Object-Oriented Programming Concepts

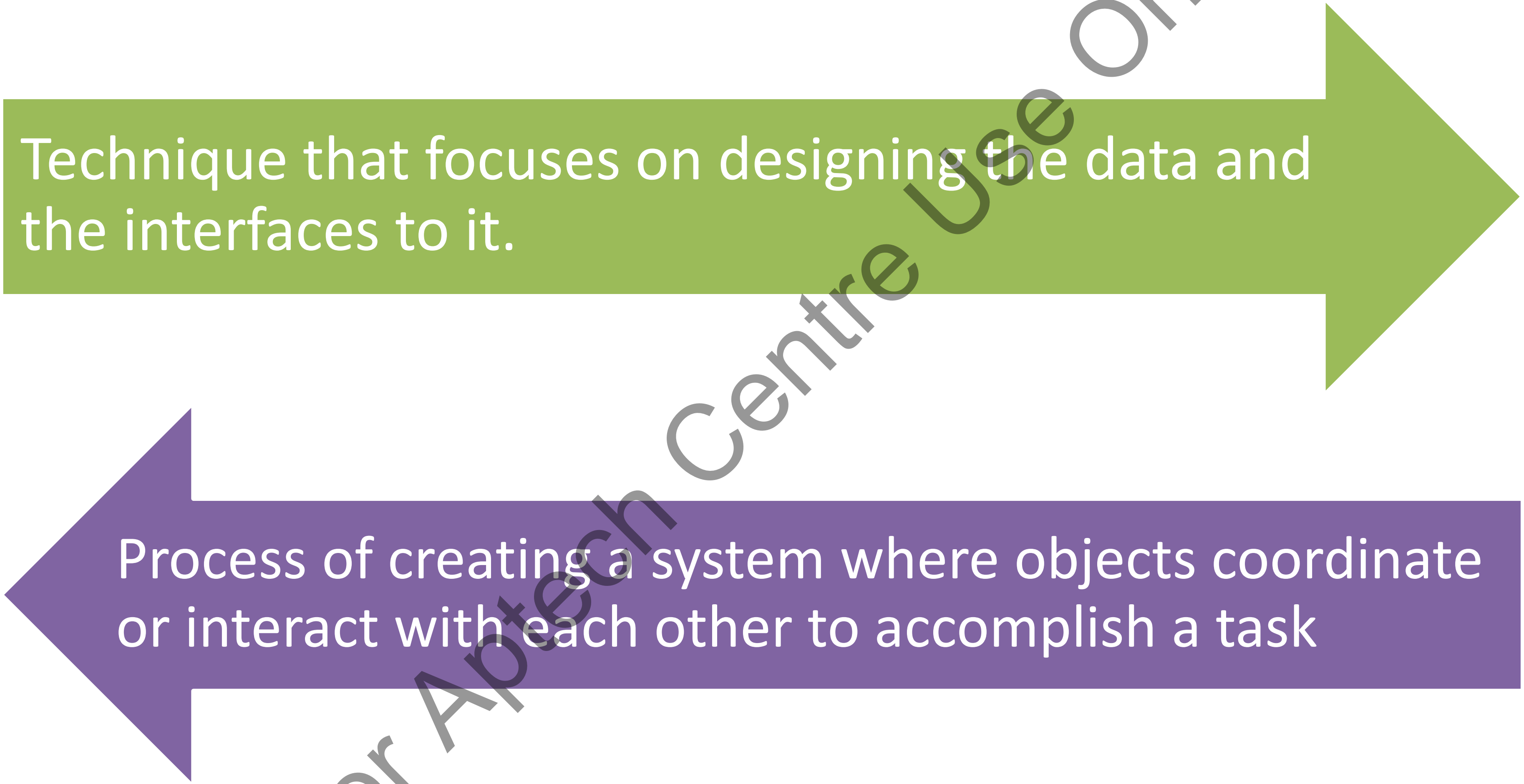
Session: 2

Object-oriented Design

Objectives

- Explain Object-oriented design (OO Design)
- Describe Responsibility-driven Design (RDD)
- Explain Agents, Classes, and Instances
- Describe Methods, Responsibilities, and Modules
- Explain Generalization, Specialization, and Patterns
- Explain Coupling and Cohesion

