COJ :: Getting Started With Java

TalentSprint

Licensed To Skill

Version 1.0.4

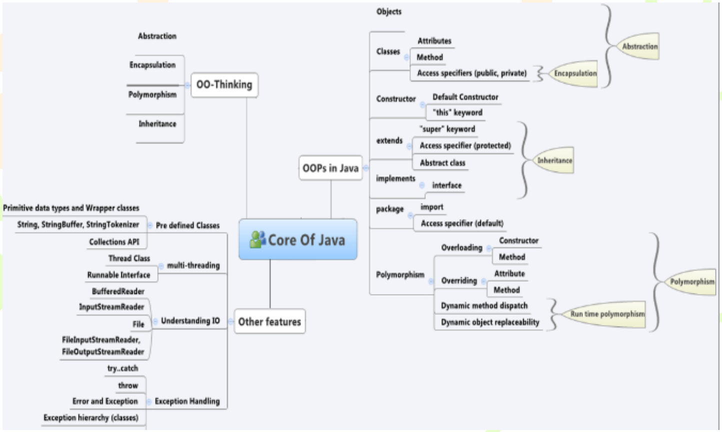
OO Thinking

The content in this presentation is aimed at teaching learners to

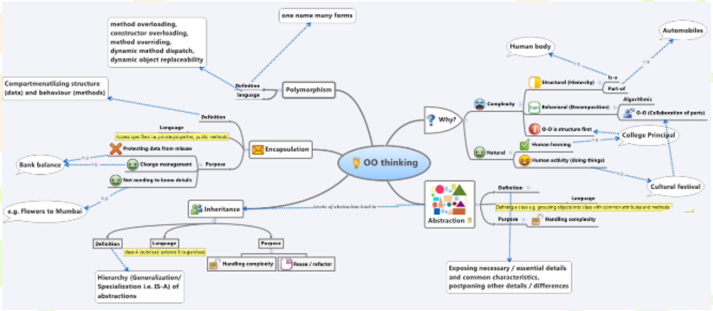
- Identify various concepts of OOPs
- Relate Object-Oriented approach to the process of understanding and analyzing complex systems in the real world.

OO Thinking

Module Snapshot

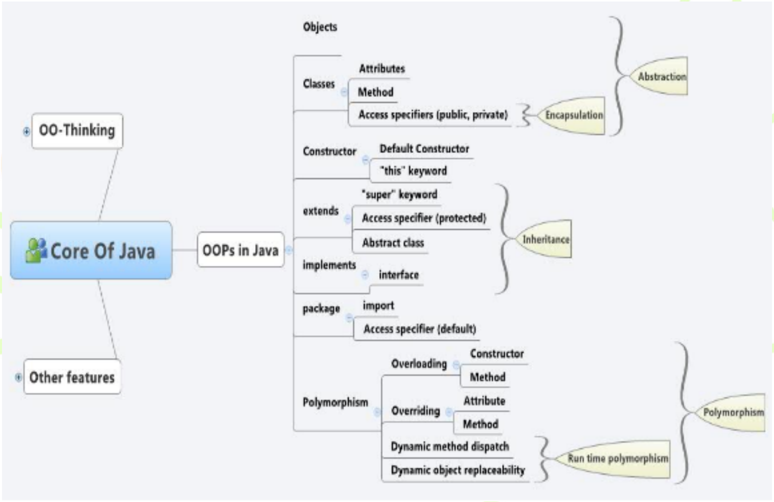


OO Thinking


one name many forms
method overloading, constructor overloading, method overriding, dynamic method dispatch, dynamic object replaceability
Definition
language
Polymorphism
Compartmenatlizing structure (data) and behaviour (methods)
Definition
Language
Access specifiers i.e. private properties, public methods
Protecting data from misuse
Bank balance
e.g.
Change management
Purpose
Not needing to know details
e.g. Flowers to Mumbai
Encapsulation
OO thinking
Inheritance
Levels of abstraction lead to
Definition
Language
class A (subclass) extends B (superclass)
Purpose
Handling complexity
Reuse / refactor
Hierarchy (Generalization/ Specialization i.e. IS-A) of abstractions
Why?
Complexity
Structural (Hierarchy)
Is-a
Part-of
Human body
Automobiles
Behavioral (Decomposition)
Algorithmic
O-O (Collaboration of parts)
O-O is structure first
College Principal
Natural
Human learning
Human activity (doing things)
Cultural festival
Abstraction
Definition
Language
Defining a class e.g. grouping objects into class with common attributes and methods
Purpose
Handling complexity
Exposing necessary / essential details and common characteristics, postponing other details / differences

