Java & JEE Training

Day 7 – Object Oriented Analysis and Design with Java

MindsMapped Consulting

Agenda

- ✓ Object Oriented Concepts
- ✓ Introduction to OO Analysis and Design

Java & JEE Training

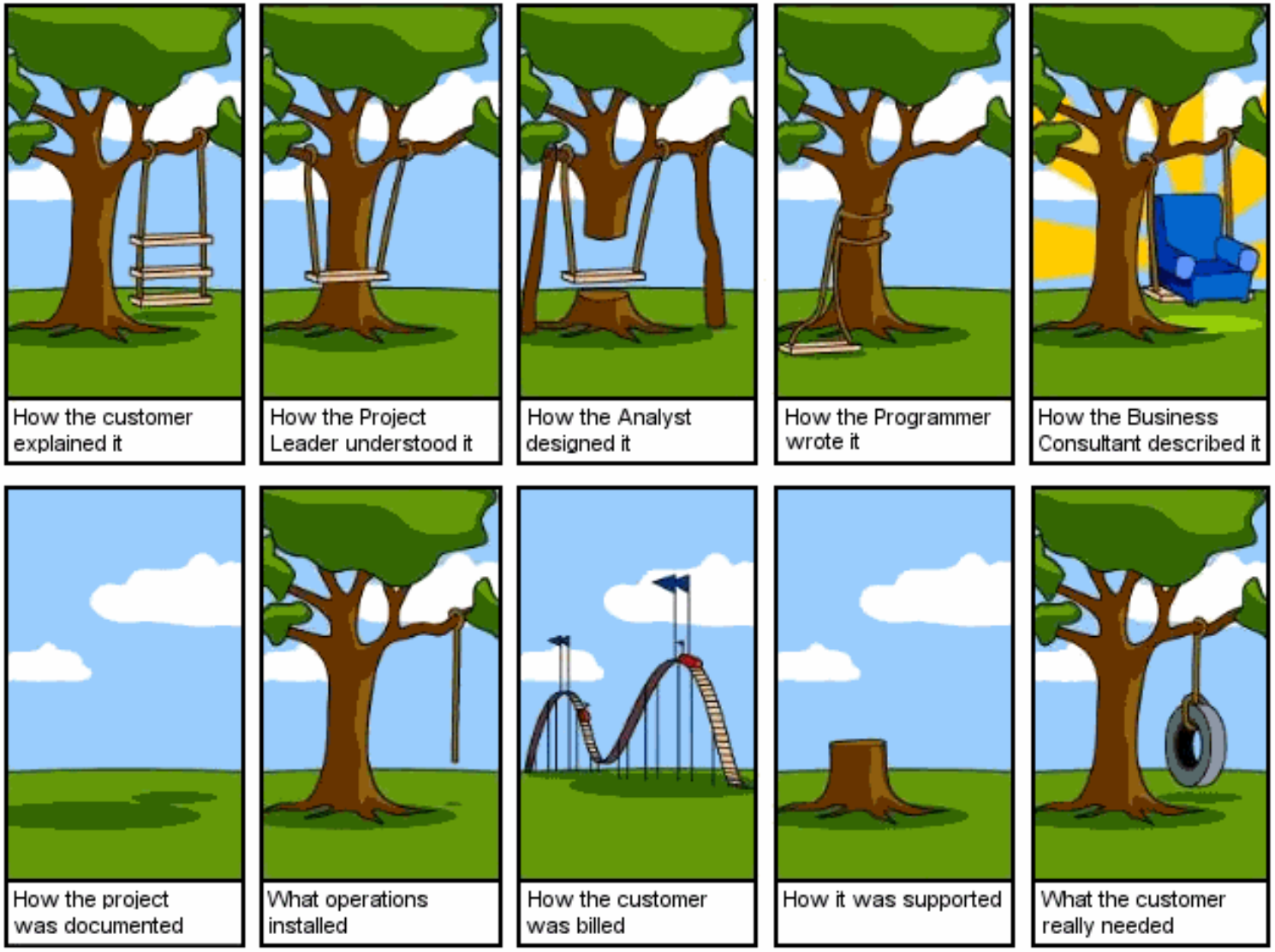
Object Oriented Programming

MindsMapped Consulting

Object Oriented Programming

- Why OO programming?
- What is OOAD? Why?

Why Object Oriented?

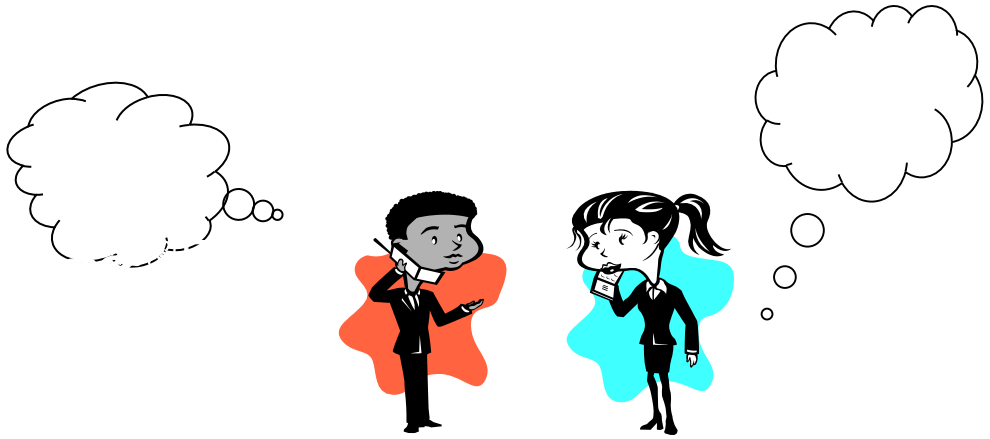


Why Object-Oriented?

