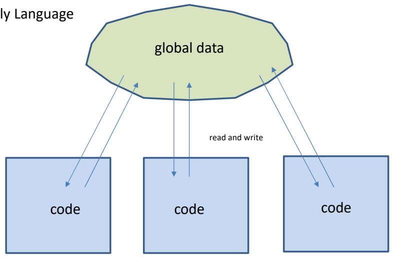
Object-Oriented Programming (OOP)

Why Objects?

what problem gave rise to objects as a solution?

Basic Programming Model

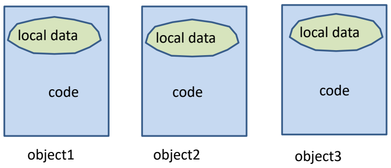
- Assembly Language
- Fortran
- Cobol
- C



Dave Parnas (Early pioneer of Software Engineer):
Came up with the concept of "*Software Aging*"

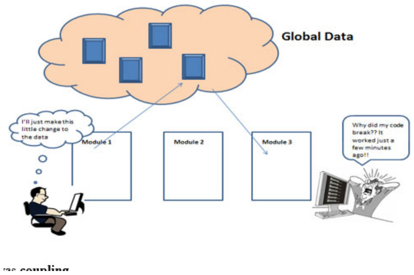
Object Model

- Combine data and code into entities called objects
- Distribute the data to where the code lives that reads and writes that data
- Don't allow other objects to directly access the data *owned* by another object



BUT, what if object3 needs the data held by object1?

Provide functions (methods in OO speak) to allow other objects to access the data. These methods are called get and set methods.


Global Data
I'll just make this little change to the data
Module 1
Module 2
Module 3
Why did my code break?? It worked just a few minutes ago!!
coupling

Evolution of Object Oriented Languages I

- 1967
  - Simula

Simula - Introduced the **Class** concept – a template for creating objects that all had the same data and code. Used to build simulations

- OOP was introduced in the mid-1960s with Simula 67, a programming language designed to model and simulate Norwegian Oil Tankers traveling around the world.
- Simula was created at Norwegian Computing Center in Oslo.

- Simula introduced the notion of classes and instances as part of the OO programming paradigm.

This concept of Objects based on Classes led to a variety of OO languages...

Object Oriented Languages II

- 1978 - 1980
  - Smalltalk @ Xerox Parc
- 1979
  - C++ Bjarne Stroustrup
- 1983
  - Objective-C
- 1995
  - Java: Sun Microsystems
- 1995
  - JavaScript

Smalltalk – Xerox ran a research institute that gave us the mouse, graphic interfaces and an OO language. Steve Jobs' visit was inspiration for the Macintosh.

C++ added objects to the C language

Objective-C was the language for the new Mac

Java was THE OO language that could run on any platform due to the Java Virtual Machine

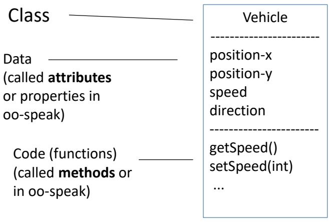
JavaScript – not designed as an OO language but the authors thought that if it had 'Java' in the name, it would be popular .. ha ha

The Big Three Concepts of OO

1. Encapsulation
2. Inheritance
3. Polymorphism

Encapsulation

- An Object is an <u>encapsulated</u> package of <u>data</u> and <u>code</u>
- An Object is data structure with associated code



Class

Data (called **attributes** or properties in oo-speak)

Code (functions) (called **methods** or in oo-speak)

Vehicle

----

position-x
position-y
speed
direction

----

getSpeed()
setSpeed(int)
...

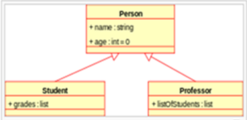
Do other objects have direct access to the data of another object?

In Java and C++ you declare object data as **private** or **public**. In Python it's all **public**.

the methods of an object have access to the data of the object

Inheritance

Inheritance



Class diagram showing generalization between the superclass *Person* and the two subclasses *Student* and *Professor*

This feature was seen as the pathway to flexible code. In practice, it often leads to very difficult to understand code. Beware! Use carefully

Polymorphism

- **Polymorphism** is a fancy word that means when you tell an entity to do something, what gets done depends on the receiver of the message.
- For example, in the real world, the command “cut” when given to different individuals has different effects.

The surgeon
The hair stylist
The actor
Cut

The surgeon would begin to make an incision.

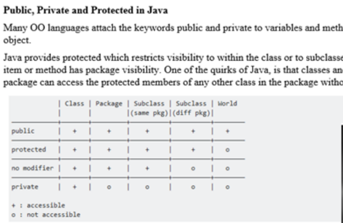
The hair stylist would begin to cut someone's hair.

The actor would abruptly stop acting out the current scene, awaiting directorial guidance.

Polymorphism = same method name, different behavior depending on the receiver

Complexity of OO Encapsulation in Java and C++

Java



Public, Private and Protected in Java

Many OO languages attach the keywords public and private to variables and methods to control access from outside an object.

Java provides protected which restricts visibility to within the class or to subclasses. If no modifier is provided, the data item or method has package visibility. One of the quirks of Java, is that classes and objects that share the same Java package can access the protected members of any other class in the package without being a subclass.

	Class	Package	Subclass (same pkg)	Subclass (diff pkg)	World
public	+	+	+	+	+
protected	+	+	+	+	o
no modifier	+	+	+	o	o
private	+	o	o	o	o

+ : accessible
o : not accessible

Table of Visibility for Java

C++

- C++ also has public, private and protected.
- However, C++ also has its own quirky behavior regarding violating encapsulation for private and protected members.
- In C++, one may define a **friend** function of a class outside that class' scope but with the right to access all private and protected members of the class.

Python

- Python has Objects and Classes BUT does NOT have public, private and protected syntax
- Python is not a pure object-oriented language
- Functions play a BIG role in Python programming
- All data that is part of an object can be accessed by any other part of a program
- Python provides guidelines for controlling access to data in objects but it is not enforced by the compiler