Classes and Objects

Object Orientated Analysis and Design

Benjamin Kenwright

Outline

- Review Previous Weeks
  - Object Model, Complexity, ..
- What do we mean by Classes and Objects?
- Summary/Discussion

Review

- Last Week Object Model and Evolution of Software Engineering
- Read Chapter 1, 2 and 3
- Reviewing material regularly
  - e.g., complexity, object orientated concepts, ..

Revision Question

- Write down the four elements that Inherent Complexity derives from?

Answer

- Inherent Complexity derives from four elements:

1. complexity of the problem domain
2. difficulty of managing the development process
3. flexibility possible through software
4. problems of characterizing the behavior of discrete systems

Revision Question

- What are the three important parts of Object-Oriented Programming (OOP)?

a) uses objects; each object is an instance of some class; classes may be related to one another via inheritance

b) use modules; hierarchical structure; structures must be related to one another via inheritance

c) hierarchical structure; collection of objects; objects must be related to one another via polymorphism

Answer

- a) uses objects; each object is an instance of some class; classes may be related to one another via inheritance

Revision Question

- Why is modularity important?

a) Enables us to partitioning a program into individual components can reduce its complexity
b) Enables us to develop more optimised algorithms
c) Modularity causes issues with boundaries (or interfaces) within the program

Answer

- a) Enables us to partitioning a program into individual components can reduce its complexity

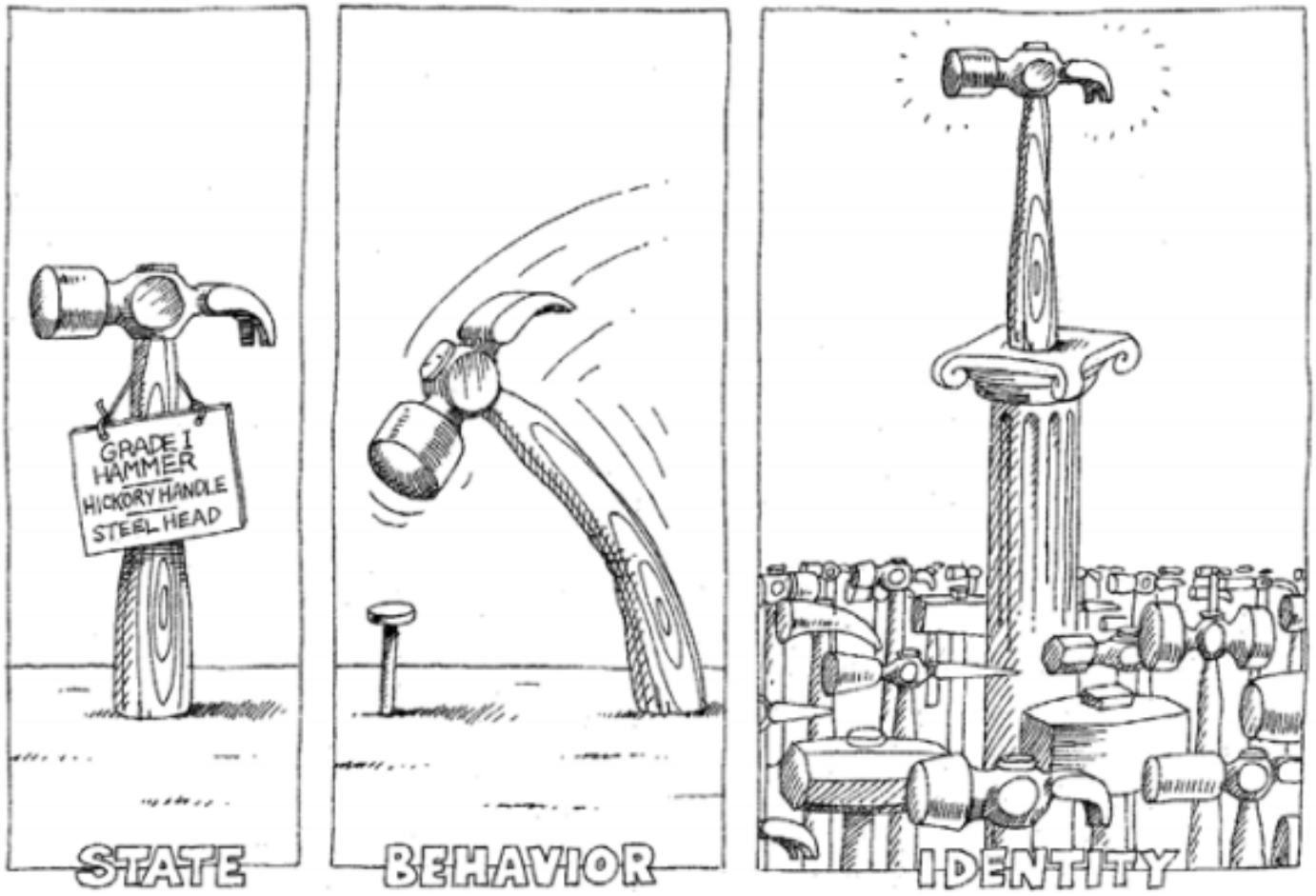
Building Blocks

- When we use object-oriented methods to analyze or design a complex software system, our basic building blocks are classes and objects

What Is and What Isn’t an Object?