OO Design 1-4

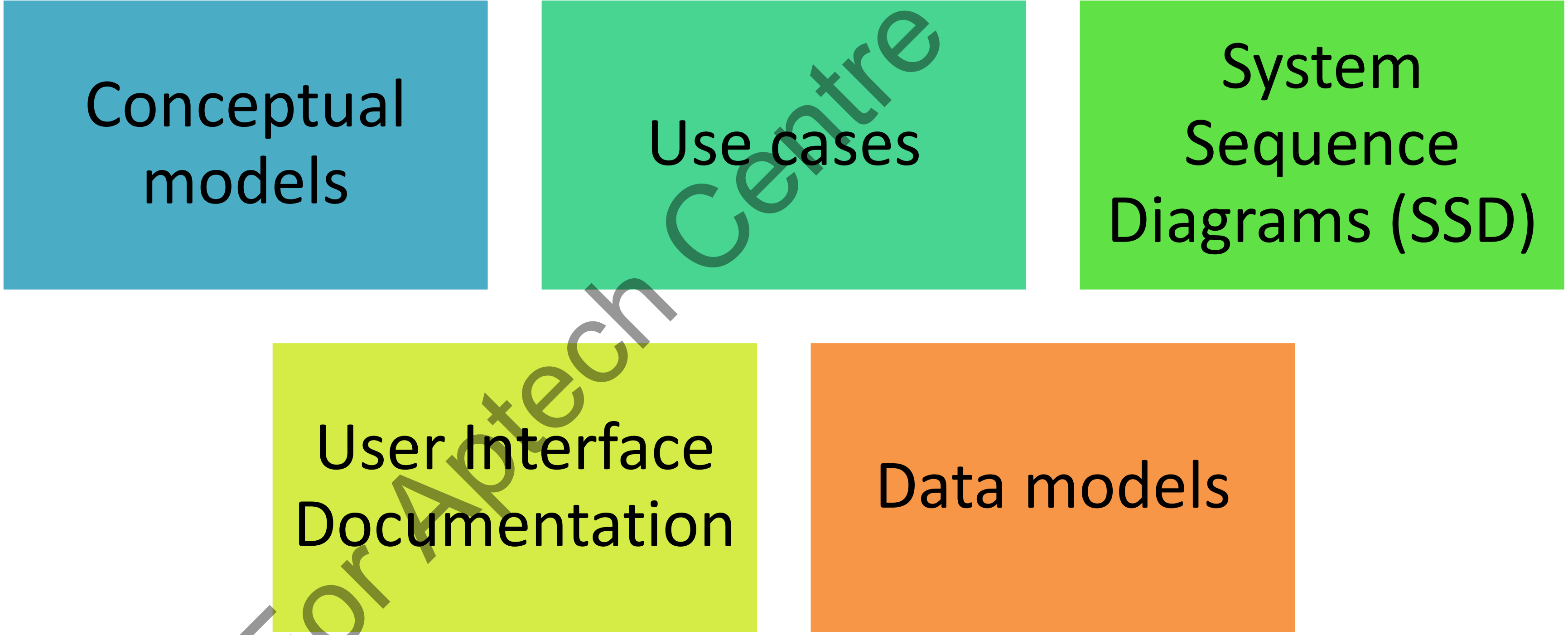


Technique that focuses on designing the data and the interfaces to it.

Process of creating a system where objects coordinate or interact with each other to accomplish a task

OO Design 2-4

- The input for OO Design is provided by Object-oriented analysis in the form of:



Conceptual models

Use cases

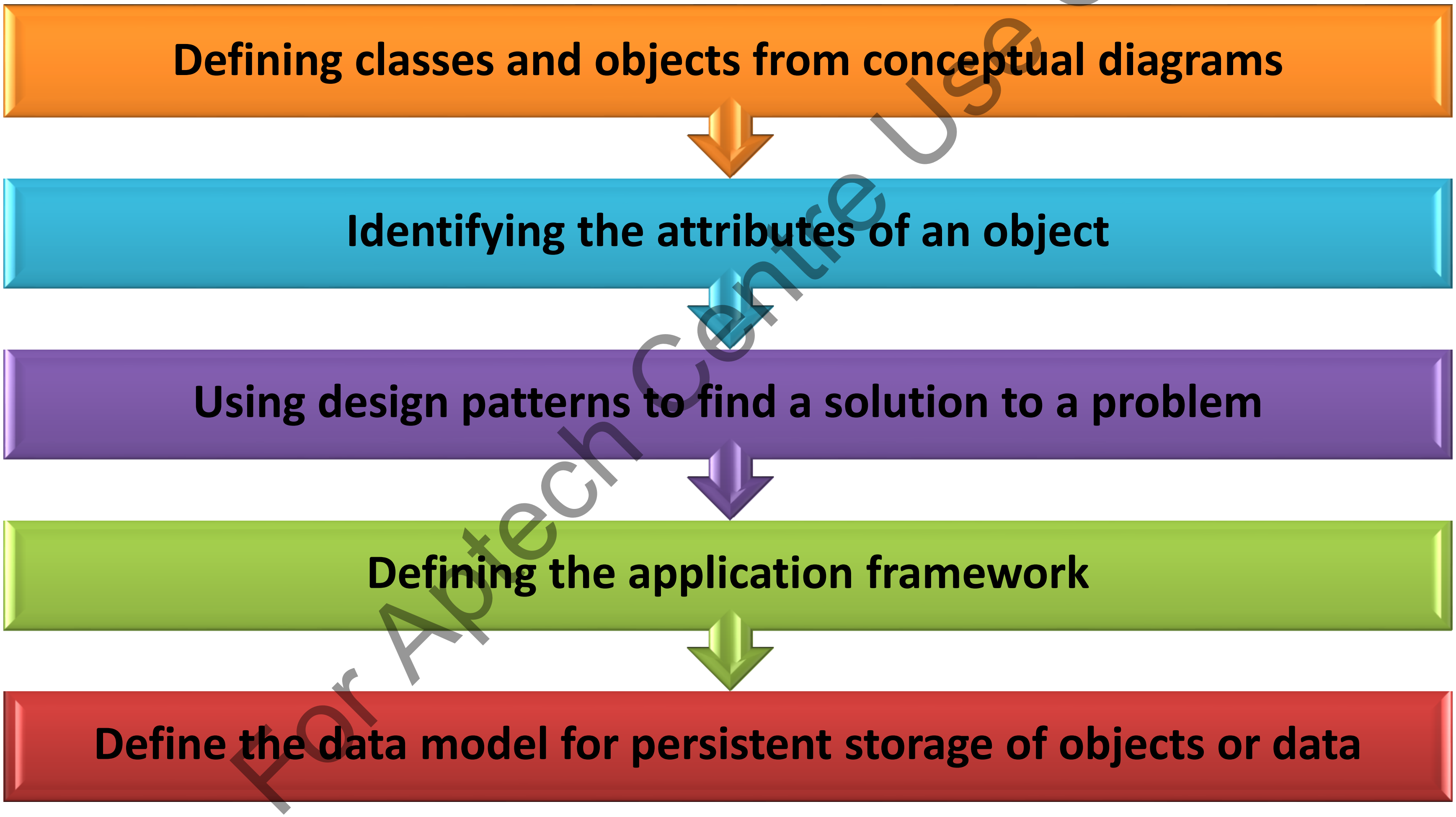
System Sequence Diagrams (SSD)

