

OOAD

"There is no silver bullet."

—Fred P. Brooks, Jr.

OO Design is an Art



30-Oct-15 8:07 PM

By Vijay – caretainings.co.in

2

OOAD books and classes can only enhance the innate talents of individuals and make them the best OO engineers they can be.

The education of artists is not focused on technique, process or tools.

The majority of an art education combines ideas, history, appreciation, experience and constructive criticism.

Different

- Advocacy of a local rather than global focus
- Practitioners of rapid prototyping instead of structured development



30-Oct-15 8:07 PM



By Vijay – caretainings.co.in

3

Collaborative rather than imperial management style

OOD -

Commitment to design based on coordination and cooperation rather than control

Driven by internal capabilities instead of conforming to external procedures.

Decide Classes

- How to decide what should be a class?
 - Nouns
 - Value is a group of items.
 - Functions associated with an item.



30-Oct-15 8:07 PM

By Vijay – caretrainings.co.in

4

Q022: tellLocalNumber

Q01: IsSameString

Class should be highly cohesive

A Class should have a single well focused purpose.

A Class should do one thing and do it well.

Find the classes

- Selling soft drinks on a vending machine.
 - Software will control the functions of the vending machine.
 - First the user enters some money. The machine displays the money entered so far. The products that can be bought light up. The user chooses his option. The vending machine dispenses the product and the change.

30-Oct-15 8:07 PM

By Vijay – caretainings.co.in

5

Answer: VendingMachine, MoneyBox, Screen, PriceList, SoftDrink, SoftDrinkList, SoftDrinkDispenser, Safe

Metaphor

- Object is like a person.
 - Both are specialists and lazy
 - Both don't like to be micromanaged.
 - Both take responsibilities

30-Oct-15 8:07 PM

By Vijay – caretainings.co.in

6

Distributed cooperation and communication must replace hierarchical centralized control as an organizational paradigm. E.g. A traffic signal and cars.

Like people, software objects are specialists and lazy. A consequence of both these facts is the distribution of work across a group of objects. Take the job of adding a sentence to a page in a book. Granted, it might be quite proper to ask the book, "Please replace the sentence on page 58 with the following." (The book object is kind of a spokesperson for all the objects that make up the book.) It would be quite improper, however, to expect the book itself to do the work assigned. If the book were to do that kind of work, it would have to know everything relevant about each page and page type that it might contain and how making a simple change might alter the appearance and the abilities of the page object. Plus the page might be offended if the book attempted to meddle with its internals.

The task is too hard (lazy object) and not the book's job (specialist object), so it delegates—merely passes to the page object named #58 the requested change. It's the page object's responsibility to carry out the task. And it too might delegate any part of it that is hard—to a string object perhaps.

Objects, like the people we metaphorically equate them to, can work independently and concurrently on large-scale tasks, requiring only general coordination. When we ask an object collective to perform a task, it's important that we avoid micromanagement by imposing explicit control structures on those objects. You don't like to work for a boss who doesn't trust you and allow you to do your job, so why should your software objects put up with similar abuse?

Metaphors

- Software is a Theater, Programmer is a director
- Ants, not Autocrats



30-Oct-15 8:07 PM

By Vijay – caretrainings.co.in

7

Q57srp – Students

Q58srp – Servers

Hierarchical and centralized control is the anathema in OO. Complex systems are characterized by simple elements, acting on local knowledge with local rules, give rise to complicated patterned behavior. Object can inherit like a person.

For example, suppose an airplane object has a responsibility to report its location. This is a hard task because the location is constantly changing; a location is a composite structure (latitude, longitude, altitude, direction, speed, and vector); the values of each part of that structure come from a different source; and someone has to remember who asked for the location and make sure it gets back to them in a timely fashion. If the task is broken up so that

- The airplane actually returns a location object to whoever asked for it after appending its ID to the location so that there is no confusion about who is where. (We cannot assume that our airplane is the only one reporting its location at any one time.)
- An instrument cluster keeps track of the instruments that must be asked for their current values and knows how to ask each one in turn for its value (a collection iterating across its contents).

- An instrument merely reports its current value.
 - A location object collects and returns a set of label:value pairs (altitude:15,000 ft.).
- None of the objects do anything particularly difficult, and yet collectively they solve a complicated problem that would be very hard for any one of them to accomplish individually.