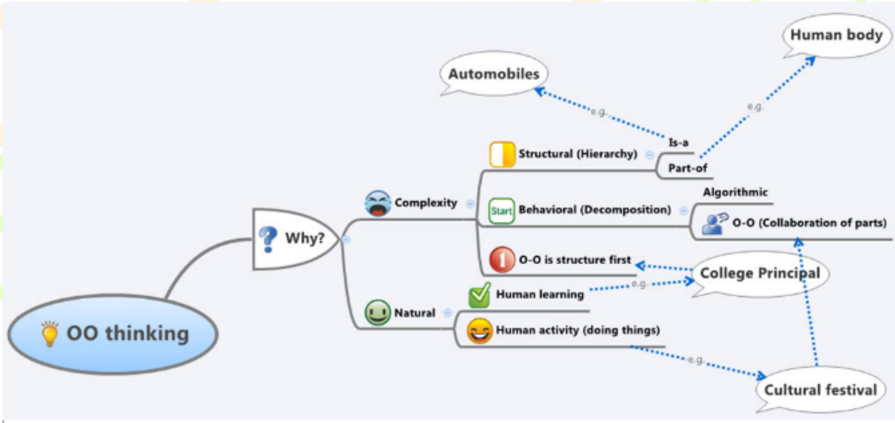
OO Thinking

OOPs in Java



OO Thinking

Why OO thinking?

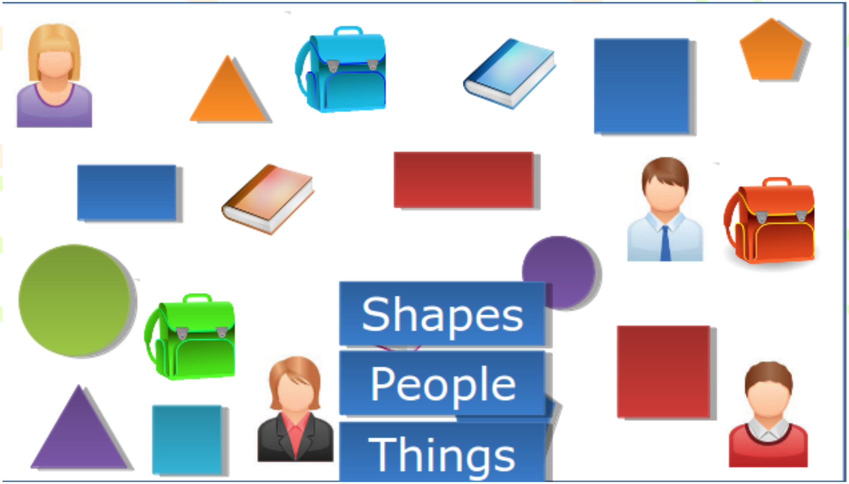


OO Thinking

Dealing with complexity

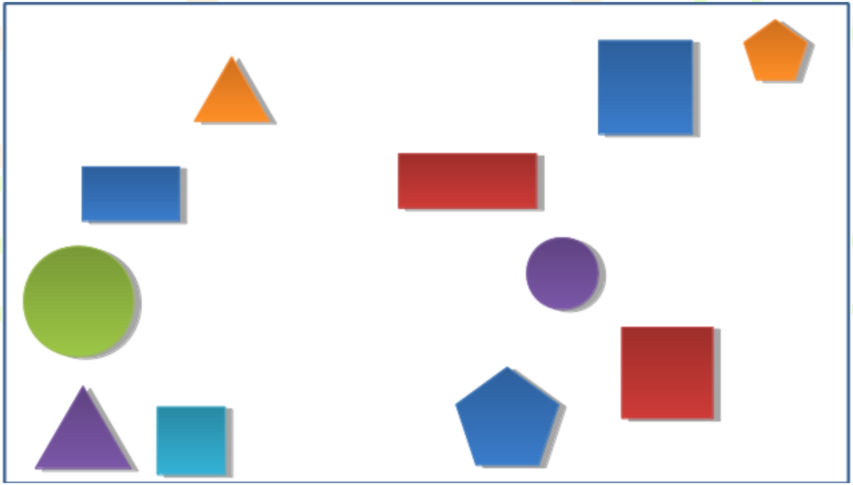
- Complexity can be in structure as well as behavior.
- The structure of complex systems is Hierarchic, - IS A hierarchy and part of hierarchy.
- Behavioral complexity can be handled by collaboration of its parts.

