

SOFTENG 701:
Advanced Software Engineering Development Methods
Part 2

Lecture 1b: Object-Oriented Design

Ewan Tempero
Department of Computer Science

Design Quality

- Design Quality

- OOP
- OOP
- Kay, Keynote at OOPSLA 1997
- OOP
- OOP
- OOD
- Finding Good Objects
- Context
- Context
- Example
- OOD
- Key Points

- A programming language is a **tool** that can be used to “write a program” — **that is, produce “source code”**
 - Source code by itself is of no value to the client—to be of value the code must be **executed**
 - From the client’s point of view, the program’s value is determined by the degree to which it meets the requirements the client has (and hopefully conveyed to the developer)
 - \Rightarrow the client does not care about the choice of language, or even the “quality” of the program
 - **So who cares about program quality?**
- \Rightarrow developer (more generally, development team)

What is the object-oriented paradigm?

- Design Quality
- OOP
- OOP
- Kay, Keynote at OOPSLA 1997
- OOP
- OOP
- OOD
- Finding Good Objects
- Context
- Context
- Example
- OOD
- Key Points

- To evaluate the quality of anything, e.g. object-oriented design, we need:
 - a good definition of what that thing is
 - a good definition of what we mean by “quality”
 - a model that explains design quality ⇒
 - observations of good and bad quality

- Design Quality
- OOP
- **OOP**
- Kay, Keynote at OOPSLA 1997
- OOP
- OOP
- OOD
- Finding Good Objects
- Context
- Context
- Example
- OOD
- Key Points

King 1989 (Object-oriented cats)

- *It's interesting to note, however, that there is considerable disagreement concerning the definition of "object-oriented." We know it's a good thing, but not everyone agrees on what it is.*

Roger King. **1989**. "My cat is object-oriented". In *Object-oriented concepts, databases, and applications*, Won Kim and F. H. Lochovsky (Eds.). ACM, New York, NY, USA 23-30.

DOI=<http://dx.doi.org/10.1145/63320.66469>

- Design Quality
- OOP
- OOP
- Kay, Keynote at OOPSLA 1997
- OOP
- OOP
- OOD
- Finding Good Objects
- Context
- Context
- Example
- OOD
- Key Points

Kay, Keynote at OOPSLA 1997

- Kay on the design of Smalltalk:
 1. Everything is an object
 2. Objects communicate by sending and receiving messages (in terms of objects)
 3. Objects have their own memory (in terms of objects)
 4. Every object is an instance of a class (which must be an object)
 5. The class holds the shared behavior for its instances (in the form of objects in a program list)
 6. To eval a program list, control is passed to the first object and the remainder is treated as its message

Alan C. Kay. **1993**. "The early history of Smalltalk". In *The second ACM SIGPLAN conference on History of programming languages (HOPL-II)*. ACM, New York, NY, USA, 69-95.

DOI=<http://dx.doi.org/10.1145/154766.155364>

- "I invented the term object-oriented, and I can tell you that C++ wasn't what I had in mind"

Alan Kay. **1997**. "The computer revolution hasn't happened yet"

<https://www.youtube.com/watch?v=oKg1hTOQXoY>, timecode 10m 34s

Stroustrup, C++ is OO!

- Design Quality
- OOP
- OOP
- Kay, Keynote at OOPSLA 1997
- OOP
- OOP
- OOD
- Finding Good Objects
- Context
- Context
- Example
- OOD
- Key Points

- “What is Object-Oriented Programming?”
 - “If the term ‘object-oriented language’ means anything, it must mean a language that has mechanisms that support the object-oriented style of programming well”

Bjarne Stroustrup. **1988**. “What Is Object-Oriented Programming?” *IEEE Software*. 5, 3 (May 1988), 10-20.

DOI=<http://dx.doi.org/10.1109/52.2020>

- **virtual function call** (a.k.a dynamic dispatch, a.k.a **polymorphism**) — determine code executed at a call site at run-time
- **type checking** — (ideally) prevent “message not understood” run-time errors
- **inheritance** — sharing of code and polymorphism
- **multiple inheritance** — a class can have multiple parents
- **encapsulation** — protection from “unauthorised access”
- **data abstraction** — data hiding, type creation

OO Language Features?