*"The "software crises" came about when people realized the major problems in software development were … caused by **communication** difficulties and the management of **complexity"** [Budd]*

The Whorfian Hypothesis:

Human beings … are very much at the mercy of the particular language which has become the medium of expression for their society … the 'real world' is … built upon the language habits …

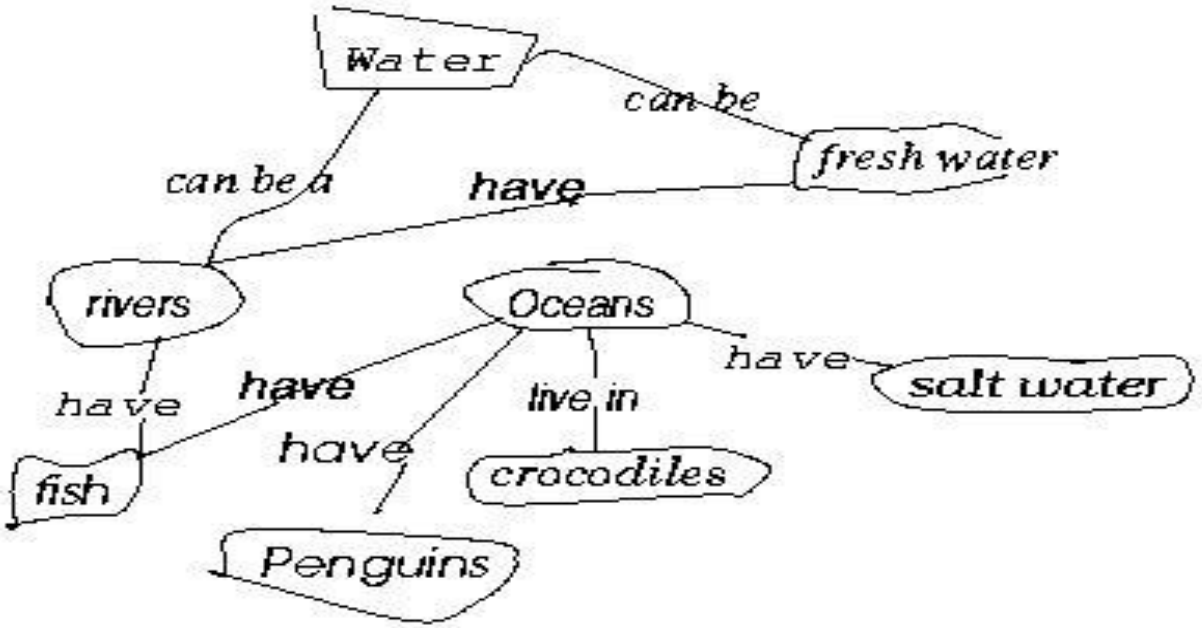


What kind of language can alleviate difficulties with communication & complexity hopefully well?

Why Object-Oriented?

For conceptual.. Modelling reasons…

What kind of language can be used to create this concept diagram, or Harry's mental image?

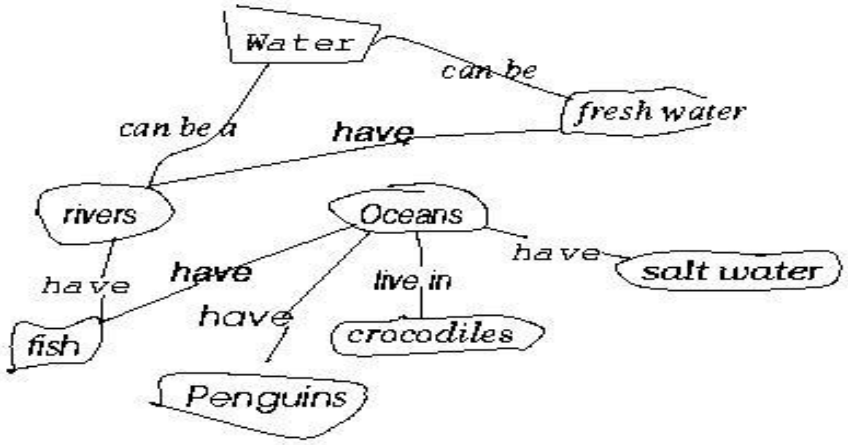


Harry's mental image

Why Object-Oriented?

For conceptual.. Modelling reasons…

What kind of language can be used to create this concept diagram, or Harry's mental image?



