



Ken Wong

Department of Computing Science
University of Alberta



REVIEW



Jeopardy Game

- Instructions:
 - clue is stated
 - raise your hand
 - you state the *question* (in that form)
- not really final exam questions
- but an interesting, “competitive” review of software engineering concepts and terms



OOAD

- Clue:
 - An object-oriented programming language, invented by James Gosling.

- Question:
 - What is Java?



OOAD

- Clue:
 - A visual design notation, that's “unified”.

- Question:
 - What is UML?



Process

- Clue:
 - Making sure you develop the system right.

- Question:
 - What is verification?



Process

- Clue:
 - Making sure you develop the right system.

- Question:
 - What is validation?



Process

- Clue:
 - Three approaches of software prototyping.

- Question:
 - What are throwaway, incremental, evolutionary?



Process

- Clue:
 - In Extreme Programming, code should conform to these rules.

- Question:
 - What are coding conventions?



Process

- Clue:
 - A practice where production code is written with two programmers actively at one machine.

- Question:
 - What is pair programming?



OOAD

- Clue:
 - Simplifying to its essentials the description of a real-world entity or concept.

- Question:
 - What is abstraction?



OOAD

- Clue:
 - Bundling data with access functions, in a way that distinguishes “what” from “how”.

- Question:
 - What is encapsulation?



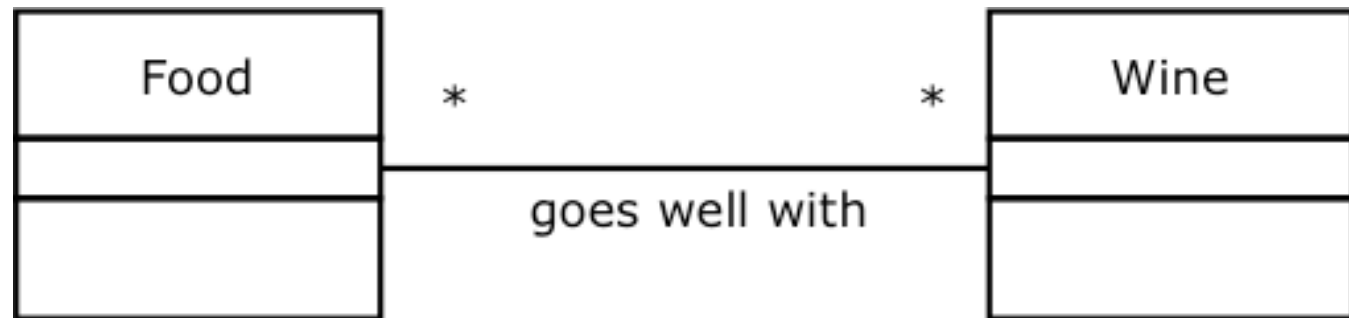
OOAD

- Clue:
 - Revealing assumptions through interfaces and hiding changeable internal details.

- Question:
 - What is information hiding?

OOAD

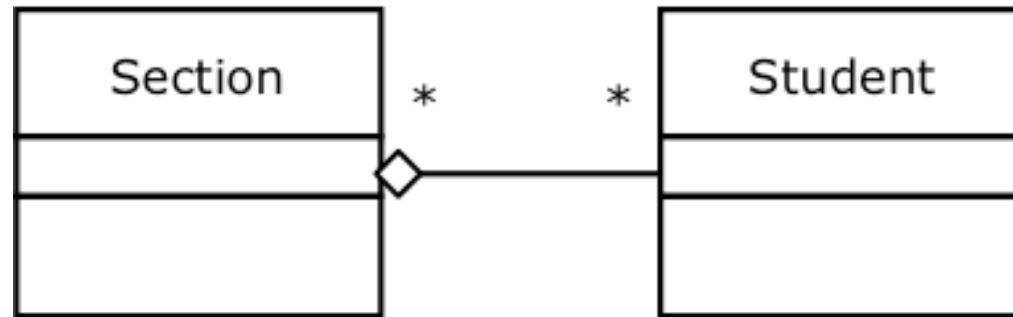
- Clue:
 - “Some” relationship between parts.