User Interface Documentation

Data models

OO Design 3-4

- The various tasks involved in OO Design are as follows:



Defining classes and objects from conceptual diagrams

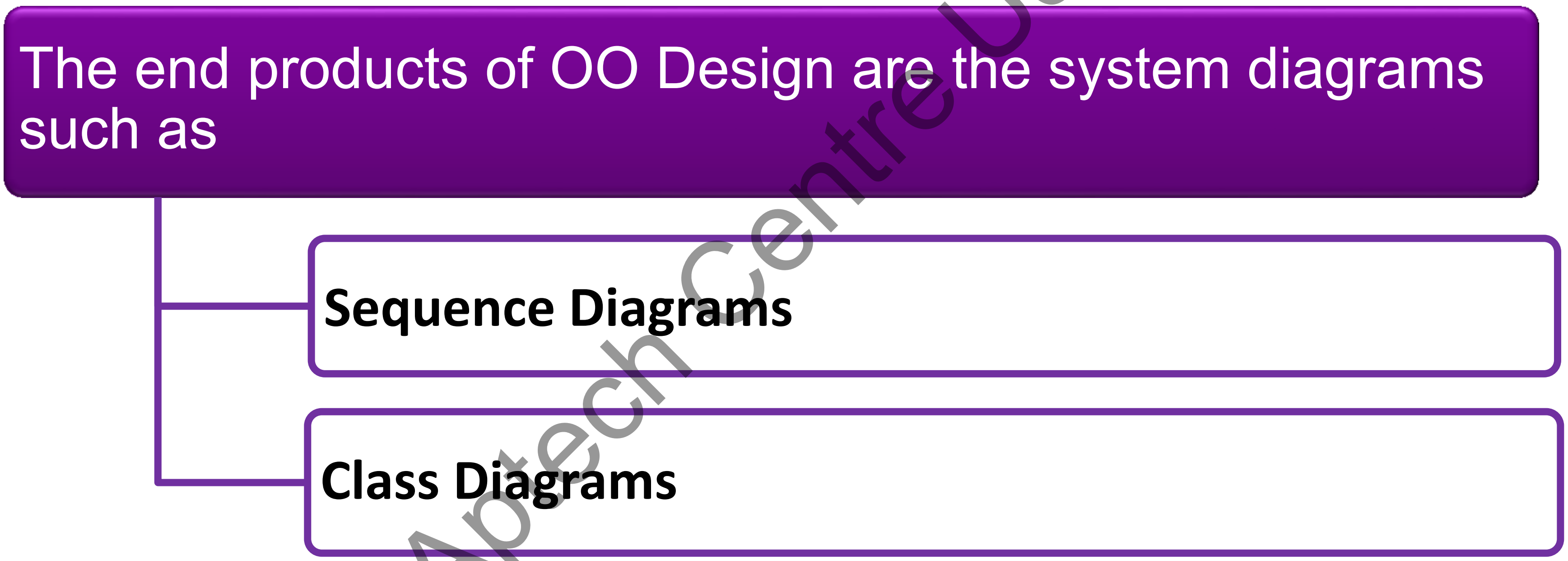
Identifying the attributes of an object

Using design patterns to find a solution to a problem

Defining the application framework

Define the data model for persistent storage of objects or data

OO Design 4-4



The end products of OO Design are the system diagrams such as

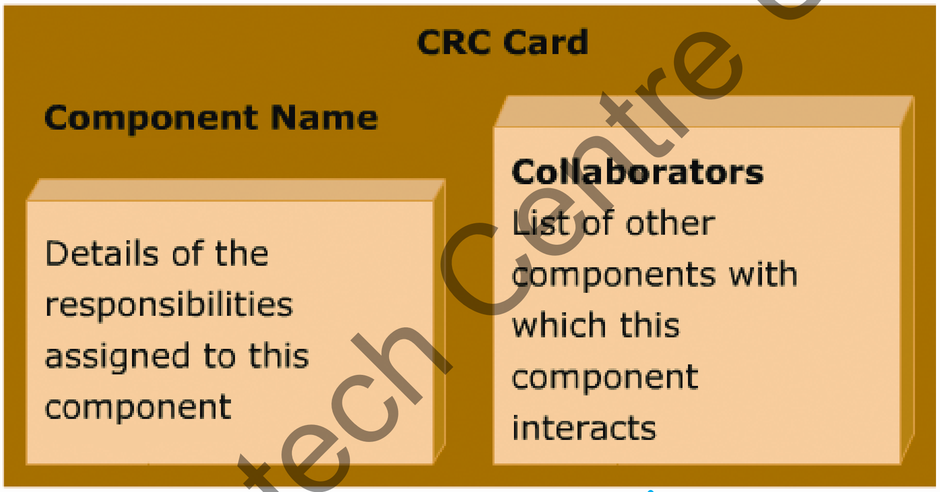
- Sequence Diagrams
- Class Diagrams

Responsibility-driven Design 1-2

- A design technique
- Based on delegation of responsibilities and determination of components responsible for executing the responsibilities

Responsibility-driven Design 2-2

To describe each component and its responsibility, the design team prepares small index cards.



CRC Card

Component Name

Details of the responsibilities assigned to this component

Collaborators
List of other components with which this component interacts

Each card contains the name of the software component, the responsibilities and names of other components to which it interacts.

Software Concepts