OO Thinking
Structural Complexity



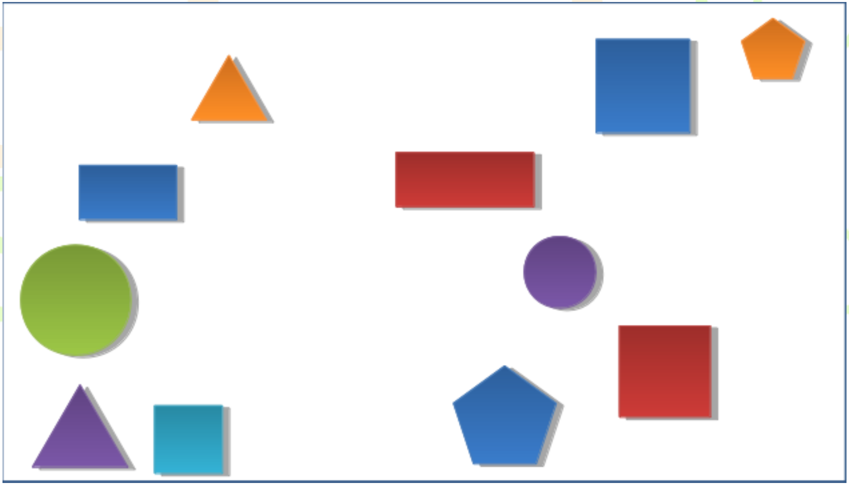
OO Thinking

Structural Complexity



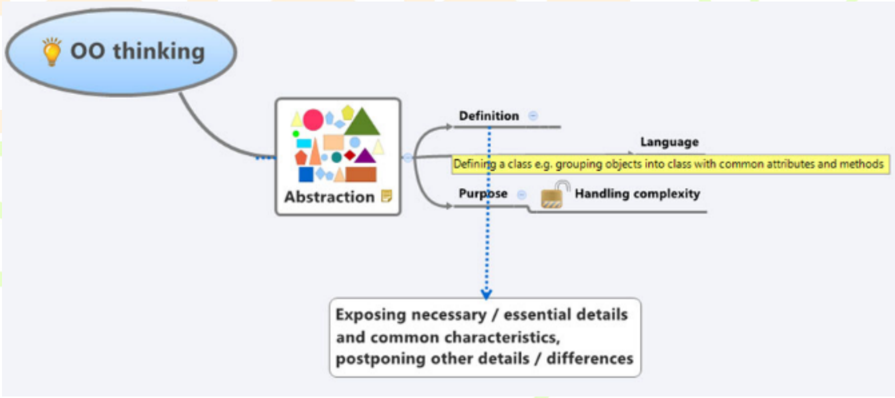
OO Thinking

Structural Complexity



OO Thinking

Abstraction



OO thinking
Abstraction
Definition
Language
Defining a class e.g. grouping objects into class with common attributes and methods
Purpose
Handling complexity
Exposing necessary / essential details and common characteristics, postponing other details / differences