- Question:
 - What is an association?

OOAD

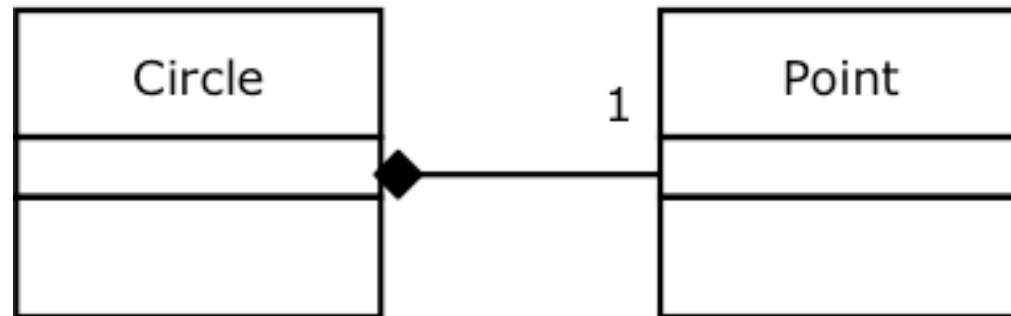
- Clue:
 - A weak “has-a” relationship.



- Question:
 - What is an aggregation?

OOAD

- Clue:
 - Contained instances are exclusive to the container in this kind of UML relationship.



- Question:
 - What is a composition?



OOAD

- Clue:
 - Looking for conceptual commonalities in abstractions.

- Question:
 - What is generalization?



OOAD

- Clue:
 - In Java, this can be considered a “contract”, specifying a capability that implementing classes must provide.

- Question:
 - What is an interface?



OOAD

- Clue:
 - If this test fails, inheritance is likely not appropriate.

- Question:
 - What is the is-a test?



OOAD

- Clue:
 - A candidate subclass should be substitutable anywhere a reference to a superclass object is used, according to this principle.

- Question:
 - What is the Liskov substitution principle?



OOAD

- Clue:
 - This kind of class cannot be instantiated.

- Question:
 - What is an abstract class?



OOAD

- Clue:
 - The method to run is selected at run time, depending on the type of the receiving object.

- Question:
 - What is dynamic binding?



OOAD

- Clue:
 - This widening type of cast is safe due to the principle of substitutability.

- Question:
 - What is an upcast?



OOAD

- Clue:
 - Using index cards to assist object-oriented analysis.

- Question:
 - What is CRC design?



OOAD

- Clue:
 - One should reduce this between classes.

- Question:
 - What is coupling?



OOAD

- Clue:
 - Time flows downward in this UML diagram to express behavior between objects.

- Question:
 - What is a UML sequence diagram?



OOAD

- Clue:
 - Each object in a UML sequence diagram plays this in a group of collaborating objects.

- Question:
 - What is a role?



Requirements

- Clue:
 - They may not know what is possible, or be able to express their needs.

- Question:
 - Who are users?



Requirements

- Clue:
 - Required qualities, such as those -ibilities.

- Question:
 - What are non-functional requirements?



Requirements

- Clue:
 - Requirements should be this, so tests can be designed to show the system fulfills them.

- Question:
 - What is verifiable?



Requirements

- Clue:
 - A tendency for developers to focus on an increasingly expert group of customers, and excluding a potential market.

- Question:
 - What is the innovator's dilemma?



Requirements

- Clue:
 - This captures the goal, conditions, and steps of a coherent interaction between the users and the system.

- Question:
 - What is a use case?

Requirements

- Clue:
 - Different types of users or roles in use cases.



- Question:
 - What are actors?