- Based on RDD, several other software concepts have been defined that are as follows:
  - Communities of Agents
  - Classes and Objects
  - Messages and Methods
  - Responsibilities
  - Modules
  - Generalization and Specialization
  - Patterns
  - Coupling and Cohesion

Communities of Agents 1-2

- An object-oriented program is viewed as a community of interacting agents called objects.

- Each object plays a unique role and provides services to or performs a task for the other members of the community.

Communities of Agents 2-2

- Consider an example where a person named Bob wants to send a cake to a friend named Brian.

Bob goes to William's bakery to order a cake for Brian

William makes the cake

Delivery boy drives to Brian's house

Brian receives the cake

Classes and Objects 1-2

- A class or a category is a group of people or things having common characteristics

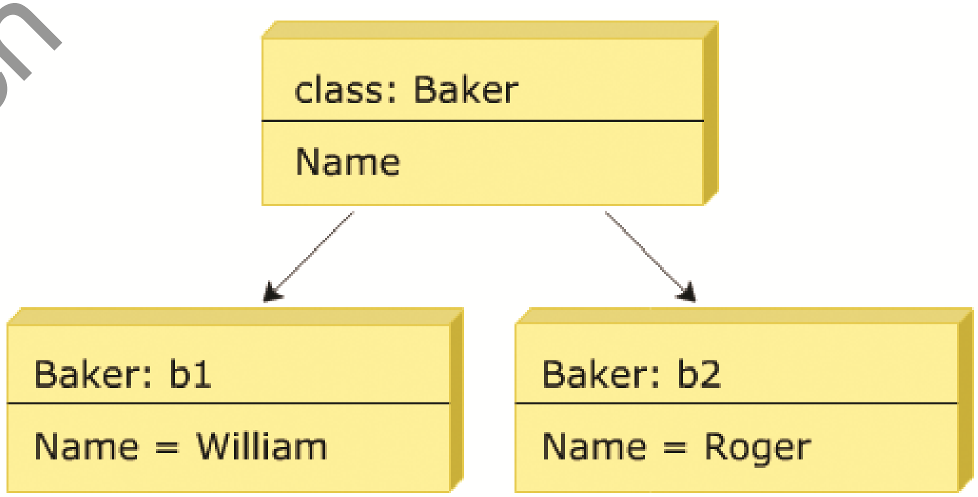
- The figure shows various objects arranged according to their category or class.

Classes and Objects 2-2

- An object is one copy of the class.

- Object can also be referred to as an instance of a class.

- The figure shows two objects of class Baker namely b1 and b2 and the value of Name for b1 is William and that of b2 is Roger.

