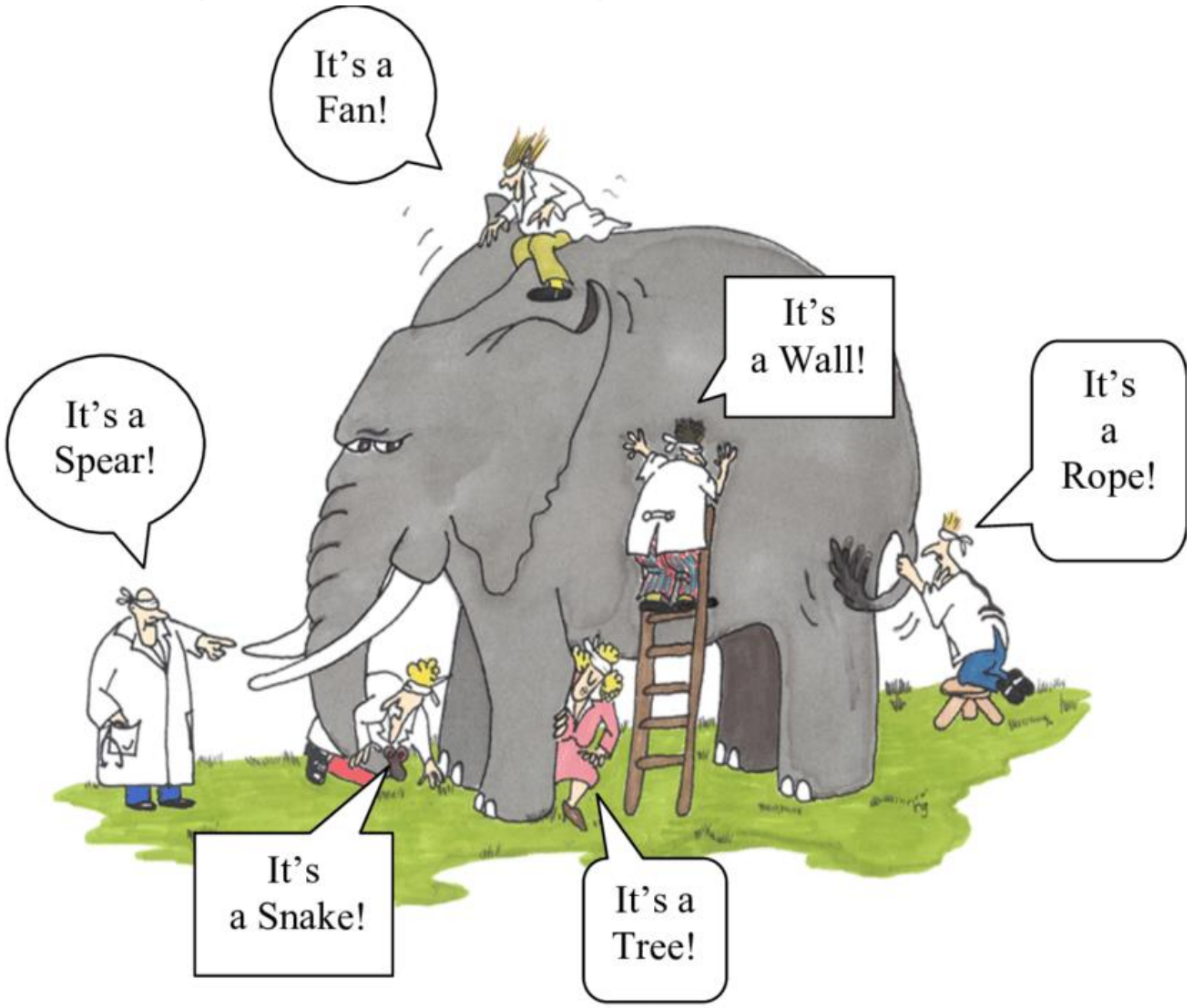


