

Sri Lanka Institute of Information Technology

Faculty of Computing

SE1020 - Object-Oriented Programming

Year 01 and Semester 02

Lecture 03

Noun Verb Analysis and CRC Cards

Recall...

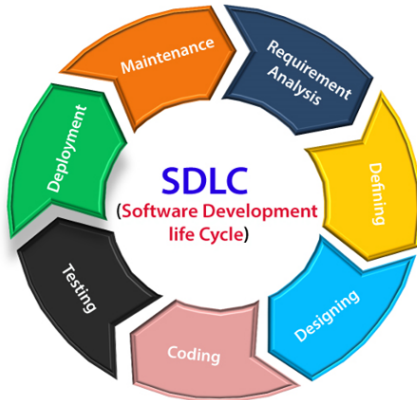
- Understand the concept of Abstraction.
- Implementing encapsulation using private attributes, getters, setters, and constructors in Java.
- Differentiate between default constructors and overloaded constructors for initializing objects.
- Implement method overloading and its use.

Learning Objectives

- Understand the Software Development Life Cycle (SDLC) in an Object-Oriented context
- Understand the importance of object-oriented analysis
- Understand the importance of using noun and verb analysis to identify classes, responsibilities and relationships.
- Understand the importance of using CRC cards in object-oriented analysis

The Software Development Life Cycle (SDLC) in an Object-Oriented context

- ① Requirements Gathering
- ② Object Oriented Analysis
- ③ Design
- ④ Implementatic
- ⑤ Testing
- ⑥ Deployment
- ⑦ Maintenance



The Software Development Life Cycle (SDLC) in an Object-Oriented context

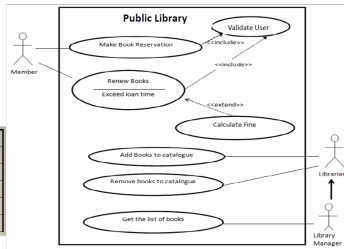
1 Requirements Gathering

Requirements should ideally be describing the new application/software that needs to be built.

Requirements are generally represented as

- Use Case Diagrams
- Use Case Scenarios or as User Stories

Number	1
Name	Borrow a book
Summary	User borrows a specific book
Preconditions	User has access to the system
Postconditions	User gets the confirmation
Primary Actor(s)	Library User



2. Object Oriented Analysis

In Object Oriented Analysis we take the requirements captured as above and;

Identifying Objects

What are the key things your software needs to deal with? Think about the real-world entities that the system represents.

For example, in a library system, objects might be books, members, librarians, etc.

Defining Classes

What are the common characteristics of these objects? A class is like a blueprint for creating objects. It defines the types of data (attributes) and actions (methods) that objects of that class will have.

For example, the "Book" class might have attributes like title, author and methods like "borrow", "return" etc.

Describing Relationships

How do these objects interact with each other? Objects don't exist in isolation. They interact with each other to perform tasks.

For example, a "Member" object might "borrow" a "Book" object. OOA identifies these relationships.

3 Design

This is where object-oriented design shines. It involves:

- Class Diagrams: Defining classes, their relationships and their methods.
- Object Diagrams
- Sequence Diagrams
- State Diagrams
- Component Diagrams
- Deployment Diagrams etc.

Key Principles Applied: Encapsulation, inheritance, polymorphism, abstraction are all considered during the design phase to create a modular, reusable, and maintainable system.

Note : Within OOP module we focus on how to draw a class diagram.

4 Implementation (Coding)

Translating the design into code using an object-oriented programming language (Java, C++, Python, etc.). Classes are implemented, objects are created, and methods are defined. The code directly reflects the classes and relationships defined in the design.

4 Testing

Testing focuses on objects and their interactions. Unit tests are written for individual classes and methods. Integration tests verify that objects work together correctly. Special attention is paid to testing polymorphism and inheritance hierarchies. Test-driven development (TDD) is often used, where tests are written before the code.

4 Deployment:

Deploying the system, which may involve configuring the runtime environment and setting up the necessary infrastructure. Deployment diagrams from the design phase guide this process.

4 Maintenance:

Maintaining and updating the system. The modularity and reusability of object-oriented code make maintenance easier. Changes to one class are less likely to affect other parts of the system.

More about Object Oriented Analysis

In order to identify classes, objects and relationships in OOA, the following techniques are used.

- ① Noun/Verb Analysis
- ② CRC Method

Noun / Verb Analysis

Noun / verb analysis is a technique used in software development to help **identify** potential **classes** and their **responsibilities** within a system. It involves examining the textual description of the system's requirements or use cases and extracting the nouns and verbs to gain insights into the system's structure and behavior.

Here are the steps:

- 1 Identify objects in our problem statement by looking for **nouns** and **noun phrases**.
- 2 Each of these **can be underlined and becomes a candidate** for an object in our solution.
- 3 We can eliminate some nouns by applying five simple rules.