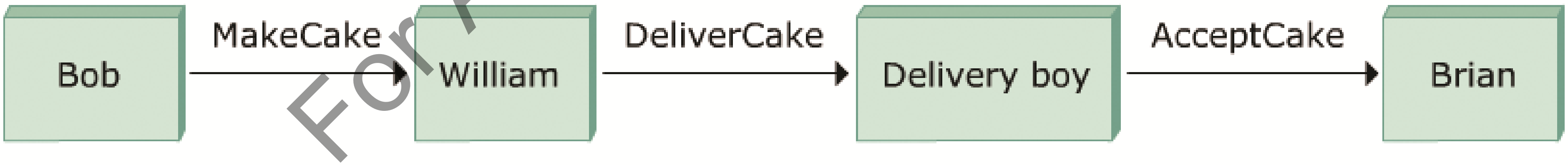


Messages and Methods 1-2

- A method is a set of steps or instructions, or a procedure to accomplish something

- Message is a set of words, a phrase, or statement to convey something

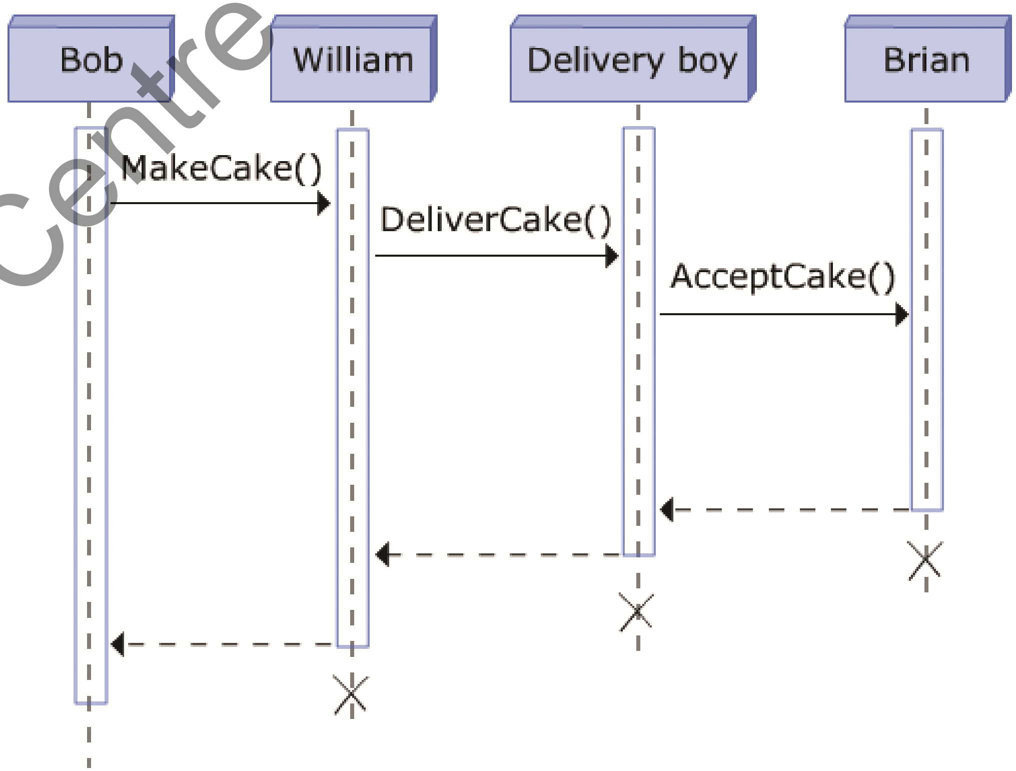
- The figure shows how the objects pass messages to each other to fulfill a task.



Messages and Methods 2-2

- Objects communicate with each other by invoking methods and passing messages or arguments to the methods to complete a task

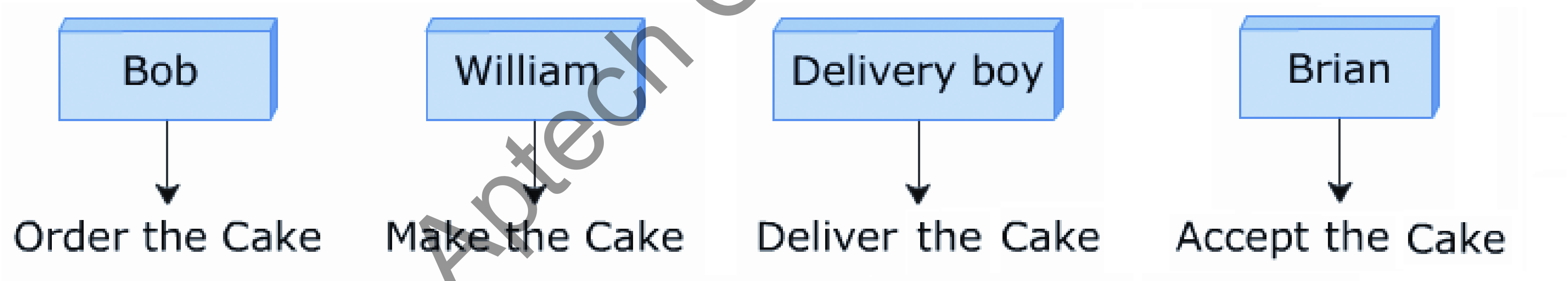
- The message passing and method invocation can also be better understood from the figure.



