PG Department of Computer Applications

I BCA

Subject : PROGRAMMING IN C++

Sub.Code : 16SCCCA2

Staff In Charge : Asst.Prof.P.Jayalakshmi
Asst.Prof.D.Arokia Mary
Asst.Prof. G.Shakunthala

Unit - V

Standard Template Library:

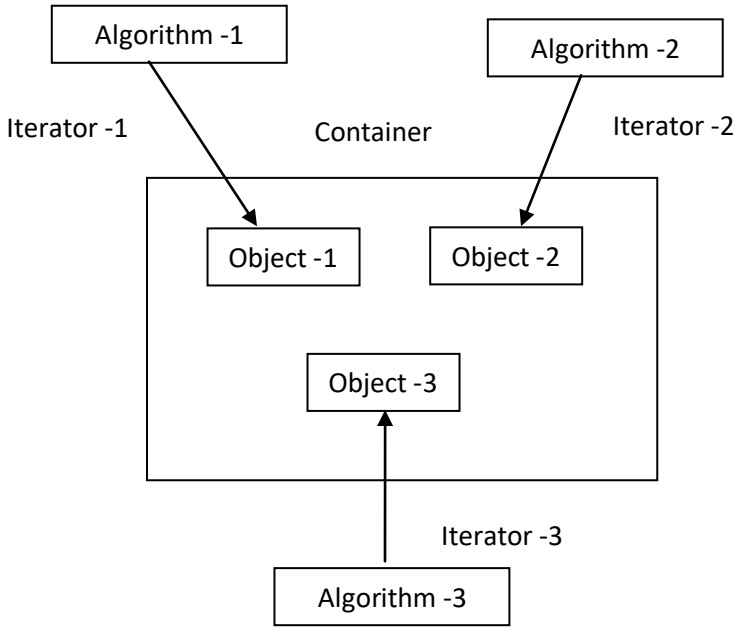
The collection of generic classes and function are called standard template library (STL).

Components of STL:

3 key components are

1. Containers
2. Algorithms
3. Iterators

These 3 components work in conjunction. They support to one another.



Containers

- Containers can object that to mark actually stored data.

- It holds different data with different data types.

Algorithm:

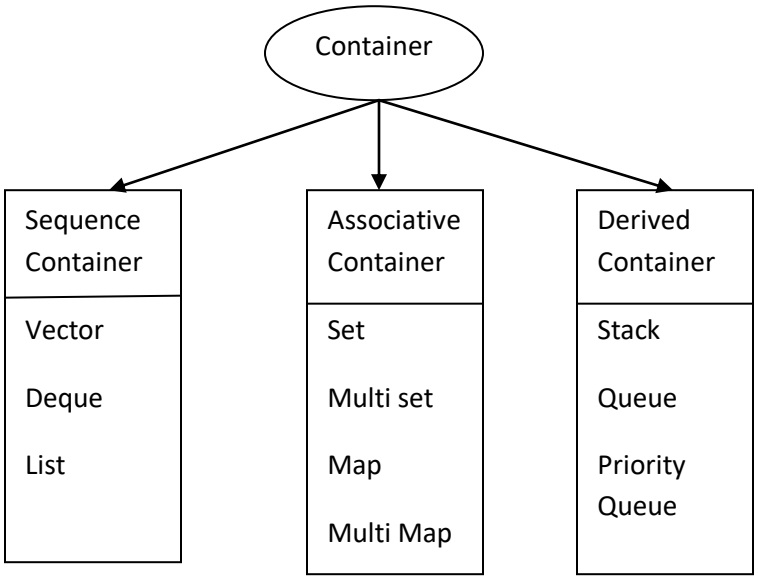
- It is a procedure that is used to process the data contained in the container.

STL Container several types of Algorithm.

1. Initializing
2. Searching
3. Copying
4. Storing
5. Merging

Containers

- It contains Data.
- STL defines ten containers which are grouped into 3 categories as follows.



Containers	Description	Header File	Iterators
Vector	It is a Dynamic Array. It permits direct access to any elements.	<vector>	Random Access
List	It is a bidirectional linear list	<list>	Bidirectional
Deque	It allows insertion / deletion at both ends.	<deque>	Random Access
Set	For storing unique set (not allowed duplicate).	<set>	Bidirectional
Multi Set	For storing unique set (allowed duplicate).	<set>	Bidirectional
Map	For storing unique set of key and values.	<map>	Bidirectional

	Each key contains only one value.		
Multi Map	Each key contains only one value	<map>	Bidirectional
Stack	Last in First Out [LIFO]	<stack>	No Iteration
Queue	Fist in First Out [FIFO]	<queue>	No Iteration
Priority Queue	The element is always the highest priority queue.	<queue>	No Iteration

Algorithm:

STL provides to standard algorithm.

