Handout 1

Object-Oriented technology for Software Systems Development:
Introduction and Motivation

- Object-Oriented Analysis
- Object-Oriented Design
- Object-Oriented Programming
- Object-Oriented Database

History

- object-oriented programming dates to late 1960s – early 70-s
  (SIMULA, Smalltalk)

- all through the 1980s structured analysis and functional decomposition ruled
  in the systems analysis and design sphere, OOA lagged behind

- in 1990
  OO concepts matured (although there are still some debates raging)
  OO Languages became more available
  systems became much more complex
  systems became more data-oriented
  OOA takes on a larger role

Why study it?
It is popular because of issues of productivity and overall speed of development.

Key idea – focus modeling on the individuals of the problem domain rather than system functions, build system description “around” the objects, because domain objects usually comprise the most stable part of the system (usually don’t change much even if system requirements change dramatically).

Another important aspect – package object data and behavior together.

Some Alternative Analysis Methods

Functional Decomposition
Data Flow Approach – structured analysis
Information Modeling
Problems with these approaches

- volatile with respect to changing requirements
- do not scale well
- usually contain disjoint models of data and processes

Key OO Concepts – Overview

1. Objects and Classes
   problem domain entities classified according to their properties

2. Encapsulation
   package an object’s data attributes and behavior together
   restrict access to data

3. Specialization (a.k.a. inheritance)
   detect commonalities between classes – create class hierarchy to “inherit”
   common properties without repeating them

4. Communication with messages
   to invoke an object – send it a message, let the object itself handle it

5. Polymorphism
   often same operation exists in many classes, example: Print
   each class implements it differently, but uses the same name for it

How these features help to avoid problems that exist with other approaches?