Responsibilities

- In OOP, describe the behavior of objects as responsibilities

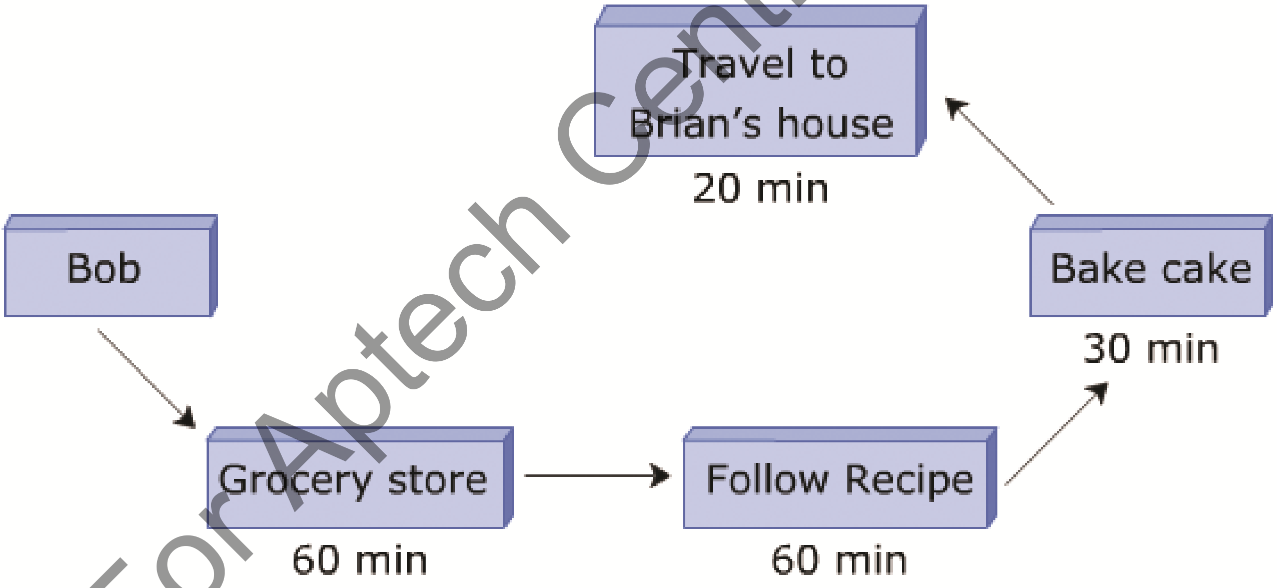
- The figure shows the responsibilities of various agents in the cake delivery process.



Modules 1-2

- To solve complex problems divide the problem into smaller units or modules

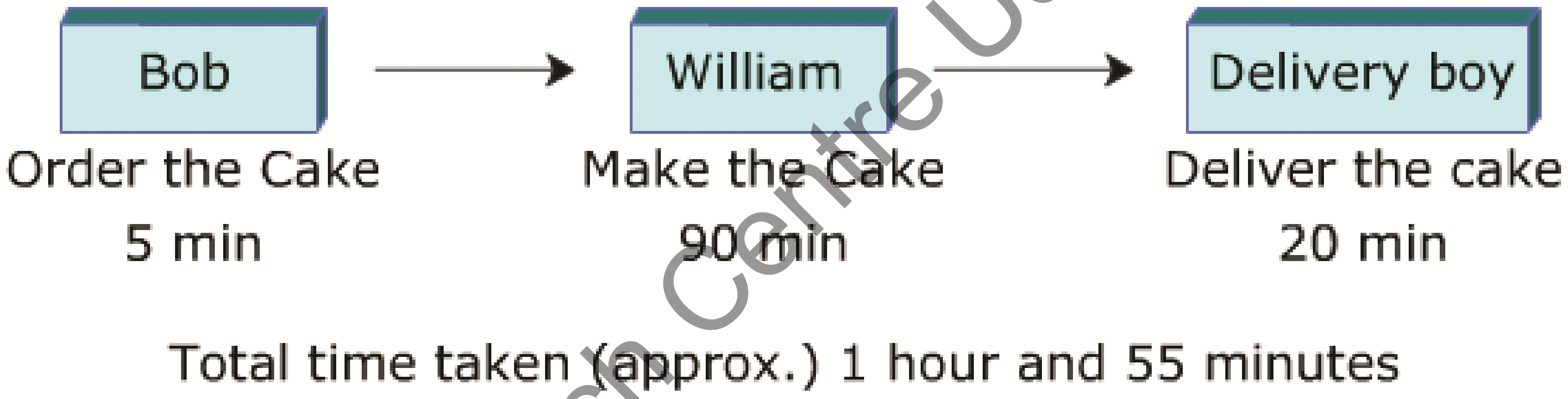
- The figure shows the approximate time it took to Bob to deliver the cake when he does everything himself.



Total time taken (approx.) 2 hours and 50 minutes

Modules 2-2

- The figure shows the time it took Bob when the task was divided among other agents.