Categories of Algorithm

1. Retrieve or Non – Mutating Algorithm
2. Mutating Algorithm
3. Numerical Algorithm
4. Sorting Algorithm
5. Set Algorithm
6. Relational Algorithm

1) Retrieve or Non - Mutating Algorithm	
Operator	**Description**
Count()	Count the occurrence of value True; it values are same and the 1st elements
2) Mutating Algorithm	
Copy()	Copy the sequence
Fill()	Fill a sequence with specified value
Remove()	Delete the element of specified value
Swap()	Swap the element
3) Sorting Algorithm	
Binary Search	Conducts a binary
Merge()	Search on sequence order
Sort()	Joined the 2 sorted sequence sort a sequence
4) Set Algorithm	
Set Difference()	Construct a sequence that is the difference of 2 ordered set
Set Union()	Construct a sequence that is Union of 2 ordered set

5) Relational Algorithm	
Equal()	Find whether two set are same
Max()	Given Maximum of two values
Min()	Given Minimum of two values
6) Numeric Algorithm	
Accumulate()	Accumulate the result of operation
Inner Product()	Accumulate the result of operations on a pair of sequence

Manipulating Strings: A string is a sequence of character operation

1. Creating a string object
2. Reading a string
3. Display a string
4. Finding a string
5. Modifying
6. Comparing
7. Concatenating
8. Accessing a character in a string
9. Length of a string.

Function	Task
Begin()	Returns a reference to the start of a string
Compare()	Compare two string
Empty()	Return true if the string is empty
End()	End of String
Erase()	Remove the character
Find()	Search the string
Insert()	Insert a character to the string
Length()	Count the number of element in a string
Swap()	Interchange the string
Replace()	Replace the string

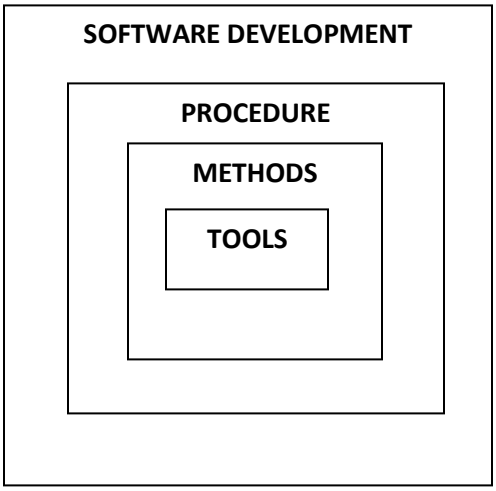
Object Oriented System Development

1. Software Engineer has been trying various tools, methods, procedure to control the process of software development.
2. They are used in all the stages of software development process namely

    a. Planning
    b. Analysis
    c. Design

d. Development
e. Maintenance

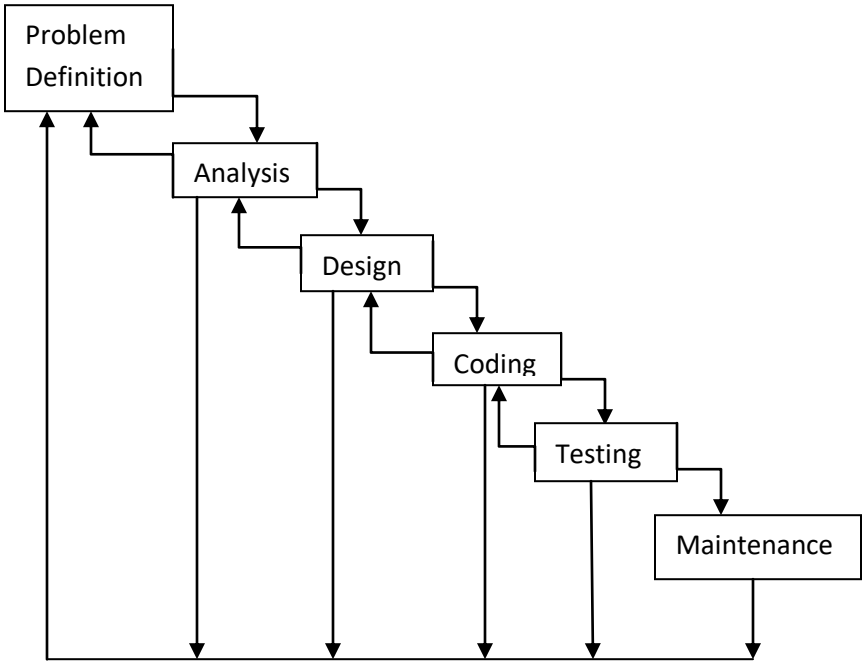
Software Development components



A successful system must include the following:

1. Satisfy the user requirement.
2. Be easy to understand by the user and operators.
3. Be easy to operate.
4. Have a good user interface.
5. Be easy to modify.
6. Be expandable.
7. Have adequate security control against misuse of data.
8. Handle the errors and exceptions satisfactorily.
9. Be delivered on schedule within the budget.