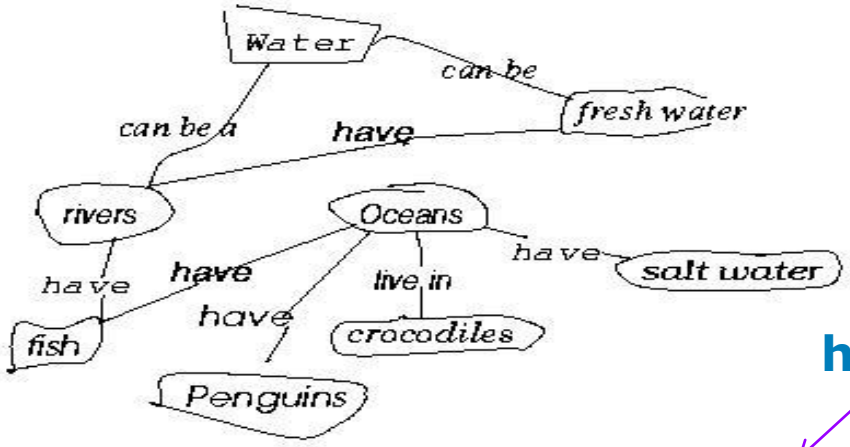
Harry's mental image

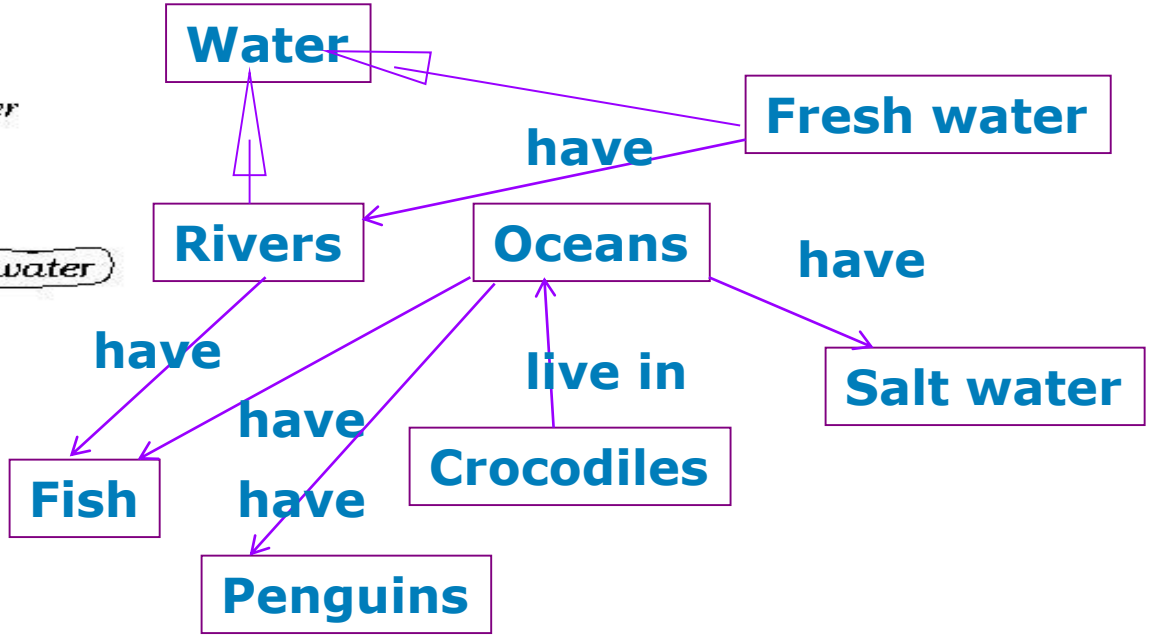
Model with Procedural language

Why Object-Oriented?

For conceptual.. Modelling reasons…

What kind of language can be used to create this concept diagram, or Harry's mental image?





Harry's mental image

Model with Object Oriented

Why Object-Oriented?

Why Model?

- To understand *why* a software system is needed, *what* it should do, and *how* it should do it.
- To communicate our understanding of why, what and how.
- To detect commonalities and differences in your perception, my perception, his perception and her perception of reality.
- To detect misunderstandings and miscommunications.