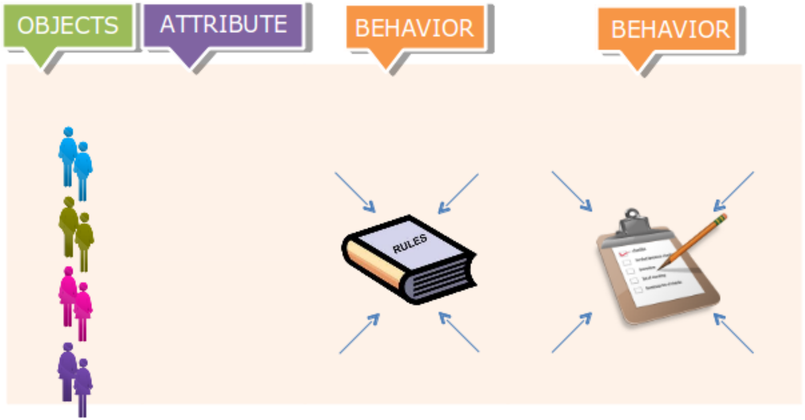
OO Thinking

Abstraction

Examples of Abstraction

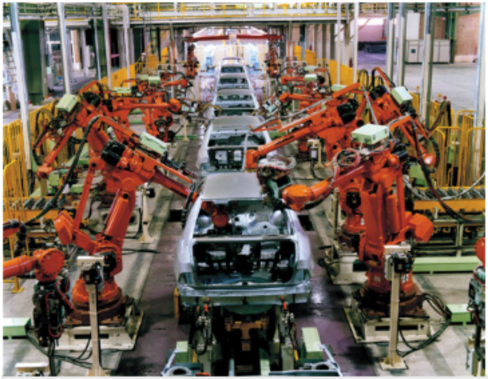
OO Thinking

Examples of Abstraction : Student



OO Thinking

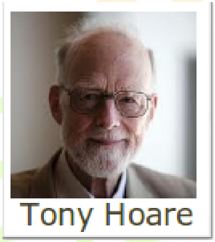
Examples of Abstraction : Manufacturing of Automobiles



Many Cars Same Process

OO Thinking

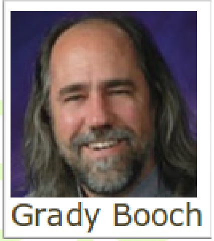
Other definitions of Abstraction:



"Abstraction arises from a recognition of similarities between certain objects, situations, or processes in the real world, and the decision to concentrate upon those similarities and to ignore for the time being the differences."

OO Thinking

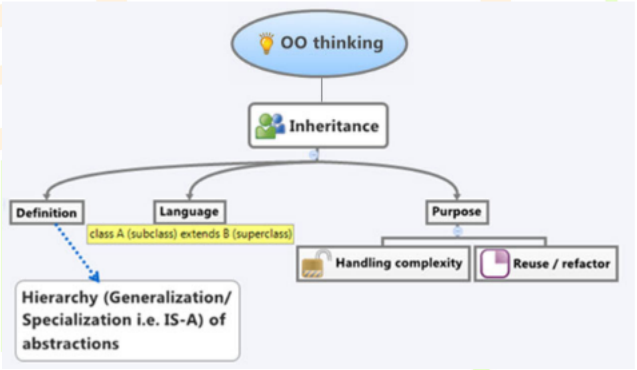
Other definitions of Abstraction:



"An abstraction denotes the essential characteristics of an object that distinguish it from all other kinds of objects and thus provide crisply defined conceptual boundaries, relative to the perspective of the viewer."

OO Thinking

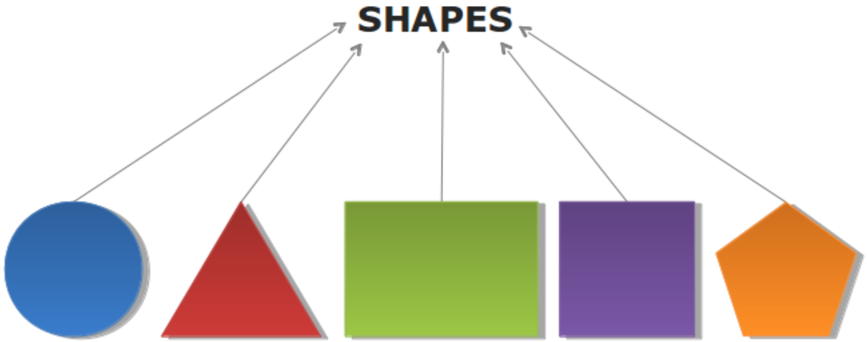
Inheritance



OO thinking
Inheritance
Definition
Language
class A (subclass) extends B (superclass)
Purpose
Handling complexity
Reuse / refactor
Hierarchy (Generalization/ Specialization i.e. IS-A) of abstractions