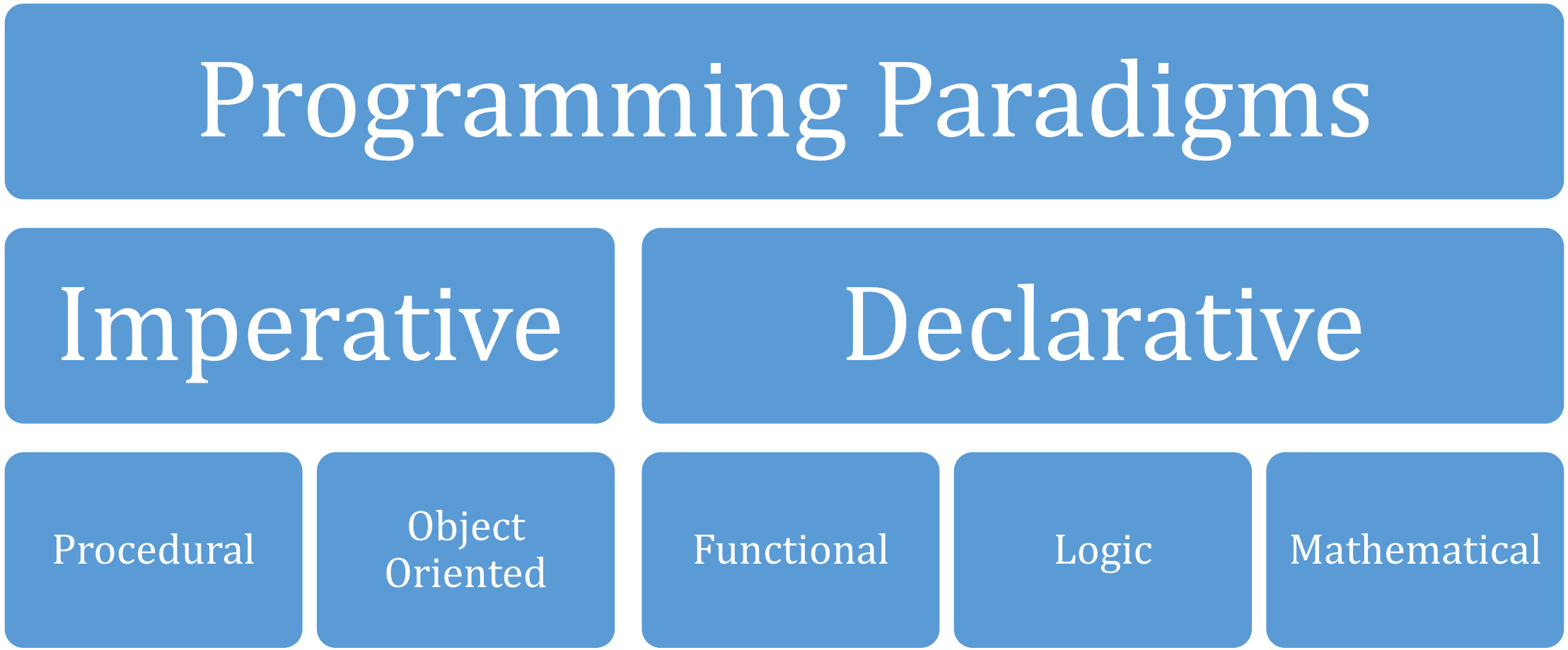
What's in a Language?

