

2.1 Introduction to object-orientation

Object-orientation (OO) is a programming paradigm based on the concept of "objects", which are instances of classes. It aligns closely with real world scenarios.

1. Objects

An object is a self-contained unit that combine data (attributes or properties) and behaviour (method or function).

2. Classes

A class is a blueprint or template for creating objects.

3 Inheritance

- ↳ Inheritance allows a class (subclass) to inherit attribute and methods from another class (super class).
- ↳ It reuse's code and has hierarchical classification.

4 Encapsulation

It is the bundling of data and methods that operate on the data within a single unit or class.

5 Polymorphism

Polymorphism allows objects of different class to be treated as objects of a common superclass.

2.2 Object-oriented system development - function / data methods

The difference between object-oriented design and functional-oriented design are

Object-oriented Design

Functional-oriented design

Design based on objects representing real world entities and their interactions.	Design based on Function or procedures to perform tasks.
--	--

Achieved through class and object	Uses function to abstract action and tasks
-----------------------------------	--

Objects manages their own data through methods actions.	Data is passed between functions.
---	-----------------------------------

Bottom up Approach	Top down approaches
--------------------	---------------------

Easily scalable	Scalability can be challenging
-----------------	--------------------------------

Java Ex Java, c++, Python	Ex C, Pascal, Fortran
---------------------------	-----------------------

2.3 Object-oriented Analysis

- ↳ This phase of s/w development is concerned with determining the system requirements and identifying class and their relationship to other classes in the problem domain.
- ↳ Scenarios are a great way of examining who does what in the interaction among object and what role they play. i.e. their interrelationships.

Techniques in object-oriented analysis

1 Use case Modeling

- ↳ Identify and describe the interaction between user (actor) and the system
- ↳ Create use case diagram to visualize these interactions.

2 Class Diagram

- ↳ Represent class and their relationships
- ↳ Show attributes and methods of each class

3 Sequence Diagram

- ↳ Illustrate how objects interact over time.

4 State Diagram

- ↳ useful for modeling business processes and workflows

6 Collaboration Diagrams.

Emphasize the structural organization that sends and receive messages

Focuses on object interaction.

The analysis model aims at creating a good platform for the system design and will also form the basis of the design.

Object-oriented analysis contains, the following activities.

- Finding the objects
- Organizing the objects
- Describing how object interact
- Defining the operations of the object
- Defining the object internally

The goal is to find domain of the problem and system responsibility by knowing all user's need which is accomplished by models.

This model concentrate on describing what the system does rather than how it does it

↳ Object oriented analysis has following steps

- ↳ Identifying the actors
- ↳ Develop a simple business process using UML activity diagram

- ↳ Develop interaction diagrams
- ↳ Identify class

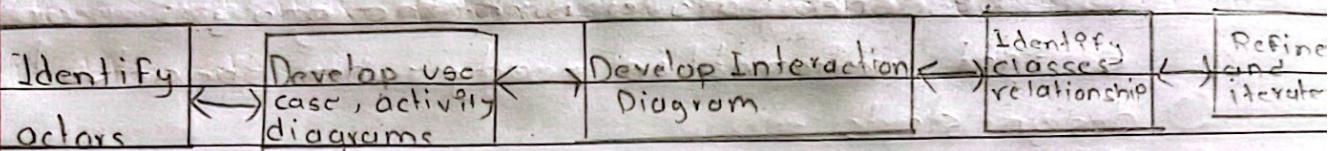


Fig 60 analysis

2.4 Object-oriented Construction and Testing

Object oriented construction means that the analysis model is designed and implemented in source code. This source code is executed in the target environment.

During the design, one must take care to follow all restrictive demands on the system.

The goal is that the object identified during the analysis should also be found within the design which is called traceability.

Construction consists of design and implementation. Construction model produces two models, the design model and the implementation models.

To develop the design models we perform the following activities

- 1) Identify the implementation environment
- 2) This step includes identifying and investigating what consequences the implementation environment will have on design. This step can be done parallel with the analysis work so that when it is ready when the actual design starts
- 3) Develop first approach to a design model
- 4) Describe how objects interact in each specific use

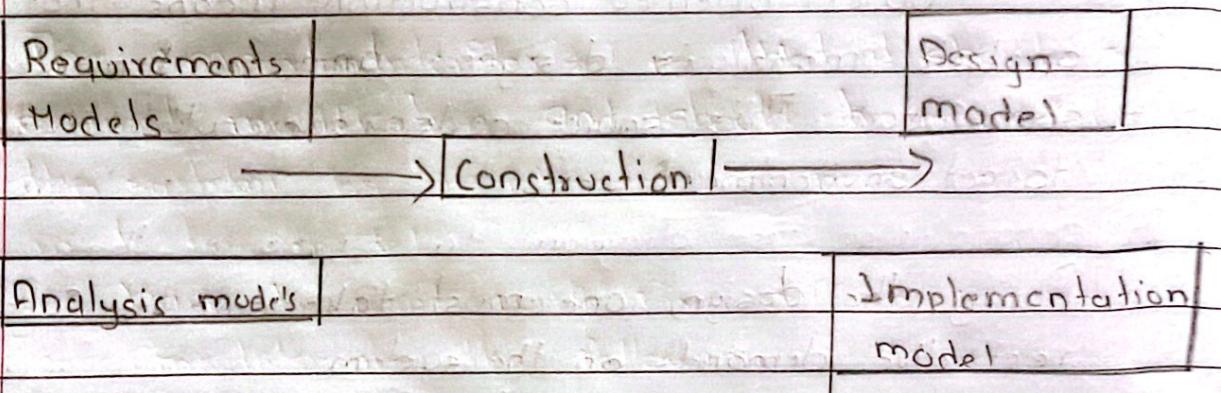


Fig:- The input and output models of construction

Object-oriented testing is methodology that focuses on testing systems designed using object oriented principles and practices. It is crucial for ensuring the reliability, maintainability and scalability of the software systems.

The testing system doesn't differ considerably from the testing system developed by other methods

The program testing begins at the lowest level with unit testing and progress to integration testing and finally testing the entire system.

Inheritance between classes can create new difficulties with testing. Testing of inheritance hierarchies requires a more exhaustive testing methods, where one is aware of how the system will appear in operation.

2.5 Object-oriented programming with examples

Object oriented programming is powerful and flexible programming paradigm that has greatly influenced on the field of software engineering. OOP is a powerful paradigm used in software engineering to design and implement real world entities as objects.

Object represent real world entities and class are blue. Examples - print for creating objects.

Example:

Class: customer

Attributes : 'ID', 'name', 'email'

Methods : 'placeOrder()', 'update profile()'

Object-oriented Program (OOP) provides a structured approaches to design software systems, making them easier to understand, maintain and extend.