Guidelines

- DRY



30-Oct-15 8:07 PM

By Vijay – caretainings.co.in

8

Q021: BookRentals; Q11: BookRental; Q33 – SurveyData; Q37 – FOC, TT

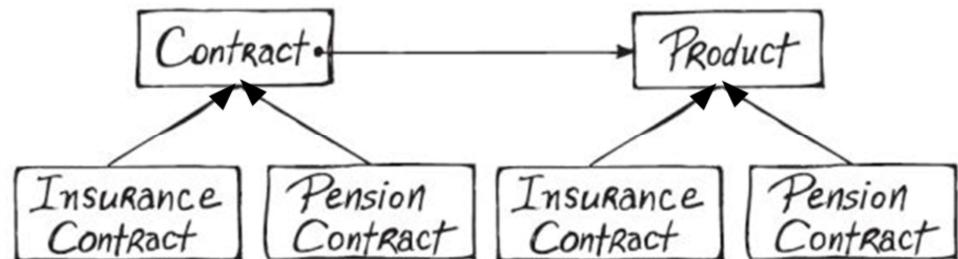
Q10 – JButton – Only for Java.

Q34 – replace; Q35 – BookRental

Q70 – Array Scan, Java only.

There are three numbers in software: 0, 1, and infinity. 0 represents the things we do not do in a system (we do those for free). 1 represents the things we do once and only once. But at the moment we do something twice, we should treat it as infinitely many and create cohesive services that allow it to be reused.

Improve

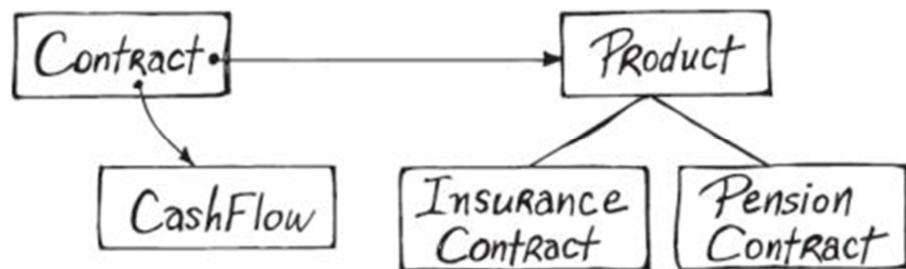


30-Oct-15 8:07 PM

By Vijay – caretrainings.co.in

9

Eliminate Parallel Hierarchies



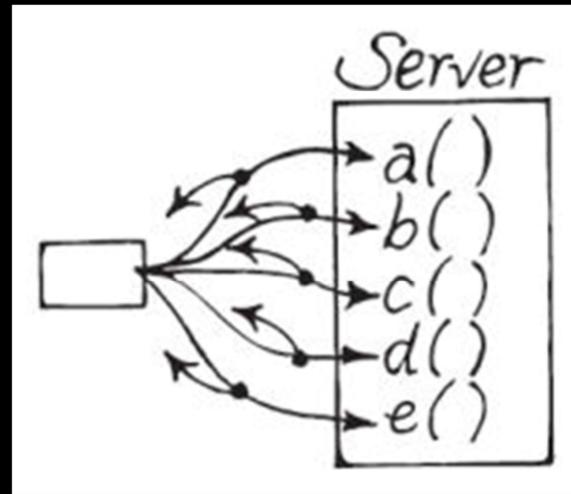
30-Oct-15 8:07 PM

By Vijay – caretrainings.co.in

10

Reduce Coupling - Problem

```
Server s = new Server();  
s.a(this);  
s.b(this);  
s.c(this);  
s.d(this);  
s.e(this);
```



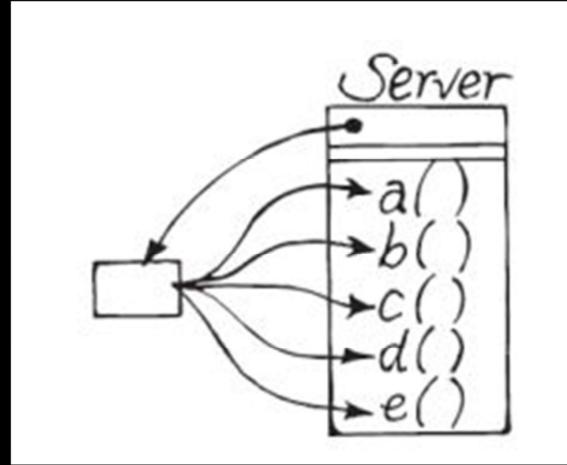
30-Oct-15 8:07 PM

By Vijay – caretainings.co.in

11

Reduce Coupling - Solution

```
Server s = new Server(this);  
s.a();  
s.b();  
s.c();  
s.d();  
s.e();
```