- Design Quality
- OOP
- OOP
- Kay, Keynote at OOPSLA 1997
- **OOP**
- OOP
- OOD
- Finding Good Objects
- Context
- Context
- Example
- OOD
- Key Points

- May discussions about “OO” are mostly about the characteristics of the **language**
- Can we talk about object-oriented **design** independent of any programming language?
- What features that people talk about for OO languages must we have?

- Design Quality
- OOP
- OOP
- Kay, Keynote at OOPSLA 1997
- OOP
- OOP
- OOD
- Finding Good Objects
- Context
- Context
- Example
- OOD
- Key Points

Language features

To be considered “OOP”, a language must have:

- objects (naturally!)
- encapsulation (what is encapsulate? \Rightarrow objects)
- classes (what about Self?, Javascript? — prototype-based inheritance)
- inheritance (Emerald? *depends what you mean by inheritance*)
- types (Smalltalk?)
- generic types (Smalltalk?, Java before generics?)
- polymorphism, (a.k.a virtual function call, dynamic dispatch, message send)

“Working in an object-oriented language [...] is neither a necessary or sufficient condition for doing object-oriented programming. . . .”

Timothy Budd 2001 “An Introduction to Object-Oriented Programming (3rd Ed)” Pearson

- Design Quality
- OOP
- OOP
- Kay, Keynote at OOPSLA 1997
- OOP
- **OOP**
- OOD
- Finding Good Objects
- Context
- Context
- Example
- OOD
- Key Points

Language features

To be considered “OOP”, a language must have:

- **objects** (naturally!)
- encapsulation (what is encapsulate? \Rightarrow objects)
- classes (what about Self?, Javascript? — prototype-based inheritance)
- inheritance (Emerald? *depends what you mean by inheritance*)
- types (Smalltalk?)
- generic types (Smalltalk?, Java before generics?)
- **polymorphism**, (a.k.a virtual function call, dynamic dispatch, message send)
- **Objects and polymorphism are run-time concepts**

“Working in an object-oriented language [...] is neither a necessary or sufficient condition for doing object-oriented programming. . . .”

Timothy Budd 2001 “An Introduction to Object-Oriented Programming (3rd Ed)” Pearson

Object-oriented design

- Design Quality
- OOP
- OOP
- Kay, Keynote at OOPSLA 1997
- OOP
- OOP
- OOD
- Finding Good Objects
- Context
- Context
- Example
- OOD
- Key Points

- An *object-oriented program* is one that **when it executes** creates **objects** that **send messages** to each other.

“... the most important aspect of OOP is the creation of a universe of **largely autonomous interacting agents**.” Timothy Budd 2001 “An Introduction to Object-Oriented Programming (3rd Ed)” Pearson

- An object-oriented **design** is one that describes an object-oriented program.

Consequences

- Design Quality
- OOP
- OOP
- Kay, Keynote at OOPSLA 1997
- OOP
- OOP
- OOD
- Finding Good Objects
- Context
- Context
- Example
- OOD
- Key Points

- encapsulation is a consequence of interaction between objects only being by message sent
 - ... and so direct access to fields is “not object-oriented”
 - classes are for describing and creating objects
 - inheritance (in any form) is not relevant to what an object-oriented program — it *is* relevant to design quality however.
 - types are about preventing certain kinds of faults
 - Static members are not part of an object-oriented design
- ⇒ What is “object-oriented design quality” independent of other quality attributes (e.g maintainability, reusability, testability, etc)?

Object-oriented design quality

- Design Quality
- OOP
- OOP
- Kay, Keynote at OOPSLA 1997
- OOP
- OOP
- OOD
- Finding Good Objects
- Context
- Context
- Example
- OOD
- Key Points

- Are the objects “good”?
 - Are the messages “good”?
- ⇒ What is an “object” and what makes it good?
- ⇒ What is a “message” and what makes it good?