Bob → William → Delivery boy

Order the Cake
5 min

Make the Cake
90 min

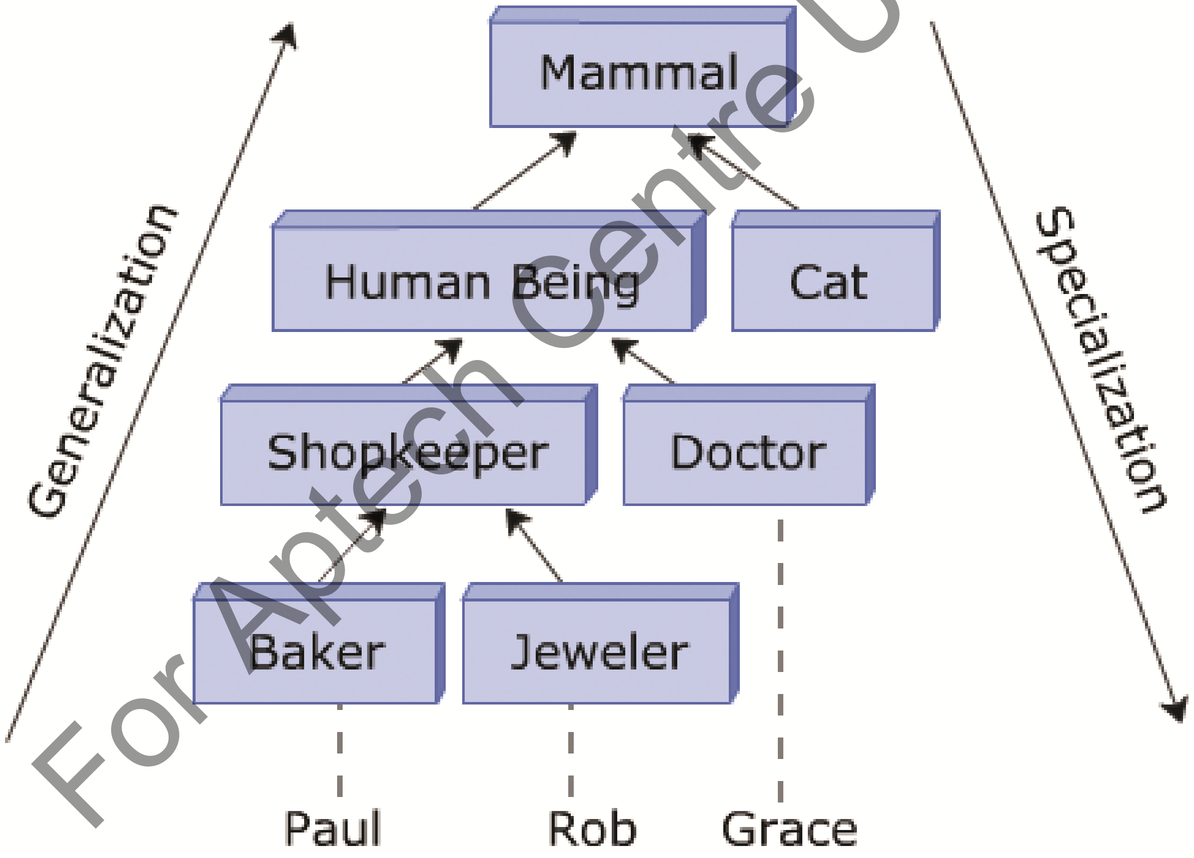
Deliver the cake
20 min

Total time taken (approx.) 1 hour and 55 minutes

- This shows that by dividing the task between multiple people, Bob was able to achieve his goal in almost half the time than he would have done it himself

Generalization and Specialization

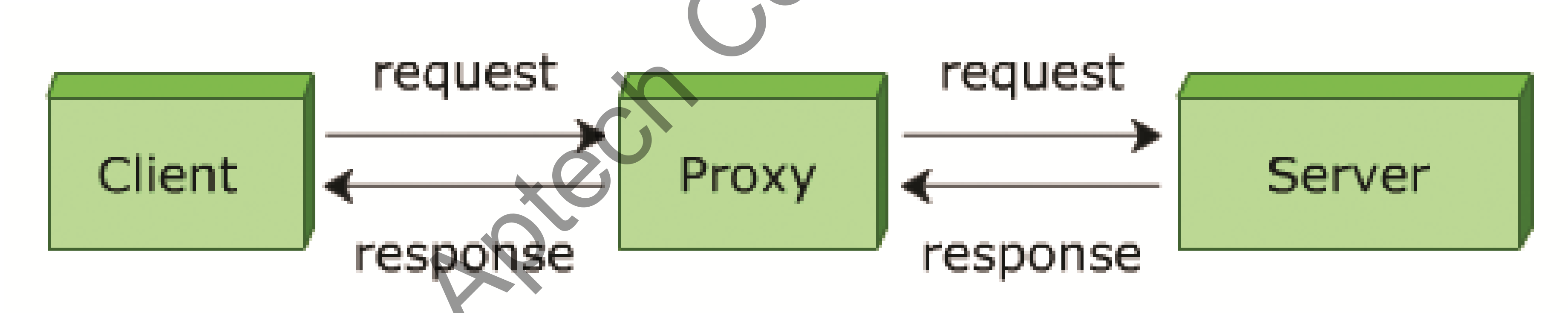
- The figure shows the concept of generalization and specialization.