30-Oct-15 8:07 PM

By Vijay – caretainings.co.in

12

Better still: use Observer design pattern

Classes in an Application

- Many simple classes means that each class
 - encapsulates less of overall system intelligence
 - is more reusable
 - is easier to implement
- A few complex classes means that each class
 - encapsulates a large portion of system intelligence
 - is less likely to be reusable
 - is more difficult to implement

30-Oct-15 8:07 PM

By Vijay – caretrainings.co.in

13

Lots of little pieces

Classes are cohesive

Methods do only one thing.

Guidelines

- A class should have less than 50 lines
- Most functions should be less than or equal to 5 lines.
 - A function taking more than 3 arguments should be rare and justified specially.

30-Oct-15 8:07 PM

By Vijay – caretainings.co.in

14

Note: On a Home PC - 1 Million function calls take 8 milliseconds and 1 Million objects are created in 23 milliseconds

Some Real Examples

Tool	Files	Lines/file (avg)	LOC/file (avg)
JUnit	88	71	39
Hibernate	1063	90	72
Eclipse	14,620	153	106
DomainObjects for .NET	422	164	98
Compiere ERP &CRM	1191	163	114
Hsqldb	290	503	198

30-Oct-15 8:07 PM

By Vijay – caretrainings.co.in

15



Q56srp – Restaurants

Q59srp - Customers

Single Responsibility Principle

- A class should have only one reason to change.

- Bad

```
public class Employee {  
    public double calculatePay();  
    public double calculateTaxes();  
    public void writeToDisk();  
    public void readFromDisk();  
    public String createXML();  
    public void parseXML(String xml);  
    public void displayOnEmployeeReport(  
        PrintStream stream);  
    public void displayOnPayrollReport(  
        PrintStream stream);  
    public void displayOnTaxReport(  
        PrintStream stream);  
}
```

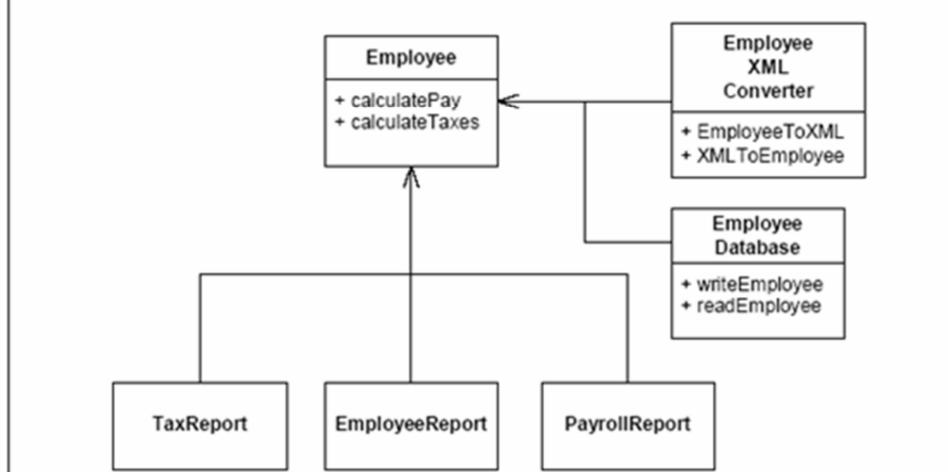
30-Oct-15 8:07 PM

By Vijay – caretrainings.co.in

16

An item such as a class should just have one responsibility and solve that responsibility well. If a class is responsible both for presentation and data access, that's a good example of a class breaking SRP.

SRP implemented



30-Oct-15 8:07 PM

By Vijay – caretrainings.co.in

17

Java API

- `java.xml.datatype.XMLGregorianCalendar` has both date and time
- `java.util.concurrent.TimeUnit` is an enum for various units and also converts from one form to another.