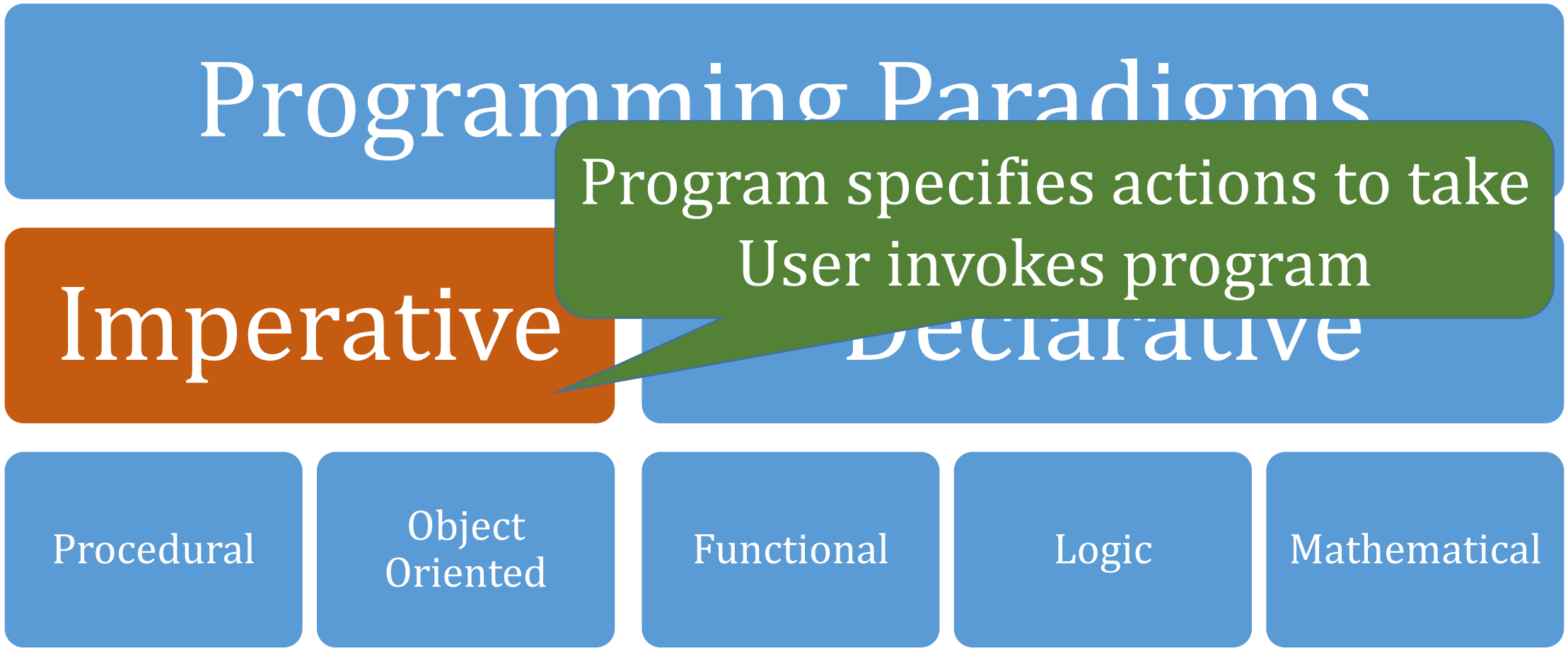
- Why learn different languages?
  - They are all expressive
- Some languages are easier to code than other languages
  - APL: $x \leftarrow 3\ 4\ \rho\ \iota\ 11$
- Some languages yield code that performs better
- Some languages have really useful libraries
- Trade-offs between complexity and effectiveness

