we will analyze what is good design

95% people in industry don't know how to program

What is a program?
- piece of code
- representation
- design
- solution to a problem
- executable model of a solution to a problem
- sequence of statements
- computational simulation of a domain

Why not "a bunch of objects"?

Programmer sees problem domain from a viewpoint

translates to a computational model

want to minimize the semantic gap

when things change in the real world, you need to make a similar change in your model

What is a good program?
- fast
- runs flawlessly
- easy to use
- fulfills requirements
- flexible
- easy to change or evolve
- easy to maintain

Why objects? Why OOP?

easier to reduce semantic gap between program and reality

What is an OO program?
- set of objects that collaborate by sending messages

Telco modeling example

goal to come up with a maintainable design

need to find the objects

will use UML "object graph"

Objects
- aCall
- aCallClassifier
- aPhoneNumber
- aTelcoManager

Naming is important

Model an object's responsibility
- what messages does it respond to?
- for each responsibility, why do we need it?

price
- Does this really belong here?
- Perhaps a call should not know its price

caller
- need to know if the call is at peak hour

receiver
- to compute cost

duration
- to compute cost

type
- international
- national
- local

Later we will make this a responsibility of another class

aTelcoInvoiceSystem

"I worked in a company where, if you wrote a class with Manager in its name, you got fired."

"End up with "a bag of cats""

name does not evoke its responsibility

To avoid the "bag of cats" problem, we model "the system" as a composition of other objects

Scenario
- International call
- Find price
- Find caller
- Prepare invoice
- Print it

processInvoices

aPriceCalculator
- priceFor(aCall)
- Need to know the type of the call to compute the price
- If we make the price calculator an object, we can replace it by a different price calculator (!)

aCallClassifier
- typeOf(aCall)
- aClassifiedCall
- combines a call and a call type
- aNumber
- country code
- area code
- local number
- problem: sometimes you can keep your number even if you move
- a cell phone number does not tell you where you are
- to do: model the customer

final step is to print invoices
- Should we model invoices as objects?
- Yes may need to correct one
- Need to model it
- Real responsibilities pertain to call ...