Objects

- Design Quality
- OOP
- OOP
- Kay, Keynote at OOPSLA 1997
- OOP
- OOP
- OOD
- Finding Good Objects
- Context
- Context
- Example
- OOD
- Key Points

- Traditionally, an “object” is described as something that has **identity**, **state**, and **behaviour**
(e.g. Grady Booch. 2004. Object-Oriented Analysis and Design with Applications (3rd Ed). Addison Wesley)
- If an object-oriented program creates messages that send messages to each other then:
 1. for one object to send a message to another, it must have some way of **identifying** that other object
 2. in order for sending a message to be useful, the consequences of the message must provide value to the sender. Consequences depend on:
 - how the receiving object responds or **behaves** upon receiving the message
 - the history or **state** of the receiving object

- Design Quality
- OOP
- OOP
- Kay, Keynote at OOPSLA 1997
- OOP
- OOP
- OOD
- Finding Good Objects
- Context
- Context
- Example
- OOD
- Key Points

Good Objects

- Is it sufficient to just use any old objects, or are some objects better than others?
- Heuristic 3.6. “Model the real world whenever possible.” Arthur J. Riel (1996) “Object-Oriented Design Heuristics” Addison-Wesley
- “Identify responsibilities [behaviour], group them into roles, define collaboration patterns between roles, and then **identify the objects that can play these roles**”

Christensen (2005) “Implications of perspective in teaching objects first and object design.” ITiCSE’05, <https://doi.org/10.1145/1151954.1067474>

Summarising Budd’s presentation of Responsibility-Driven Design (Wirfs-Brock, Wilkerson, Wiener (1990) “Designing Object-Oriented Software” Prentice-Hall)

- “This is especially true if we take an object-oriented view of the world, because objects, **as abstractions of entities in the real world**, represent a particularly dense and cohesive clustering of information.” Booch
- Look at a description of what needs to be built and identify nouns and verbs, because “. . . the nouns will often correspond to classes and objects, whereas verbs will correspond to the things those object so. . .” Barnes, Kölling (2006) “Objects First with Java” Pearson.

- Design Quality
- OOP
- OOP
- Kay, Keynote at OOPSLA 1997
- OOP
- OOP
- OOD
- Finding Good Objects
- Context
- Context
- Example
- OOD
- Key Points

Finding Good Objects

- Some nouns in the description don't correspond to things we need to build (e.g. "The details of the game (adapted from the Wikipedia **article** on Kalah) are as follows.")
- Some things we need to write code for don't appear in the real world (e.g. pull-down menus).
- Some reasonable-sounding objects appear in designs that are not part of any description of what needs to be built (e.g. HashMap)
- **But**, all advice points to choosing objects that "make sense" with respect to the problem being solved.

- Design Quality
- OOP
- OOP
- Kay, Keynote at OOPSLA 1997
- OOP
- OOP
- OOD
- Finding Good Objects
- **Context**
- Context
- Example
- OOD
- Key Points

Understanding the problem

- If choice of good objects depends on what “makes sense” in what we are trying to do, then we need to be clear about what we are trying to do.
- Many phrases get used to describe “figuring out what we are trying to do” E.g. Requirements (Elicitation), Analysis, (Problem) Domain Modelling. There is not complete agreement as to what these all mean (or even if they are in fact about figuring out what we are trying to do. . .)
- So we will use “context schema” and “design schema”

Understanding the problem

- Design Quality
- OOP
- OOP
- Kay, Keynote at OOPSLA 1997
- OOP
- OOP
- OOD
- Finding Good Objects
- Context
- **Context**
- Example
- OOD
- Key Points

- **schema** “a mental representation of some aspect of experience, based on prior experience and memory, structured in such a way as to facilitate (and sometimes to distort) perception, cognition, the drawing of inferences, or the interpretation of new information in terms of existing knowledge.” A. Colman, A Dictionary of Psychology, third ed., Oxford University Press, 2008.
- **design schema** “The collection of all of the design agents beliefs about the diversity of possible design artifacts, including design decisions and candidate solutions; the design agents mental picture of the design object and its alternatives”
- **context schema** “The collection of all of the design agents beliefs about the context; the design agents mental picture of the context.”