http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html

Programming Paradigm

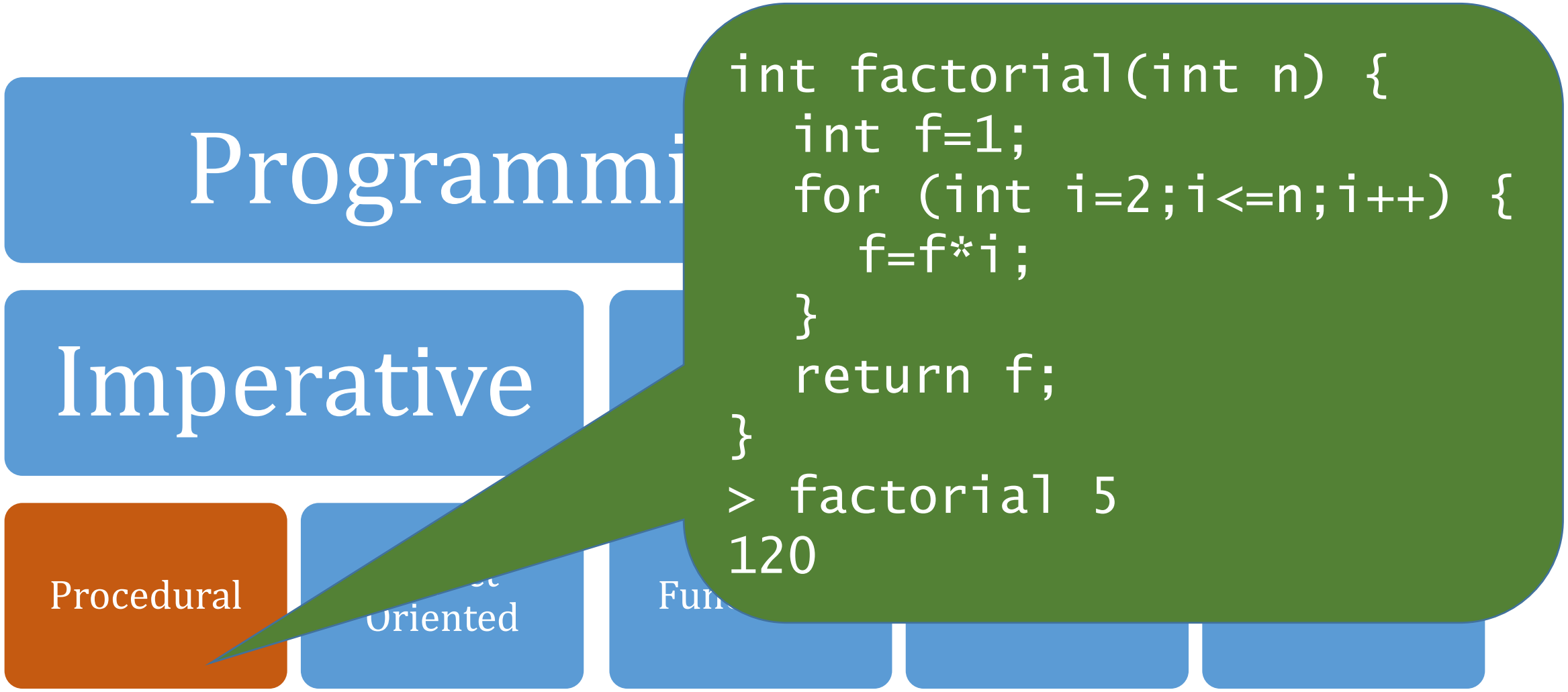


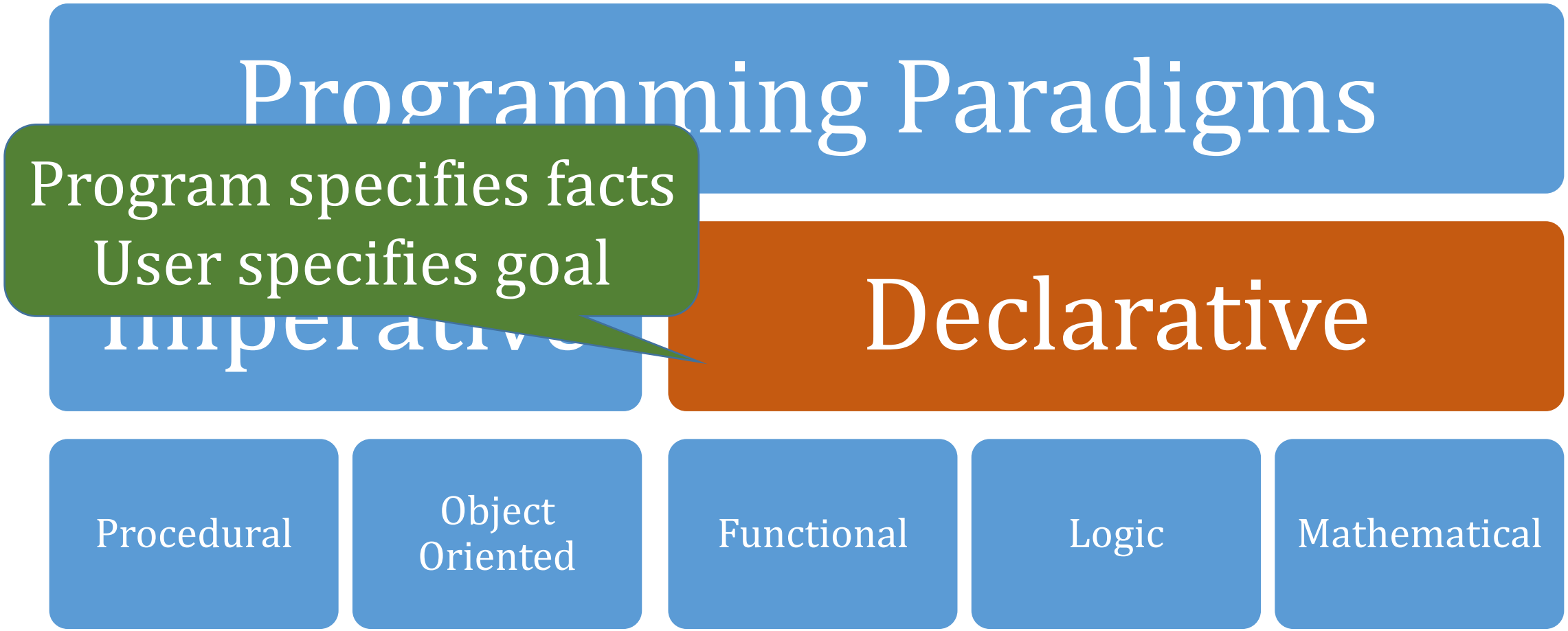

Programming Paradigms
Imperative
Declarative
Procedural
Object Oriented
Functional
Logic
Mathematical


Programming Paradigms
Imperative
Declarative
Program specifies actions to take
User invokes program
Procedural
Object Oriented
Functional
Logic
Mathematical

Imperative Programming

- Imperative: “giving an authoritative command”
- The kind of programming we are doing uses the following:
  - We need to work with variables
  - We need to give them values (assignment)
  - We need conditional statements
  - We need loops
  - We need functions
- Imperative programming: “Commanding” the computer to solve our problems by writing a computer program


Programmi
Imperative
Procedural
Oriented
Fun
int factorial(int n) {