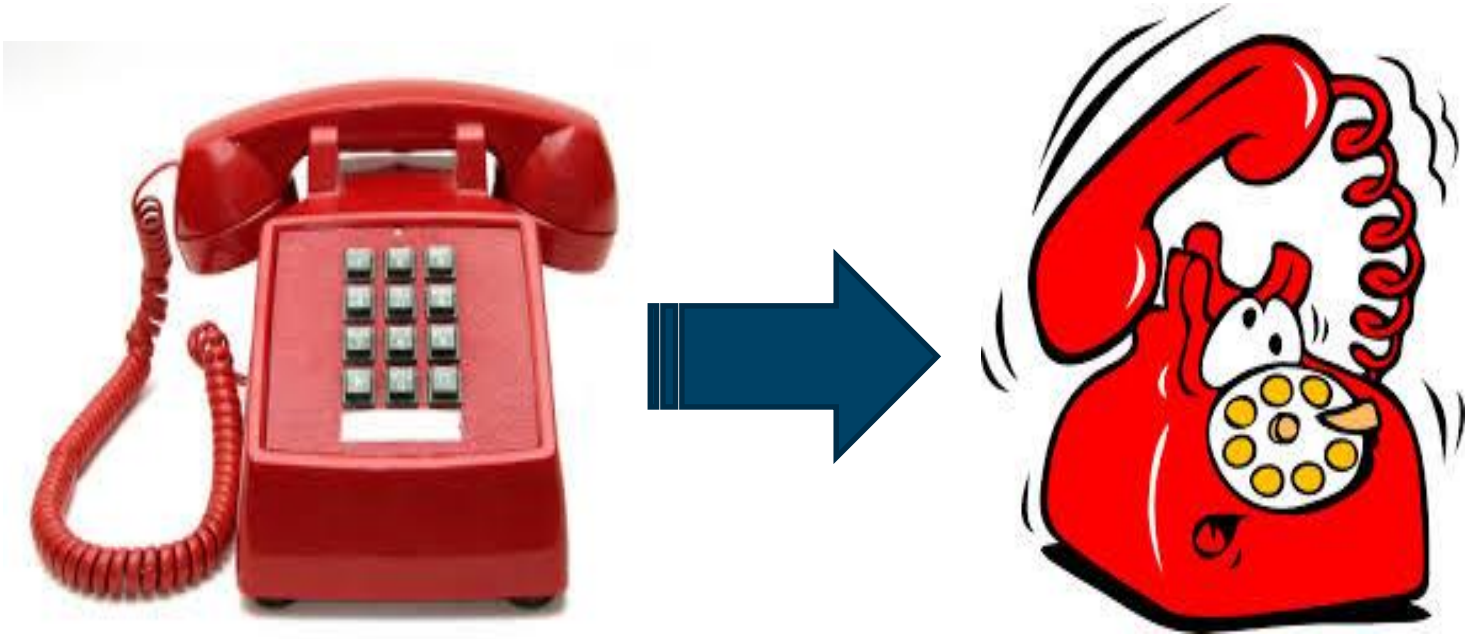
Object Oriented Programming

- Consists of classes and object.
- Object communicates with each other by passing messages

What is an object????

Object States

An Object has State and Behavior



Objects

- Objects have state and behavior

  State: What an object knows about itself

  Behavior :What an object can do.

Object Name
State = charecterstics
Behaviour:

Dog
State: Name Breed Height Weight
Behaviour: ***eat()*** ***run()*** ***walk()***