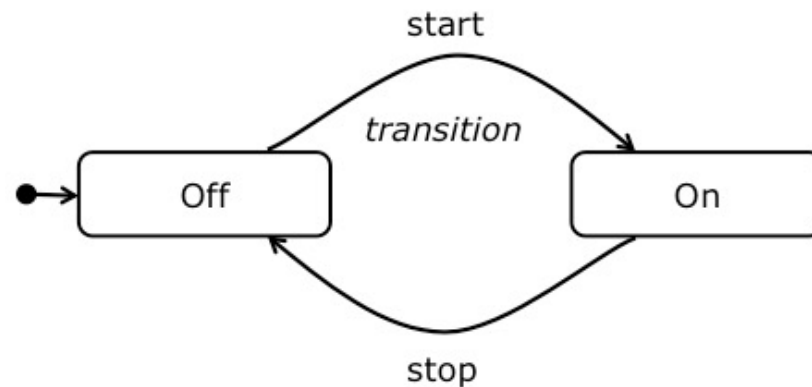
Requirements

- Clue:
 - A way to specify a need often written in the form: as a «user role», I want «goal».

- Question:
 - What is a user story?

Requirements

- Clue:
 - A UML diagram used to model the behavior of an object in response to external events.



- Question:
 - What is a UML state diagram?



Testing

- Clue:
 - This leads to faults in work products, and may cause failures in running software.

- Question:
 - What is human error?



Testing

- Clue:
 - This kind of testing is to prevent previous problems from reoccurring.

- Question:
 - What is regression testing?



Testing

- Clue:
 - The correct way to test a theory is to seek this.

- Question:
 - What is to refute it?



Testing

- Clue:
 - Use this technique to separate out dependency resolution from the constituent classes and enhance testability.

- Question:
 - What is dependency injection?



Testing

- Clue:
 - A kind of testing object that mimics the real object but can be further instrumented.

- Question:
 - What is a mock object?



Testing

- Clue:
 - A way of development where tests are generally written before the code.


- Question:
 - What is test-driven development?



Testing

- Clue:
 - A commonly used Java framework for writing unit tests.

- Question:
 - What is JUnit?

- 
- Clue:
 - A practical, proven solution to a recurring design problem.

 - Question:
 - What is a design pattern?



Design Patterns

- Clue:
 - This design pattern ensures a class only has one instance, and provides a global point of access to it.

- Question:
 - What is the singleton pattern?



Design Patterns

- Clue:
 - This design pattern composes individual objects to form a tree structure, and treats individual and composed objects uniformly.

- Question:
 - What is the composite pattern?



Design Patterns

- Clue:
 - This design pattern encapsulates a request as an object, so you can later undo/redo the request.

- Question:
 - What is the command pattern?



Design Patterns

- Clue:
 - This design pattern defines the skeleton of an algorithm, deferring some steps to subclasses.

- Question:
 - What is the template method pattern?



Design Patterns

- Clue:
 - An object whose main responsibility is to make other objects.

- Question:
 - What is a factory object?



Design Patterns

- Clue:
 - This design pattern defines an interface for creating an object, but lets subclasses decide which class to instantiate.

- Question:
 - What is the factory method pattern?



Design Patterns

- Clue:
 - This design pattern allows an object to alter its behavior when its internal state changes.

- Question:
 - What is the state pattern?