int f=1;
for (int i=2;i<=n;i++) {
f=f*i;
}
return f;
}
> factorial 5
120


Programming Paradigms
Program specifies facts
User specifies goal
Imperative
Declarative
Procedural
Object Oriented
Functional
Logic
Mathematical

Prograr

Imperative

```
factorial(x){
   x<2: x;
  x>=2: x*factorial(x-1);
}
>factorial(5)
120
```

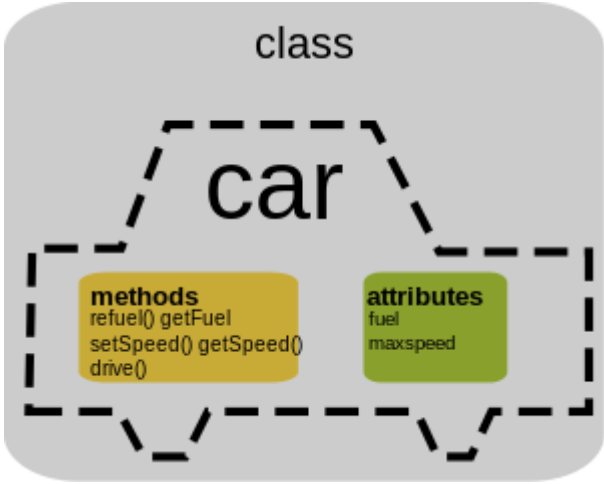
Procedural | Object Oriented | Functional | Logic | Mathematical

Object Oriented Programming

- Sees the entire world as "objects" – computer models of real things
- "Object" dictionary definition:
  - a material thing that can be seen and touched
  - a person or thing to which a specified action or feeling is directed

- Objects are defined by two things:
  - What is the data that describes the object: fields
  - What are the actions that can be performed on the object: methods

Object Orientation – Why classes?