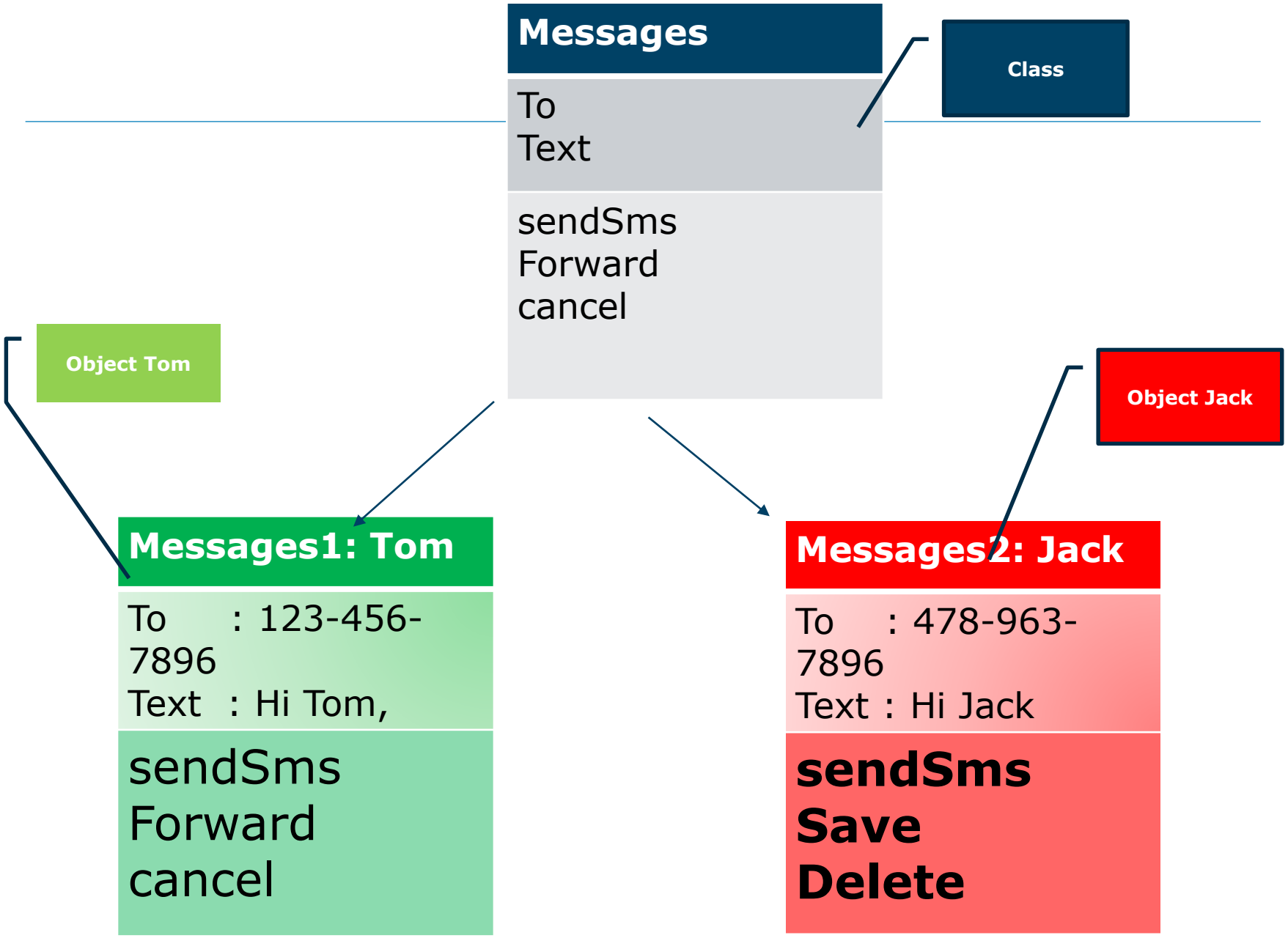
Class

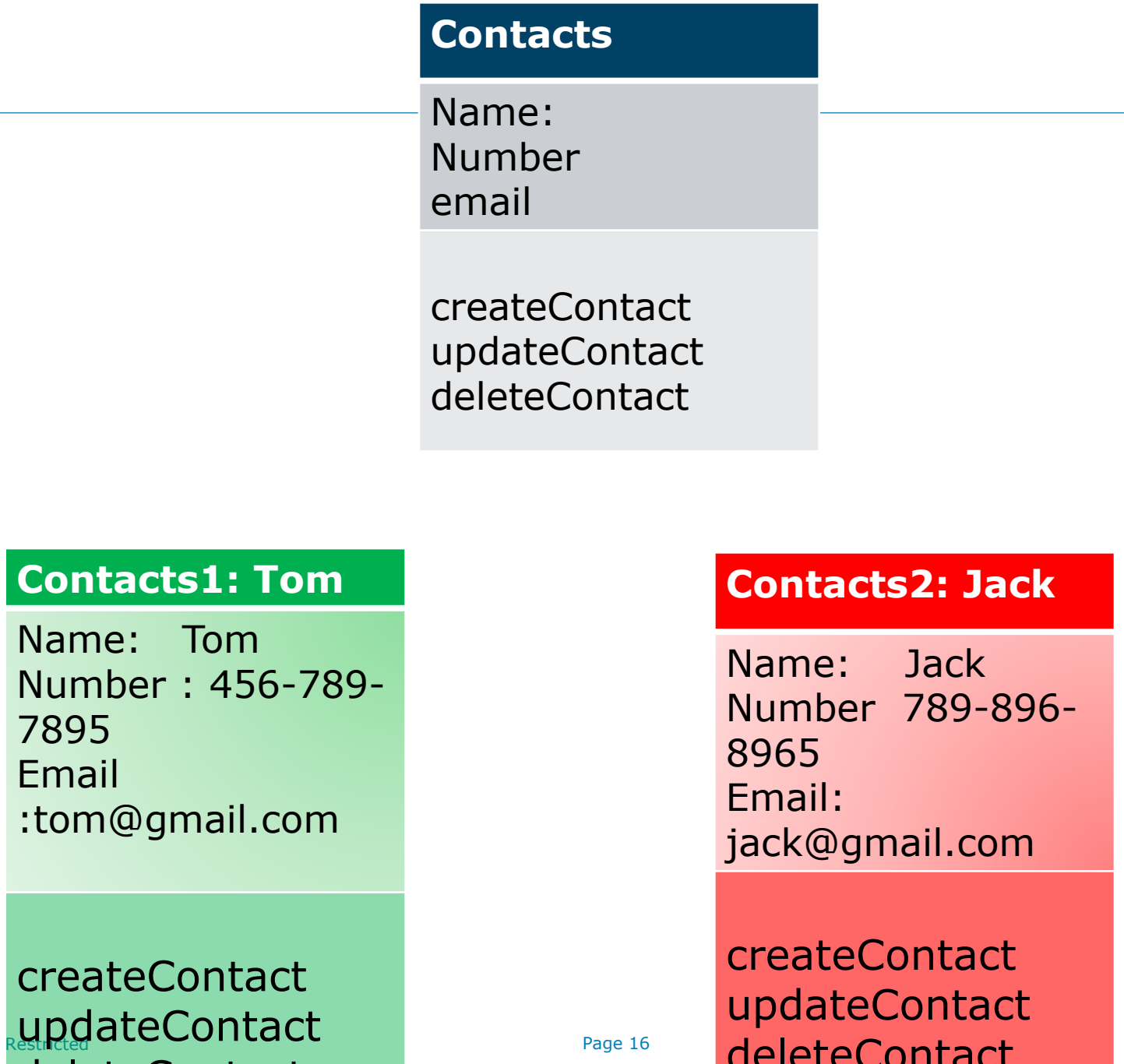
- Class is Blue Print
  - Logical structure
  - Set of instructions given to JVM , how to create instance (object) out of it.

Class

- Class consists of
  - Member variables and member methods.
  - State/ characteristics is represented via member variables
  - Member methods defines the responsibility of the class
  - Data within object represents its state.