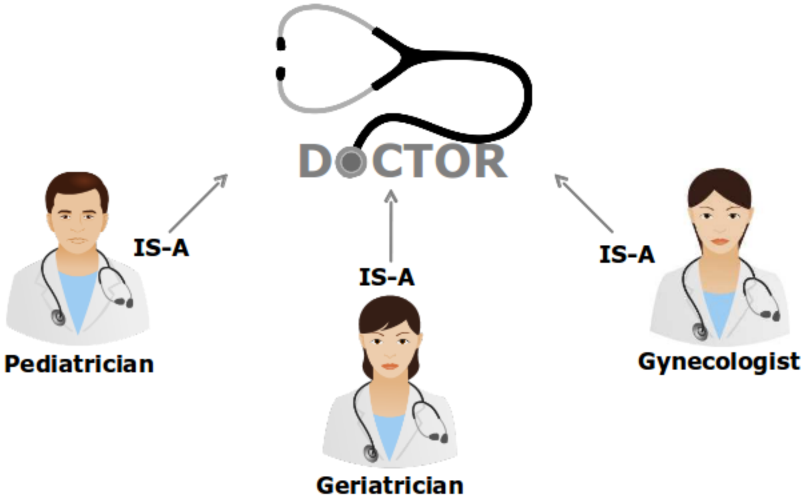
OO Thinking

IS-A Hierarchy



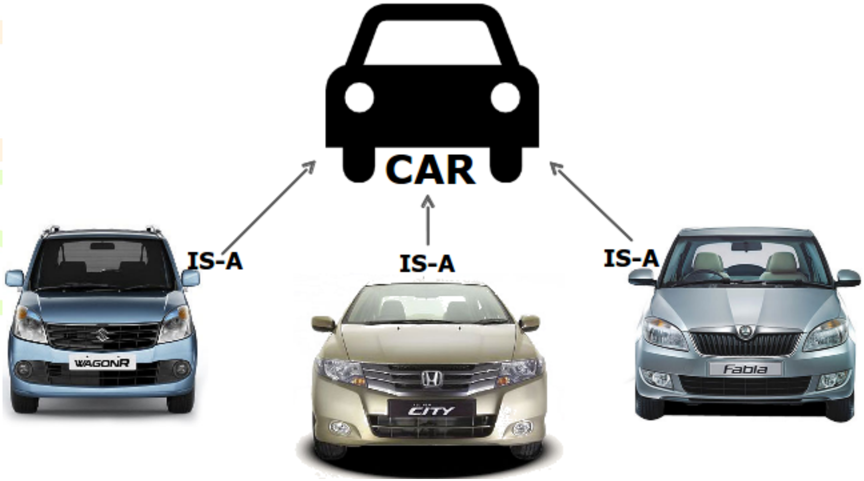
OO Thinking

Inheritance Example : Doctor



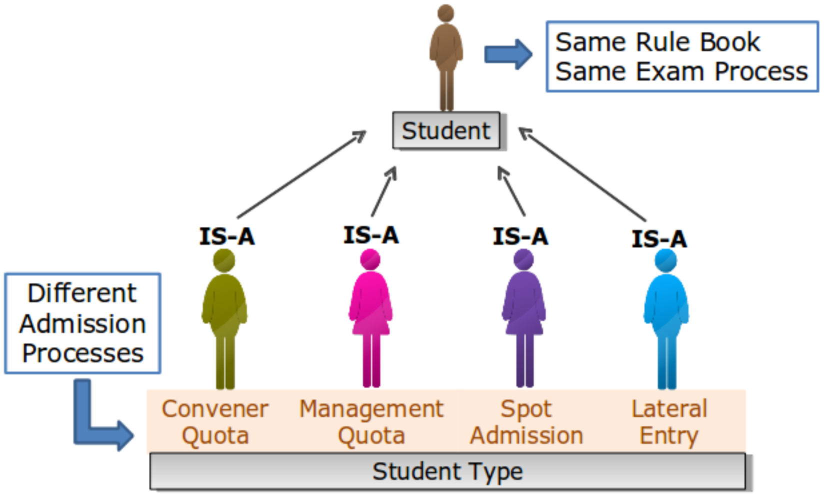
OO Thinking

Inheritance Example: Car



OO Thinking

Inheritance Example: Student



OO Thinking

Encapsulation

Tablet

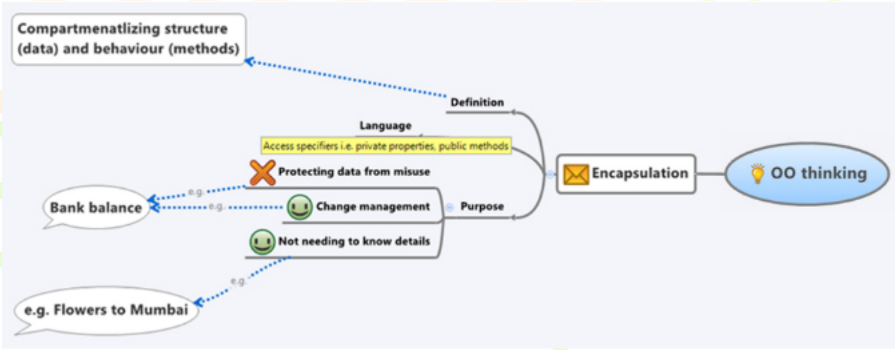
Capsule

Which one you prefer?