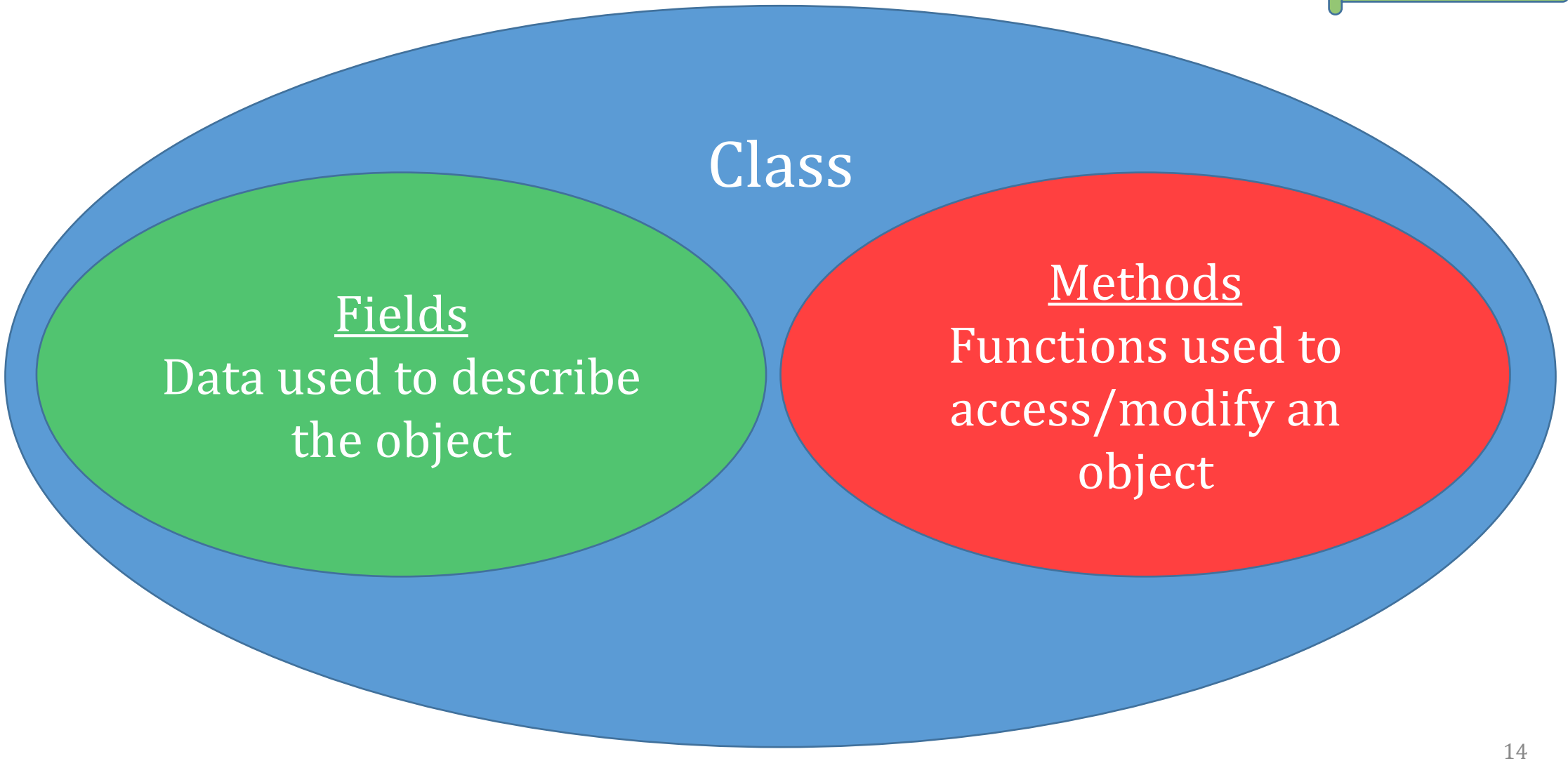


- Similar objects share:
  - Same list of fields (attributes)
  - Same list of actions

- Group similar objects into a "class"
  - a set or category of things having some property or attribute in common and differentiated from others by kind, type, or quality
  - class is the abstract view of ***any*** object in that class
    - Actual physical object or
    - Hypothetical object