    - State - Member Variables
      - To:
      - Text:
    - Behavior – Member functions
      - sendSms
      - Forward
      - delete


Messages
To
Text
sendSms
Forward
cancel
Class
Object Tom
Object Jack
Messages1: Tom
To : 123-456-7896
Text : Hi Tom,
sendSms
Forward
cancel
Messages2: Jack
To : 478-963-7896
Text : Hi Jack
sendSms
Save
Delete


Contacts
Name:
Number
email
createContact
updateContact
deleteContact
Contacts1: Tom
Name: Tom
Number : 456-789-7895
Email :tom@gmail.com
createContact
updateContact
Contacts2: Jack
Name: Jack
Number 789-896-8965
Email: jack@gmail.com
createContact
updateContact
deleteContact

Features of OOP: 4 pillars

- **Inheritance**: When one object acquires all the properties and behaviours of parent object i.e. known as inheritance. It provides code reusability.
- **Polymorphism**: When one task is performed by different ways i.e. known as polymorphism. For example: to convince the customer differently, to draw something, shape or rectangle etc. In Java, we use method overloading and method overriding to achieve polymorphism.
- **Abstraction**: Hiding internal details and showing functionality. In Java, we use abstract class and interface to achieve abstraction.
- **Encapsulation**: Binding (or wrapping) code and data together into a single unit is known as encapsulation. For example: capsule, it is wrapped with different medicines. A Java class is the example of encapsulation. Java bean is the fully encapsulated class because all the data members are private here.

What is Object-Orientation

- Abstraction and Encapsulation

Abstraction

- **Focus on the essential**
- **Omits tremendous amount of details**
- **…Focus on what an object "is and does"**

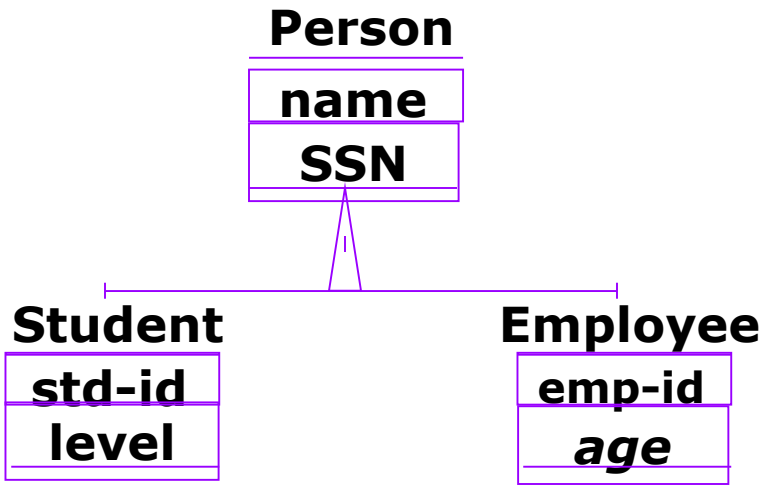
Encapsulation

- **a.k.a. information hiding**
- **Objects encapsulate:**
  - **property**
  - **behavior as a collection of methods invoked by messages**
  - **…state as a collection of instance variables**

What is Object-Orientation

- Subclass vs. Superclass / Inheritance

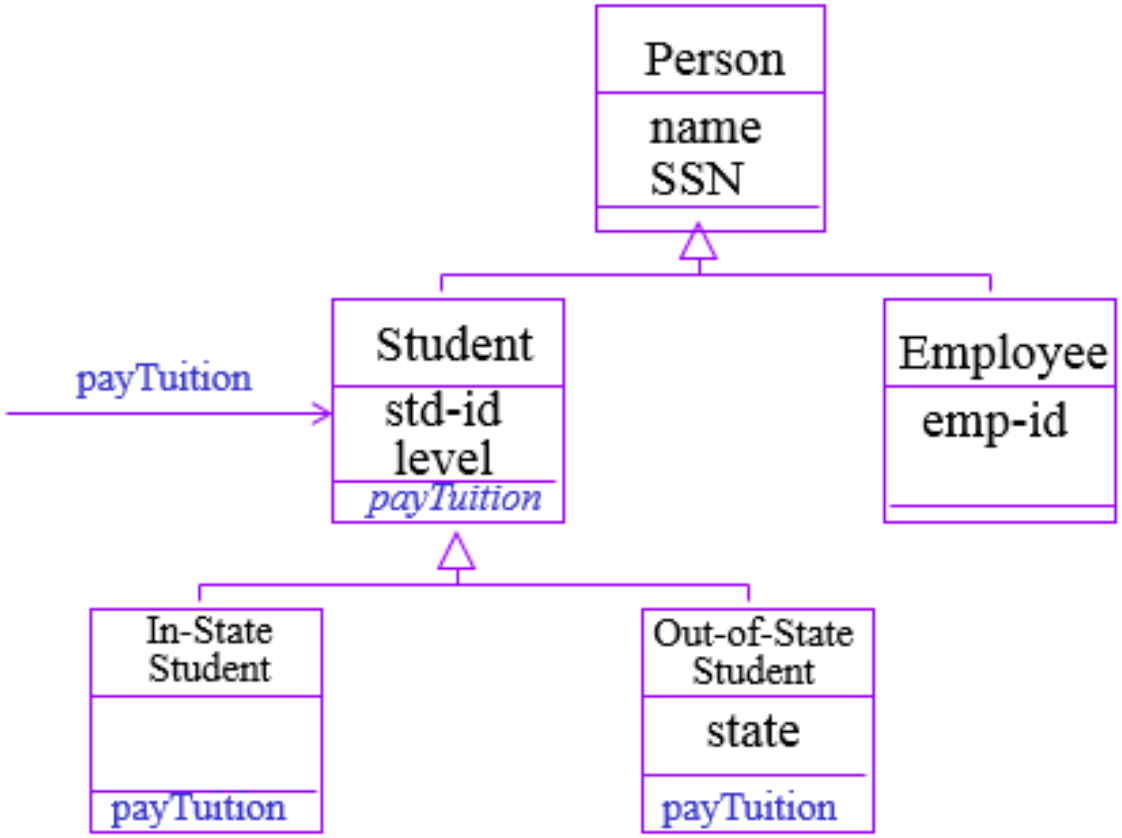
- **Specialization**: The act of defining one class as a refinement of another.
- **Subclass**: A class defined in terms of a specialization of a superclass using inheritance.
- **Superclass**: A class serving as a base for inheritance in a class hierarchy
- **Inheritance**: Automatic duplication of superclass attribute and behavior definitions in subclass.