Prefer capsule to avoid bitter taste of medicine.

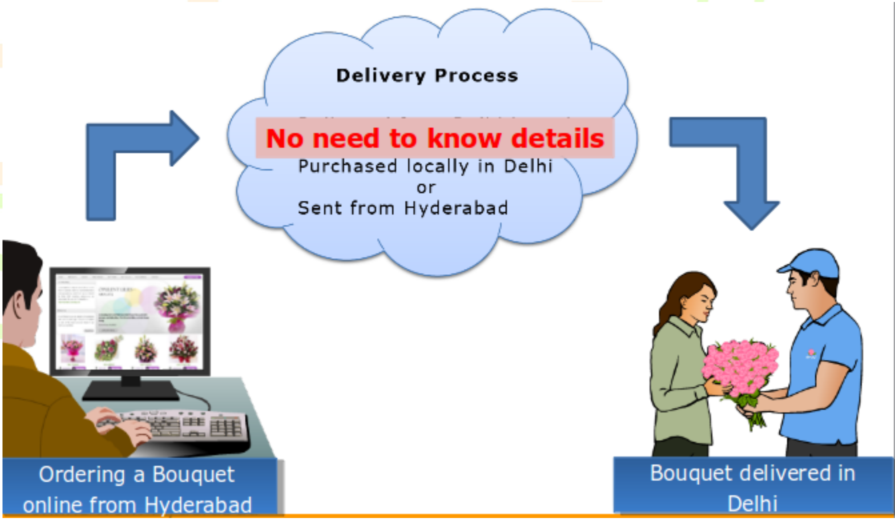
OO Thinking

Encapsulation


OO thinking
Encapsulation
Definition
Compartmenatlizing structure (data) and behaviour (methods)
Language
Access specifiers i.e. private properties, public methods
Purpose
Protecting data from misuse
Change management
Not needing to know details
e.g.
e.g.
Bank balance
e.g.
e.g. Flowers to Mumbai

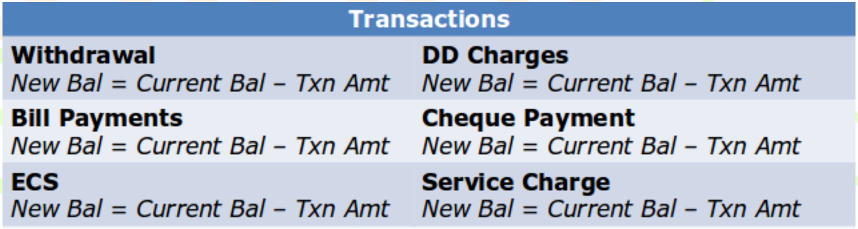
OO Thinking

Encapsulation Example: Bouquet to Delhi


Delivery Process
No need to know details
Purchased locally in Delhi
or
Sent from Hyderabad
Ordering a Bouquet online from Hyderabad
Bouquet delivered in Delhi

OO Thinking

Encapsulation Example: Bank Account Balance



Transactions	
Withdrawal *New Bal = Current Bal – Txn Amt*	**DD Charges** *New Bal = Current Bal – Txn Amt*
Bill Payments *New Bal = Current Bal – Txn Amt*	**Cheque Payment** *New Bal = Current Bal – Txn Amt*
ECS *New Bal = Current Bal – Txn Amt*	**Service Charge** *New Bal = Current Bal – Txn Amt*