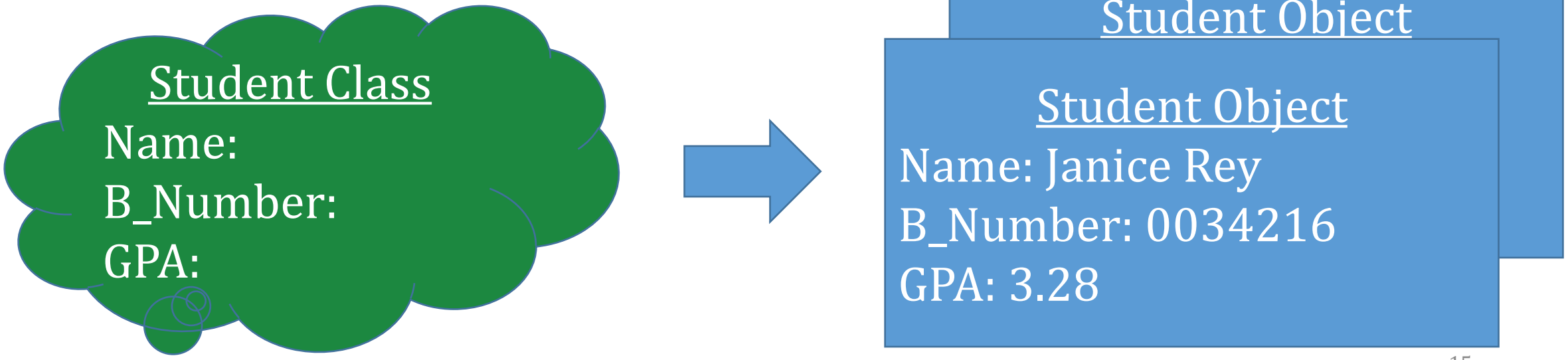
What's "in" a class?

Chap 2.1



Object Instantiation

- The process of creating an object of a specific class
- Often requires specification of object attribute values
  - Some attributes can have default or automatically generated values
- Student Janice Rey is one "instance" of the Student class



Programm

Imperative

Procedural

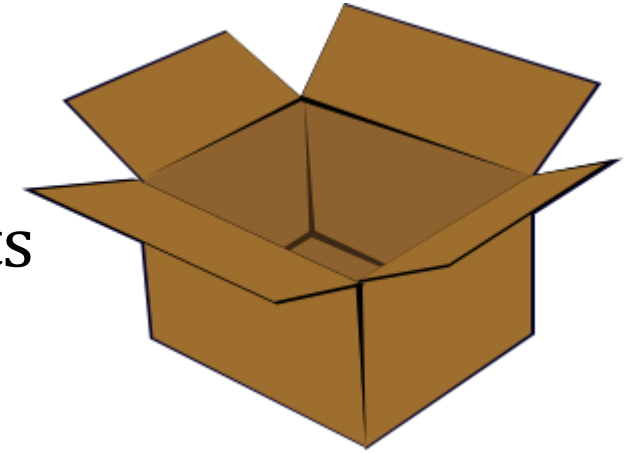
Object Oriented

```
class Factorial {
        int n;
        Factorial(int n) { this.n=n; }
        int value() {
                int v=1;
                for(int i=2;i<=n;i++) { v=v*i; }
                return v;
        }
        public static void main(String args[]) {
                int n=Integer.parseInt(args[0]);
                Factorial f=new Factorial(n);
                System.out.println("Factorial " + n +" = " + f.value());
        }
}
> java Factorial 5
Factorial 5 = 120
```

Object Oriented Design – First Pass

- Think about the kinds of objects you want to model
- Classify those objects
  - Divide the objects into classes – objects which share data and methods
- Define each class, tell the compiler:
  - The name of the class
  - The attributes used to define the class
  - The actions that work on objects in that class
- Then write imperative code to implement methods (actions)
  - instantiate and manipulate objects

Object Oriented Encapsulation



- Concept: Leave dealing with cars up to the car experts
- If you aren't a car expert, don't go under the hood!
- Make one place the "auto-mechanic" place
- That place has the only code that modifies car objects!
- If any one outside of that place wants to interact with cars, it has to invoke a service provided by the expert

Bad Object Oriented Design

- The “Hello World” function is imperative, there is no “object” or “class” associated with that function
- All java code must be in a class!
- Abuse OO design: create a “class” of “HelloWorld” objects
  - makes no sense, but we need to do it this way to satisfy java rules

Why Object Orientation?

- Imposes structure on design
  - Forces everything into an object/action way of thinking
  - Reduces the number of choices we need to consider
- Establishes Responsibility / Traceability
  - If a “car” object has inconsistent values, there is a bug in the car class… it can’t be anywhere else!
- Establishes Areas of Expertise
  - Go to the car mechanic to get our car fixed… she knows how to fix it
- Re-Use
  - I can use the same classes in different programs