Let's write some code.

- Write a *procedural* program for calculating the average salaries for each department (*Accounting, HR, IT*) in a company.

- Assume I have the following input file:

```plaintext
1, Joe, Smith, HR, 60000
2, George, Jetson, Accounting, 50000
3, Bill, Cannon, HR, 45000
4, Fred, Flintstone, IT, 65000
5, Barney, Rubble, Accounting, 55000
6, James, Kirk, IT, 70000
7, John, Doe, HR, 30000
8, Jane, Doe, IT, 85000
9, Marge, Simpson, HR, 35000
10, Alice, Krabden, Accounting, 42000
```
Object Oriented Concepts

- In some cases we’ll learn the concepts prior to learning the terminology.
- Other times we’ll hear the terms, before we know what they mean.
- In either case, what we call the concepts are just as important as the concepts themselves.
- If you want to talk “OO” with others, you will need to learn the vocabulary.

I am Dungeon Master, your guide in the world of Object-Orienting Programming

Sometimes by looking back you can see a clearer path through what lies ahead.

Just follow that path. But beware. You must never touch - the beauty - that breathes the beast.

The right road is not the left.

Beware, for only beauty can defeat the eye of the beholder.
What makes a language an **Object-Oriented**?

- The goal of OOP is **abstraction**.
- In general you will see three big terms used:
  - Encapsulation
  - Inheritance
  - Polymorphism (this has multiple meanings)

Weisfeld adds an additional one:
- Composition

Procedural Programming consists of writing **procedures** that take in some data as **input** and provide **output**.

- Data is **distinct** from the process. It has no relation to the procedure.
- Programs **process** data as defined by the **procedure**.
- Likewise the **functions** written in the program have no direct **relationship** with each other.
- Programmer is all knowing, beginning to end.
Processing Data

- My first job after college was programming in COBOL.

- The department I worked in was called Administrative Data Processing (ADP).

- I wrote very long programs that would process data.

How is OOP different?

- In Object Oriented Programming (OOP) the data and the functions that operate on that data are bundled together into an object.

  - Objects are self-contained entities that maintain their own data.

  - A program uses objects that interact to solve a problem.

  - Abstraction hides the unnecessary details from the programmer.
What is not OO programming?

- You can write **procedural code** in any language, including Java.

- Simply using **functions** or **classes** in your program does **not** make it Object Oriented.

Procedural data design.

- In addition to the parameters, all the functions share access to the **global** variables.

- If data is **changed** by FunctionA, the other functions use the **new** value.
Object Oriented data design

- The data and related functions are *encapsulated* together.
- The *object* is a self contained *entity*, and other objects cannot inadvertently *change* it’s data.

Learning OO

“The most difficult problem in teaching *object-oriented* programming is getting the learner to give up the *global knowledge* of control that is possible with procedural programs, and rely on the *local knowledge* of objects to accomplish their tasks.”

(Beck and Cunningham OOPSLA '89)
What did you come up with?

- A **procedural** approach may take the form of:
  - Read the file, line by line and load the data into variables or perhaps an array or list.
  - Once the data is loaded calculate the average salary for each department, by summing up and dividing by the number of individuals in the department.
  - Print out or display the resulting calculations.

What would an Object Oriented approach look like?

- Use a `FileLoader` object to load the data and create objects (Departments and Employees).
- Each `Employee` object knows information about itself, like its id, name, salary.
- Each `Department` object knows information about itself, like which Employees are in the department.
- Once the data is loaded I can have each Department object provide me with the average salary of its employees.
In OO terminology objects in our program interact by sending messages to each other. How do objects interact?

- FileLoaderFrank, please load the Edit1.txt file.
- Ok. Let me introduce you to Department.Accounting, Department.HR and Department.IT
- Department.Accounting, what is your average salary?
  - $49,000
- Department.IT, what is your average salary?
  - $73,333.33

Another program

- Using a **procedural** design, what would I do if I wanted to create a program that found the maximum salary in a dept?
  
  Most likely you will copy the program and change it to use new calculations.

- How does the same problem work with my OO design?
  
  I can **reuse** my existing objects, and just add a **new behavior** to the department object.
One more example...

- Create a procedural program that takes input from the user as to their weight, and outputs to the screen their weight on the other planets.

- The formulas you will need are as follows:
  - Mercury = .37 * weight
  - Venus = .876 * weight
  - Mars = .381 * weight
  - Jupiter = 2.637 * weight
  - Saturn = 1.151 * weight
  - Uranus = .79 * weight
  - Neptune = 1.12 * weight
  - Pluto = .025 * weight

- weight = the weight on Earth.

procedural View

- Mercury = .37 * weight
- Venus = .876 * weight
- Mars = .381 * weight
- Jupiter = 2.637 * weight
- Saturn = 1.151 * weight
- Uranus = .79 * weight
- Neptune = 1.12 * weight
- Pluto = .025 * weight
Calculations are distributed and contained within various objects.