What Is and What Isn't an Object?

- An object is an entity that has state, behavior, and identity. The structure and behavior of similar objects are defined in their common class. The terms instance and object are interchangeable



An object has state, exhibits some well-defined behavior,
and has a unique identity.

Question

- An object is an entity that has:

a) state, action, dependencies
b) identity, state, behavior
c) behaviour, action, state
d) class, state, memory

Answer

- b) identity, state, behavior

State

- The state of an object encompasses all of the (usually static) properties of the object plus the current (usually dynamic) values of each of these properties.

Behavior

- Behavior is how an object acts and reacts, in terms of its state changes and message passing.

Identity

- Identity is that property of an object which distinguishes it from all other objects

Question

- State is how an object acts and reacts, in terms of its state changes and message passing

a) True
b) False

Answer

- b) False

Behavior is how an object acts and reacts, in terms of its state changes and message passing.

What Is and What Isn’t a Class?

What Is and What Isn't a Class?

- A class is a set of objects that share a common structure, common behavior, and common semantics
- A single object is simply an instance of a class
- What isn't a class?
  - An object is not a class. Objects that share no common structure and behavior cannot be grouped in a class because, by definition, they are unrelated except by their general nature as objects

Inheritance

Inheritance

- Inheritance is an interesting concept for representing concrete relationships
  - Including generalization/specialization relationships
- A subclass may inherit the structure and behavior of its superclass

Polymorphism

- Helps develop more concise clean error free code
  - Without polymorphism, the developer ends up writing code consisting of large case or switch statements (conditional logic)
- Inheritance without polymorphism is possible, but it is certainly not very useful
- Polymorphism and late binding go hand in hand

The Role of Classes and Objects in Analysis and Design

- During analysis and the early stages of design, the developer has two primary tasks:

1. Identify the classes that form the vocabulary of the problem domain

2. Invent the structures whereby sets of objects work together to provide the

behaviors that satisfy the requirements of the problem

Measuring the Quality of an Abstraction

- How can one know if a given class or object is well designed?

Measuring the Quality of an Abstraction

- Five meaningful metrics to measure the quality of abstraction are:

1. Coupling
2. Cohesion
3. Sufficiency
4. Completeness
5. Primitivenes

Question

- What are the five meaningful metrics to measure the quality of abstraction?

a) Contracts, Cohesion, Completeness, Primitivenes, Sufficiency

b) Cohesion, Coupling, Sufficiency, Completeness, Primitivenes

c) Coupling , Cohesion, Safety, Completeness, Primitivenes

Answer

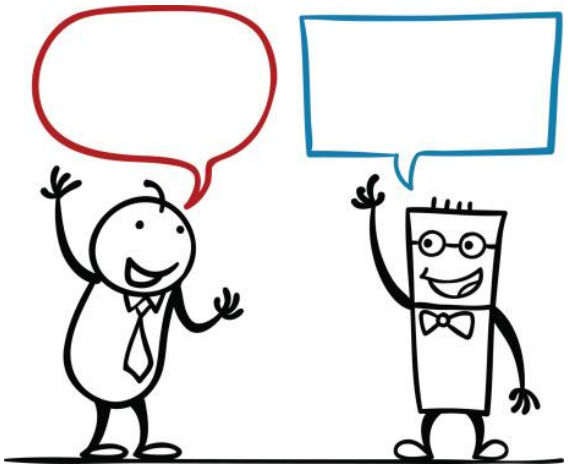
- b) Cohesion, Coupling, Sufficiency, Completeness, Primitivenes

Design Decisions

- Criteria to be considered when making class design decisions:
  - Reusability: Would this behavior be more useful in more than one context?
  - Complexity: How difficult is it to implement the behavior?
  - Applicability: How relevant is the behavior to the type in which it might be placed?
  - Implementation knowledge: Does the behavior's implementation depend on the internal details of a type?

Discussion Activity

- Explain some of the trade-offs decisions that might be made during the analysis and design of a software project?



Summary

- Clear understanding of classes and objects
- An object has state, behavior, and identity
- The structure and behavior of similar objects are defined in their common class
- Behavior is how an object acts and reacts in terms of its state changes and message passing

This Week

- Review Slides
- Quizzes Online
- Read Chapter 4

Questions/Discussion

Question

- Identity is that property of an object which distinguishes it from all other objects

a) True

b) False

Answer

- a) True

Question

- A class is the same as an object. Objects share common structure and behavior as a class because and by definition are related by their general nature as objects

a) True

b) False

Answer

- b) False

An object is not a class. Objects that share no common structure and behavior cannot be grouped in a class because, by definition, they are unrelated except by their general nature as objects