multiple inheritance?

What is Object-Orientation
- Polymorphism

Objects of different classes respond to the same message differently.



Advantages of Object Oriented Approach

- Realistic Modelling

Advantages of Object Oriented Programing

- Realistic Modelling

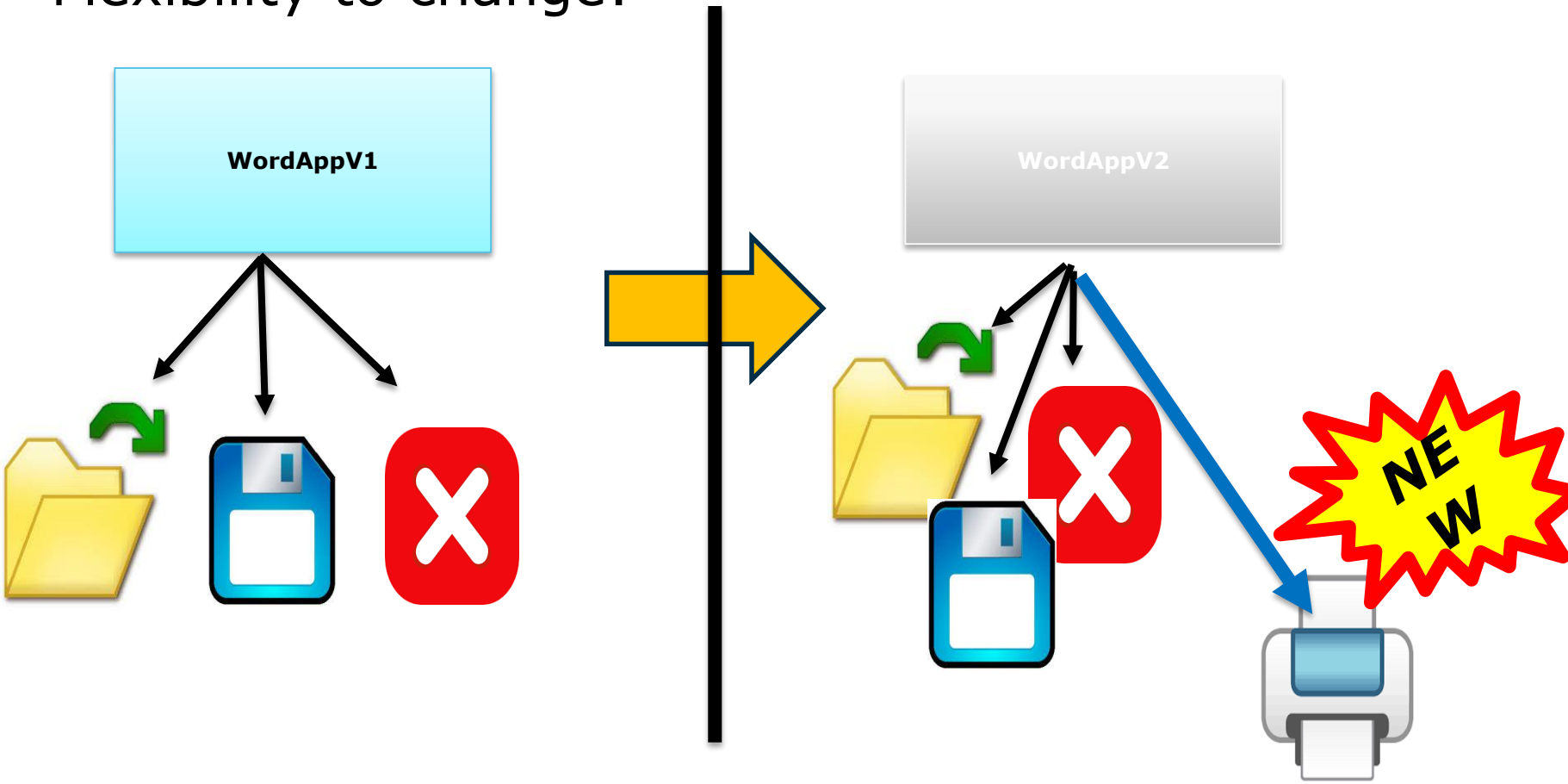
Bike
String color; String model; Integer speed;
Accelerate() Decelerate() Break()

Advantages of Object Oriented Approach

- Code Reusability

Advantages of Object Oriented Programing

- Flexibility to change:



Advantages of Object Oriented

- Modularity

Java & JEE Training

Object Oriented Analysis and Design

MindsMapped Consulting

Difference between Analysis and Design…

- A basic system (or subsystem)

Difference between Analysis and Design…

- Analysis

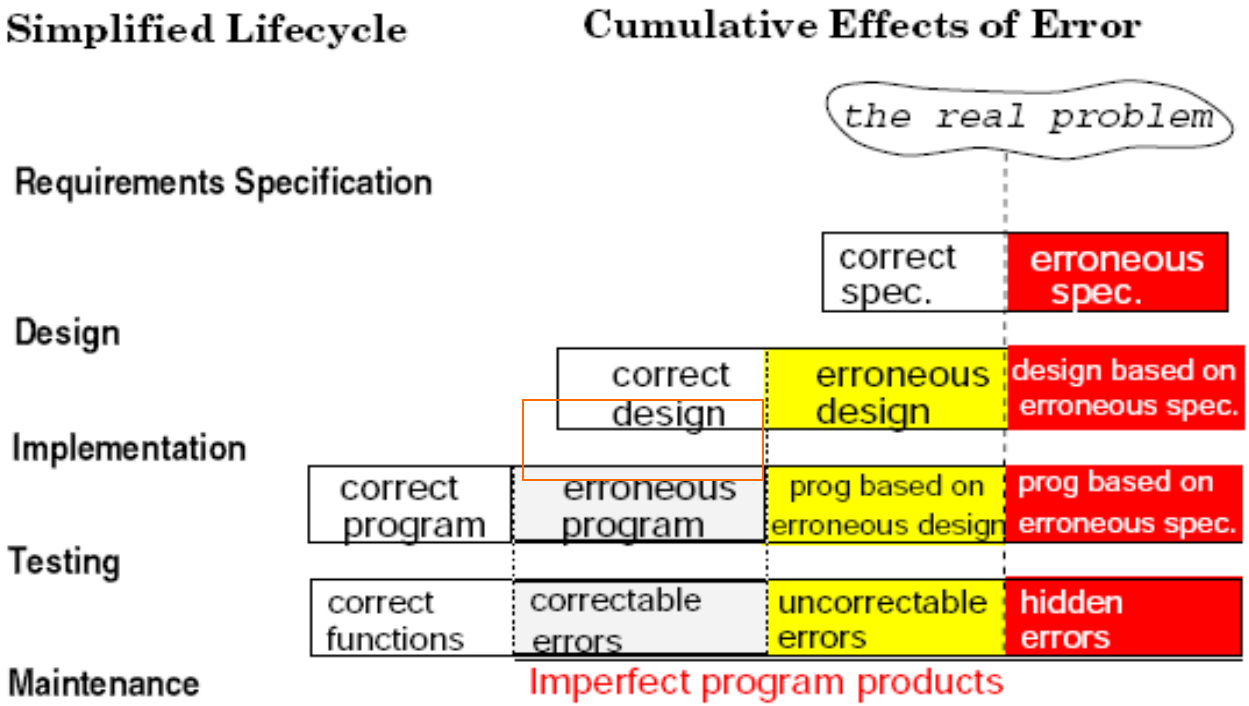
- Design

What is OOAD?

- Analysis — understanding, finding and describing concepts in the problem domain.
- Design — understanding and defining software solution/objects that represent the analysis concepts and will eventually be implemented in code.
- OOAD — Analysis is object-oriented and design is object-oriented. A software development approach that emphasizes a logical solution based on objects.

Maintainability through Traceability!

Error Propagation in Lifecycle [Mizuno82]



Artificial problem

Accidental design

How big is the erroneous spec.?
How costly is it?

Traceability => Maintainability

More later…

- More practical examples of OOAD in later sessions…
- Now let us start looking at Object Oriented Programming concepts using Java

Java & JEE Training

Object Oriented Programming with Java - Basic Class Demo

MindsMapped Consulting

Naming Conventions

Name	Convention
class name	should start with uppercase letter and be a noun e.g. String, Color, Button, System, Thread etc.
interface name	should start with uppercase letter and be an adjective e.g. Runnable, Remote, ActionListener etc.
method name	should start with lowercase letter and be a verb e.g. actionPerformed(), main(), print(), println() etc.
variable name	should start with lowercase letter e.g. firstName, orderNumber etc.
package name	should be in lowercase letter e.g. java, lang, sql, util etc.
constants name	should be in uppercase letter. e.g. RED, YELLOW, MAX_PRIORITY etc.

Object and Class in Java - Demo

A basic example of a class:

```
class Student1{
 int id;//data member (also instance variable)
 String name;//data member(also instance variable)

 public static void main(String args[]){
  Student1 s1=new Student1();//creating an object of Student
  System.out.println(s1.id);
  System.out.println(s1.name);
 }
}
```

Another Example of Objects and Classes in Java

```
class Student2{
 int rollno;
 String name;

 void insertRecord(int r, String n){  //method
  rollno=r;
  name=n;
 }

 void displayInformation(){System.out.println(rollno+" "+name);}//method

 public static void main(String args[]){
  Student2 s1=new Student2();
  Student2 s2=new Student2();

  s1.insertRecord(111,"Karan");
  s2.insertRecord(222,"Aryan");

  s1.displayInformation();
  s2.displayInformation();

 }
}
```