Design Patterns

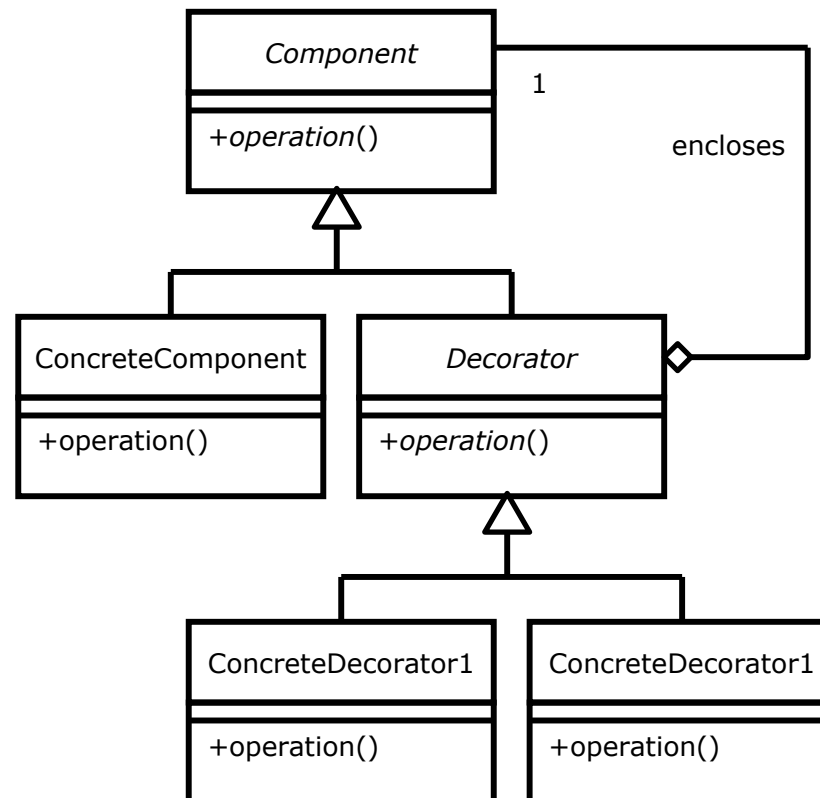
- Clue:
 - This design pattern adapts the interface of a class into another interface that clients expect.

- Question:
 - What is the adapter pattern?

Design Patterns

- Clue:

-



- Question:

- What is the decorator pattern?



Design Patterns

- Clue:
 - In this design principle, classes should be open for extension but closed for modification.

- Question:
 - What is the open-closed principle?



Design Patterns

- Clue:
 - In this design principle, depend on abstractions or generalizations, not on concrete classes.

- Question:
 - What is the dependency inversion principle?



Design Patterns

- Clue:
 - In this design principle, for a class, reduce the number of classes it knows about and interacts with.


- Question:
 - What is the principle of least knowledge?



Design Patterns

- Clue:
 - This law suggests the only methods that may be called, to conform with the principle of least knowledge.

- Question:
 - What is the Law of Demeter?

- 
- Clue:
 - Change a software system so that the external behavior does not change but the internal structure is improved.

 - Question:
 - What is refactoring?



Refactoring

- Clue:
 - Risk is reduced in refactoring by proceeding in small steps and doing this after each step.

- Question:
 - What is testing?



Refactoring

- Clue:
 - Indications that the code may need refactoring.

- Question:
 - What are code smells?



Refactoring

- Clue:
 - Code with very complex, tangled control flow typified by lots of gotos.

- Question:
 - What is spaghetti code?



Refactoring

- Clue:
 - A class that gets increasingly larger, which may indicate poor separate of concerns.

- Question:
 - What is a blob class?



Refactoring

- Clue:
 - When a class is commonly changed in different ways for different reasons.

- Question:
 - What is “divergent change”?



Refactoring

- Clue:
 - When making a change requires many little changes across many different classes or methods.

- Question:
 - What is “shotgun surgery”?



Refactoring

- Clue:
 - When a method seems more interested in the details of a class other than the one it is in.

- Question:
 - What is “feature envy”?



Refactoring

- Clue:
 - When using the built-in types too much rather than classes to represent concepts in the problem domain.

- Question:
 - What is “primitive obsession”?



Refactoring

- Clue:
 - When code is created because “we might need it someday”, which adds design complexity.

- Question:
 - What is “speculative generality”?



Refactoring

- Clue:
 - When a subclass inherits something that is not needed.

- Question:
 - What is “refused bequest”?



Refactoring

- Clue:
 - Potentially deodorant for bad smelling code.

- Question:
 - What are comments?