30-Oct-15 8:07 PM

By Vijay – caretrainings.co.in

18

Q50 – Account

Q51 – Department

Q55 – processReport1

Functional Programming

- Why are functional languages like Clojure, Scala and F# getting popular?
 - Languages like Java can get some functional features by proper design, using libraries like “Functional Java” and “Akka”.

30-Oct-15 8:07 PM

By Vijay – caretainings.co.in

19

Functional language advantages:

functions are first class citizens

functions are side-effect free

declarative. Uses recursion to reduce the size of the problem.

Pure Functions

- A function that computes output based on parameters passed only.
 - No access to instance variables or global variables
- The function does not alter any input parameters.



30-Oct-15 8:07 PM

By Vijay – caretrainings.co.in

20

Also known as side-effect free functions.

Q60.

Q61

Pipe Simple Commands

- Imagine a CSV file book.csv

```
ISBN,Title,Price  
123,Java,500  
234,C#,700  
345,OOAD,600
```

- To get sum of all prices

```
sed 1d book.csv | cut -d, -f 3 | awk '{s+=$1} END {print s}'
```

- To get sum of 3 costliest books

```
sed 1d book.csv | cut -d, -f 3 | sort -rn | head -n 3 |  
awk '{s+=$1} END {print s}'
```



30-Oct-15 8:07 PM

By Vijay – caretainings.co.in

21

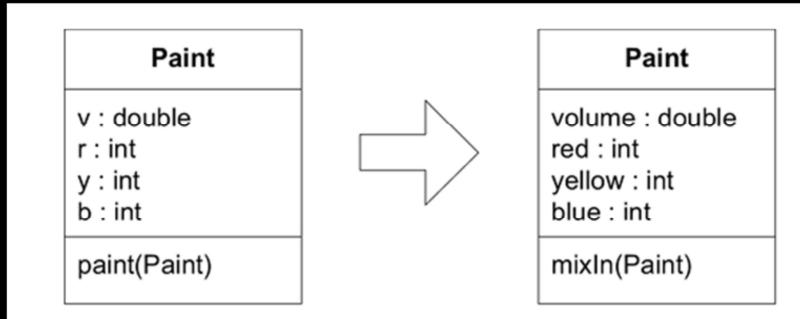
This is Linux philosophy. Do a complex process by performing simple tasks and then piping them.

Instead of building a huge complex class, build simple classes. Pipe output of one class to another. This will result in more flexible coding. More portions will be reusable.

Q62

Rule: Don't abbreviate

- Code should be self documenting.
 - Tools: Eclipse, Codepro, PMD, PMD-CPD, Findbugs, Checkstyle, Cobertura.



30-Oct-15 8:07 PM

By Vijay – caretrainings.co.in

22

Public API's have to be documented i.e. every class, function, interface, exceptions. Mutable objects that can / cannot be modified.

This is possible by choosing the right variable and function names.

Comments are secondary because they tend to lie

C#: FxCop, Simian, Ncover, NDepend for cyclomatic complexity.

C++: Simian or PMD CPD for duplication, coverity for source code analysis

```
public List<int[]> getThem() {  
    List<int[]> list1 =  
        new ArrayList<int[]>();  
    for (int[] x : theList)  
        if (x[0] == 4)  
            list1.add(x);  
    return list1;  
}  
public List<Cell> getFlaggedCells() {  
    List<Cell> flaggedCells =  
        new ArrayList<Cell>();  
    for (Cell cell : gameBoard)  
        if (cell.isFlagged())  
            flaggedCells.add(cell);  
    return flaggedCells;  
}
```

30-Oct-15 8:57:11

By Vijay caretrainings.co.in

23

```
class DtaRcrd102 {  
    private Date genymdhms;  
    private Date modymdhms;  
    private final String pszqint = "102";  
    /* ... */  
};  
  
class Customer {  
    private Date generationTimestamp;  
    private Date modificationTimestamp;;  
    private static final String  
        RECORD_ID = "102";  
    /* ... */  
};
```

30-Oct-15 8:07 PM

By Vijay – caretrainings.co.in

24

Guidelines

- Classes and objects should have noun or noun phrase names like Customer, WikiPage, Account, and AddressParser.
 - Avoid words like Manager, Processor, Data, or Info in the name of a class.
 - A class name should not be a verb.
- Methods should have verb or verb phrase names like postPayment, deletePage, or save.