Note:

*Common Nouns – Correspond to **Classes***

*A common noun (e.g. **Person**) is a name of beings or things.*

*Proper Nouns – Correspond to **Objects***

*A name used for an individual person, place, or organization, spelled with an initial capital letter (e.g. **Pushpamala**, **IT**, **CSSE** and **SLIIT**)*

Activity 01

Library System – A Description

- In the library, a member can borrow and return books
- The people using the library can also search for books.
- The users of the library from the Faculty of Computing belong to the Department of IT/CSSE/CSE.
- Each book has an ISBN.
- The SLIIT librarian is Ms. Pushpamala Perera.

Library System (Nouns in blue)

- In the **library**, a **member** can borrow and return **books**.
- The **people** using the **library** can also search for **books**.
- The **users** of the **library** from the **Faculty of Computing** belong to the **Department of IT/CSSE/CSE**.
- Each **book** has an **ISBN**.
- The **SLIIT librarian** is **Ms. Pushpamala Perera**.

Rules for rejecting nouns

Think about the library system,

- 1 Redundant - In the library system, **member** and **user** refers the same person. So we keep one noun and eliminate the other noun(s)
- 2 An event or an operation – **Search book** is an operation of the library system.
- 3 Outside scope of system – **Library, Faculty, Department (CSSE/IT/CSE), Librarian** is outside scope of library.
- 4 Meta-language (Meta language is words or symbols for talking about language itself.) - In a library system, **people** who are using library system can call as member.
- 5 An attribute – **ISBN** of a book is an attribute.

Remaining nouns are identified as classes through noun/verb analysis

- Member
- Book

Activity 02

System of DVD rental store – A Description

- In a DVD rental store, there are two types of users. A registered member can borrow up to three DVDs at a time. These members have already paid a deposit and only need to pay 50/= per DVD.
- Unregistered members can also borrow DVDs at the rate of 75/= per DVD. They are required to provide their id card for this purpose.
- Members can keep the DVD for three days and when they are returned appropriate fines may be calculated.

System of DVD rental store (Nouns in blue)

- In a DVD rental store, there are two types of users. A registered member can borrow up to three DVDs at a time. These members have already paid a deposit and only need to pay 50/= per DVD.
- Unregistered members can also borrow DVDs at the rate of 75/= per DVD. They are required to provide their id card for this purpose.
- Members can keep the DVD for three days and when they are returned appropriate fines may be calculated.

List of the nouns

- DVD Class
- User / Member Redundant
- Unregistered member Class
- Registered member Class
- ID card ID number is an attribute
- Fine Attribute
- Deposit Attribute / Operation
- DVD rental store Outside of scope
- Rate Attribute

Identified classes through noun/verb analysis

- DVD
- Registered member
- Unregistered member

Activity 03

Online order system – A Description

- A customer in an online store needs to register providing details such as name, address and contact number.
- The online store administrator can add new items to the store, restock (increase quantity), generate a list of items that need to be restocked.
- A customer can place an order from an online store. An order consists of multiple items.
- The customer can see the status of the orders placed and get a list of previous orders made.
- The customer specifies a payment method (credit card, debit card, paypal) for each order.
- Once the customer confirms the order and the payment is validated, the order is placed and items are updated.

Online order system (Nouns in blue)

- A **customer** in an online **store** needs to register providing details such as **name**, **address** and **contact number**.
- The online **store administrator** can add new **items** to the store, restock (increase quantity), generate a **list of items** that need to be restocked.
- A customer can place an **order** from an online store. An order consists of multiple items.
- The customer can see the **status** of the orders placed and get a **list of previous orders** made.
- The customer specifies a **payment** method (**credit card**, **debit card**, **paypal**) for each order.
- Once the customer confirms the order and the payment is validated, the order is placed and items are updated.

List of the nouns

- List of items Redundant
- Paypal, credit card, debit card Redundant
- List of previous orders Redundant
- Item Class
- Payment Class
- Order Class
- Administrator Outside Scope of System
(The administrator is actually a user of the system [An Actor])
- Store Outside the scope
- Customer Class
- Name, address, status, contact number Attributes

Identified classes : Item, Payment, Order, Customer

CRC Cards

CRC cards are a simple, yet powerful tool used in object-oriented analysis to understand and define the behavior of classes. CRC stands for **Class**, **Responsibility**, and **Collaboration**, which are the three key pieces of information captured on each card.

- **Class:** This is the name of the class, representing a group of similar objects in your system. For example, in a library system, "Book" could be a class.
- **Responsibilities:** These are the things a class is supposed to do. What actions can it perform? For the "Book" class, responsibilities might include "add book details", "borrow book", "return book".
- **Collaborators:** These are the other classes that , a class needs to interact with to fulfill its responsibilities. A "Book" might collaborate with a "Member" class when it's borrowed.

A CRC Card

- A CRC card is a 4×6 inches card divided into three sections.
- The **name of the class** on the top of the card.
- The **responsibilities** of the class on the left of the card.
- The class **collaborations** on the right of the card.

Class name:	
Responsibilities:	Collaborations:

Identifying Responsibilities

- To identify the responsibilities of classes during Noun-Verb Analysis, we carefully examine the verbs used in the problem description.
- These verbs describe the actions, tasks, or behaviors that occur within the system and help us understand what each class should be responsible for.
- Once we have identified the nouns as possible classes, the verbs connected to these nouns guide us in determining the methods or operations those classes need to perform.