OO Thinking

Encapsulation Example: Bank Account Balance

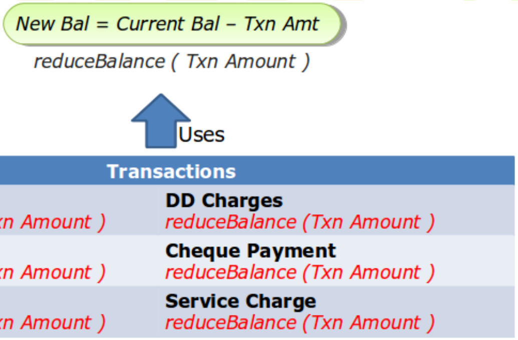
Requirement Changed :Minimum Balance Required Rs 5000

Transactions	
Withdrawal *New Bal = Current Bal – Txn Amt* *if New Bal < 5000 Decline Txn*	**DD Charges** *New Bal = Current Bal – Txn Amt* *if New Bal < 5000 Decline Txn*
Bill Payments *New Bal = Current Bal – Txn Amt* *if New Bal < 5000 Decline Txn*	**Cheque Payment** *New Bal = Current Bal – Txn Amt* *if New Bal < 5000 Decline Txn*
ECS *New Bal = Current Bal – Txn Amt* *if New Bal < 5000 Decline Txn*	**Service Charge** *New Bal = Current Bal – Txn Amt* Forgot to change

Change Management : So Many Changes

OO Thinking

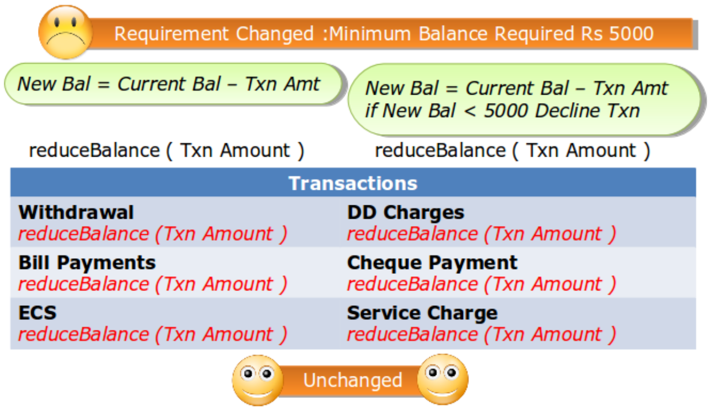
Encapsulation Example: Bank Account Balance



Transactions	
Withdrawal reduceBalance (Txn Amount)	**DD Charges** reduceBalance (Txn Amount)
Bill Payments reduceBalance (Txn Amount)	**Cheque Payment** reduceBalance (Txn Amount)
ECS reduceBalance (Txn Amount)	**Service Charge** reduceBalance (Txn Amount)

OO Thinking

Encapsulation Example: Bank Account Balance



Requirement Changed :Minimum Balance Required Rs 5000

New Bal = Current Bal – Txn Amt	*New Bal = Current Bal – Txn Amt* *if New Bal < 5000 Decline Txn*
reduceBalance (Txn Amount)	reduceBalance (Txn Amount)

Transactions	
Withdrawal *reduceBalance (Txn Amount)*	**DD Charges** *reduceBalance (Txn Amount)*
Bill Payments *reduceBalance (Txn Amount)*	**Cheque Payment** *reduceBalance (Txn Amount)*
ECS *reduceBalance (Txn Amount)*	**Service Charge** *reduceBalance (Txn Amount)*

Unchanged

OO Thinking

Encapsulation Example: Catering

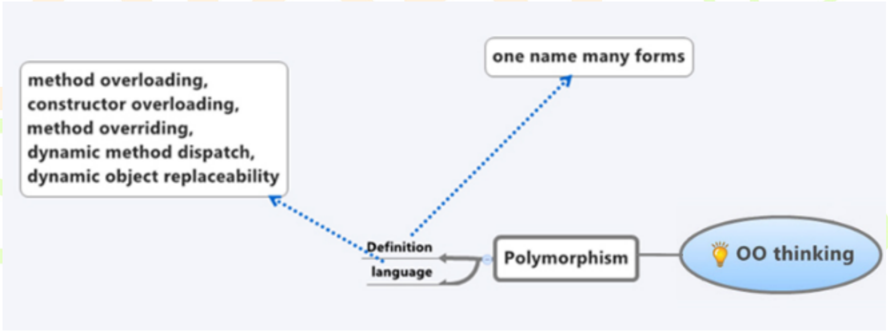
No need to know details

OO Thinking

Polymorphism

OO Thinking

Polymorphism



OO Thinking