Based on: Paul Ralph (2015) “The Sensemaking-Coevolution-Implementation Theory of software design” Science of Computer Programming. Vol101
<https://doi.org/10.1016/j.scico.2014.11.007>

- Design Quality
- OOP
- OOP
- Kay, Keynote at OOPSLA 1997
- OOP
- OOP
- OOD
- Finding Good Objects
- Context
- **Context**
- Example
- OOD
- Key Points

Context Schema

- The system developed needs to provide value to the client
 - Typically the value is that it helps the client (or at least intended users of the system) perform some tasks
- ⇒ the tasks need to be identified
- Some like to describe a task using a “scenario” — a description of a specific instance of a meaningful activity using the system.

- Design Quality
- OOP
- OOP
- Kay, Keynote at OOPSLA 1997
- OOP
- OOP
- OOD
- Finding Good Objects
- Context
- Context
- Example
- OOD
- Key Points

Fitness Class Management (FCM)

- You have been asked to develop a sports club management system for a club that provides group fitness classes to its members (extra to other club activities).
- The club has different types of members, such as full, family (for a small fee others in the immediate family of a full member may get some club privileges), and guest. Only full members may attend fitness classes.
- A class consists of a group of sessions. The class refers to the type of activity (e.g. Zumba, KickFit, Tai Chi, or Country Dance), a session is a specific time when that class is on.
- Each class has an instructor, who is the person who runs the sessions. However occasionally a session may have a substitute instructor (e.g. if the instructor is unavailable).
- Each class costs a certain amount (e.g. Zumba is \$3 per session, KickFit is \$5 per session, Tai Chi is free), but some classes consist of a sequence of sessions (e.g. Self Defence is 3 sessions) with a fee for the full sequence.
- Members can choose to pay for each class on the day they attend (except for classes that are part of a sequence), or be invoiced for their attendance at the end of each month.
- There is a “frequent flyer” discount of 10% if they attend more than 10 sessions (of any class) per month (not for class sequences).
- Classes are held in one of several rooms the club has available. Different sessions for a class may be in different rooms, but some classes can only be held in certain rooms (e.g. Zumba needs a room with a sound system).

- Design Quality
- OOP
- OOP
- Kay, Keynote at OOPSLA 1997
- OOP
- OOP
- OOD
- Finding Good Objects
- Context
- Context
- Example
- OOD
- Key Points

FCM Concepts

- List the most relevant concepts needed to describe what the sports club management system has to do

- Design Quality
- OOP
- OOP
- Kay, Keynote at OOPSLA 1997
- OOP
- OOP
- OOD
- Finding Good Objects
- Context
- Context
- Example
- OOD
- Key Points

FCM Scenarios

- Chris, the club secretary, prints out the fitness class schedule for the Main Hall for next week.
- Chris, the club secretary adds a new session to the Zumba class, after he has checked that there is a room available at the time requested.
- Alex, the club events manager, adds a new class on Modern Dance and schedules sessions on Tuesdays at 6pm for 55 minutes in the Main Hall.
- Jesse, the Zumba instructor, records the 5 people who took the session just finished but want their fees invoiced monthly.

- Design Quality
- OOP
- OOP
- Kay, Keynote at OOPSLA 1997
- OOP
- OOP
- OOD
- Finding Good Objects
- Context
- Context
- Example
- OOD
- Key Points

FCM Design first steps

- Based on the scenarios, which of the previously identified concepts may **not** need corresponding objects?

Object-Oriented Design

- Design Quality
- OOP
- OOP
- Kay, Keynote at OOPSLA 1997
- OOP
- OOP
- OOD
- Finding Good Objects
- Context
- Context
- Example
- **OOD**
- Key Points

- Some of the objects will correspond to concepts identified in the context schema
- Some of the objects will correspond to concepts needed to provide what the system is responsible for
- Not all concepts will need to be represented as objects

- Design Quality
- OOP
- OOP
- Kay, Keynote at OOPSLA 1997
- OOP
- OOP
- OOD
- Finding Good Objects
- Context
- Context
- Example
- OOD
- Key Points

Key Points

- Object-oriented design is creating a program that, when it executes, results in objects sending messages to each other.
- A good object-oriented design has objects that make sense with respect to the context schema.
- It is not the case that first the context schema is created and then the design schema is created — usually they will be developed together (*coevolution*)