Note:

- The list of responsibilities on the CRC card is *not the list of methods* in the class.
- A responsibility may be realized by several methods.

Example

An *Item* class in a Supermarket may have the responsibility of "*Manage details of an Item*"

We may need the following methods to realize the above responsibility.

addItem()

updateItems()

deleteItems()

Activity 04

System of DVD rental store – A Description

- In a DVD rental store, there are two types of users. A registered member can borrow up to three DVDs at a time. These members have already paid a deposit and only need to pay 50/= per DVD.
- Unregistered members can also borrow DVDs at the rate of 75/= per DVD. They are required to provide their id card for this purpose.
- Members can keep the DVD for three days and when they are returned appropriate fines may be calculated.

CRC Cards

DVD Class	
Responsibilities	Collaborations
Store details of DVDs	
Borrow DVDs	Registered Member , Unregistered Member
Return DVDs	Registered Member , Unregistered Member

Registered Member Class	
Responsibilities	Collaborations
Store details of the member	
Calculate payment	
Calculate fine	

Unregistered Member Class	
Responsibilities	Collaborations
Provide ID number	
Calculate payment	
Calculate fine	

Analysis Class

In object-oriented analysis (OOA), an analysis class is a representation of a key concept or entity within the system you're trying to model.

It's a way to capture and describe the essential characteristics and behaviors of these important elements of the problem domain.

In short, an **Analysis Class** focuses on "what" an entity does" rather than "how it will be implemented"

Example

You're building a system for a library. In the real world, there are things like books and members.

Analysis class: In your OOA, you'd create analysis classes to represent these concepts. You might have a "Book" class and a "Member" class.

Types of Analysis Classes

1 Entity Classes:

These classes represent the core data or information that the system manages. These are the classes that we have considered up to now.

Examples:

In a library system, "Book" and "Member" would be entity classes.

In an e-commerce system, "Product" , "Customer" and "Order" are good examples.

2 Boundary Classes:

These classes handle the interaction between the system and the outside world (users, other systems, devices). They act as an interface.

Examples:

In a library system, "LoginForm" and "PrintReport" could be boundary classes.

In an e-commerce system, "ShoppingCartPage," "PaymentGateway," and "ShippingNotification" are examples.

Control Classes:

These classes manage the flow of activity within the system. They coordinate how other classes interact and often implement the logic of use cases or business rules.

Examples:

In a library system, "ReservationHandler," and "OverdueNoticeGenerator" could be control classes.

In an e-commerce system, "InventoryManager," and "PaymentVerifier" are examples.

They often contain logic that determines the sequence of events, handles transactions, and enforces rules. They don't usually store data themselves but rely on entity classes for that.

Note

Actors as Entity Classes

- *Actors are users of the system in a use case diagram.*
- *If the actors only have login credentials in the system, they will typically not be classes. e.g. user accounts, administrator.*
- *However, if the actor needs to do provide his/her data and that is directly relevant to the application domain then such actors would be classes.*
- *e.g. A customer in an online Store, has to register and provide details of shipping address, billing address, contact details would be a class.*

Activity 05

Online order system – A Description

- A customer in an online store needs to register providing details such as name, address and contact number.
- The online store administrator can add new items to the store, restock (increase quantity), generate a list of items that need to be restocked.
- A customer can place an order from an online store. An order consists of multiple items.
- The customer can see the status of the orders placed and get a list of previous orders made.
- The customer specifies a payment method (credit card, debit card, paypal) for each order.
- Once the customer confirms the order and the payment is validated, the order is placed and items are updated.

Identified classes : Item, Payment, Order, Customer

CRC Cards

Customer Class	
Responsibilities	Collaborations
Register	

Item Class	
Responsibilities	Collaborations
Add item details	
Restock items	
Generate a list of items	

Payment Class	
Responsibilities	Collaborations
Add payment details	Order, Customer
Calculate payment	Order
Validate payment	Order, Customer
Add details of payment method	Customer

Order Class	
Responsibilities	Collaborations
Add order details	Customer, Item
Show order status	
Confirm order	
Generate a list of previous orders	Customer

Note

Handling list of objects

Since an entity class is typically responsible for handling one object at a time.

Generating a list of previous orders should not be in the order class.

Similarly, generating a list of items to be reordered should not be in the Item Class.

Refine CRC Cards

Customer Class	
Responsibilities	Collaborations
Register	

Item Class	
Responsibilities	Collaborations
Add item details	
Restock items	

Report Class	
Responsibilities	Collaborations
Generate a list of items	Item
Generate a list of previous orders	Order, Customer

Payment Class	
Responsibilities	Collaborations
Add payment details	Order, Customer
Calculate payment	Order
Validate payment	Order, Customer
Add details of payment method	Customer

Order Class	
Responsibilities	Collaborations
Add order details	Customer, Item
Show order status	
Confirm order	

Summary

- Introduction to Software Development Life Cycle (SDLC) in an Object-Oriented context
- Introduction to Object-Oriented analysis
- Using noun and verb analysis
- Using CRC cards
- Introduction to analysis classes

Thank You!