Patterns

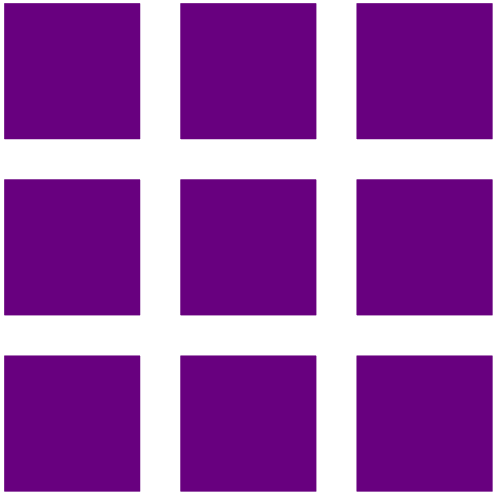
- A shape, model, or structure appearing again and again to create a design

- The figure shows the situation of a client making request through a proxy.



Coupling and Cohesion 1-2

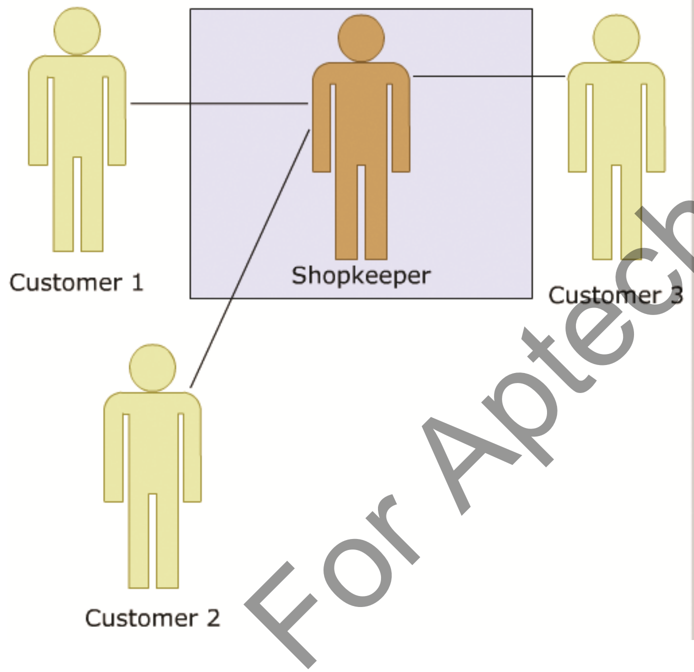
- Coupling is the degree to which the components know about each other
- Coupling can be loose or tight
- Loose coupling means a group of components which can be operate independent of each other



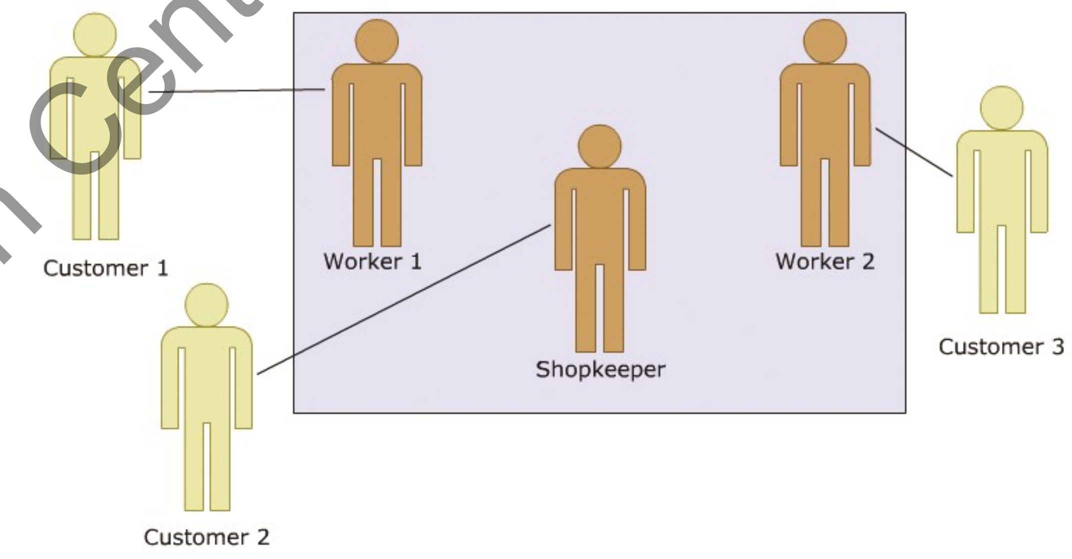
Coupling and Cohesion 2-2

- Cohesion deals with a single component and how focused it is on a specific task

- The figures show a less cohesive component and a highly cohesive component.



Less Cohesive



Highly Cohesive

Summary

- Responsibility-driven design is a design technique based on delegation of responsibilities and determination of components responsible for executing the responsibilities.
- An object-oriented program is viewed as a community of interacting agents called objects.
- A class is a structure that defines and encloses the data members and methods pertaining to a particular entity. An instance is one copy of the class.
- The commonly used method of solving complex problems is to divide the problem into smaller units called modules.
- A pattern is a solution to a problem which has been used effectively in a similar problem encountered earlier.
- Coupling is the degree to which the components know about each other. Cohesion is the degree to which a particular component is designed for specific and focused purpose.