Procedure - oriented paradigms



- The Classic life cycle is based on an underlying model commonly referred to as the **"Waterfall Model"**
- The activities include problem definition requirement analysis, design, coding, testing and maintenances.

Problem definition – This activity required a precise definition of the problem in user term

Analysis – This covers a detailed study of the requirement of both the user and the software.

Design – The design phase deals with various concept of system design such as data structures, software architecture and algorithm. This phase translate the requirements into a representation of software. This stage answer the question of how.

Coding - It is the past of the system development life cycle process actually the program code has being developed.

Testing – It is an important phase of the software development process. It ensure that errors or rectified.

Maintenance: It covers the documentation section it also include the activities of improvement of the software development. It also used to improve the software and its quality.

Each phase of life cycle has its own goals and outputs.

	PHASES	**OUTPUT**
1	Problem Definition (Why)	• Problem statement sheet • Project request
2	Analysis (What)	• Requirement document • Feasibility report • Specifications document • Acceptance test criteria
3	Design (How)	• Design document • Test class design
4	Coding (How)	• Code document (Program) • Test plan • User manual
5	Testing (What & How)	• Testing code • Testing results • System manual
6	Maintenance	• Maintenance log sheet • Version document

- The Software life cycle is implemented using the functional decomposition techniques.
- It's also known as top down, modular approach.
- The techniques are based on interpretation of the problem space & translation of solution space as an inter-dependent set of function.

Procedure Oriented Development Tools

These tools and techniques provide answers to the how question of the system development.

Process	First Generation	Second Generation	Third Generation
Physical Process	System Flowcharts	Context Diagram	Inheritance graphs Object relationship chart
Data Representation	Layout forms Grid charts	Data Dictionary	Objects Object dictionary
Logical process	Play script English narratives	Decision tables and trees, data flow diagram	Inheritance graphs Data flow diagrams
Program Representation	Program flowchart I/O layout	Structure of charts Warnier / Orr diagrams	State change diagrams Ptech diagram Coad / Yourdon chart

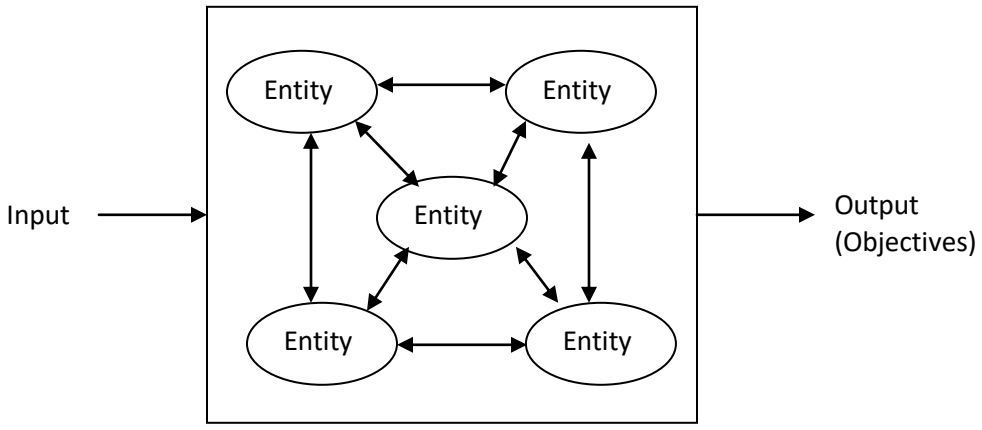
Most frequently used tools are follows:

1. System flowchart – A graphical representation input, output, data flow and key points in the system.
2. Program flowchart – A graphical representation of program logic.
3. Play script – A narrative description of executing a procedure.
4. Layout form – A format designed for putting the input data or displaying result.
5. Grid charts – Chart showing the relationship between different modules of a system.
6. Context diagrams – Outlines the system boundary.
7. Data flow diagram - It is a network representation of system which includes processes and data files.
8. Data dictionary – It contains a list of terms and their definition for all the data items and data stores.
9. Structure chart – A graphical representation of the control logic of functions (modules) representing a system.
10. Decision table - A table of contingencies for defining a problem and the actions to be taken.
11. Decision tree – A graphical representation of the condition & Outcomes that resemble the branches of a tree.
12. Warnier / Orr diagrams – A horizontal hierarchy chart using nested sets of braces, pseudo codes and logic symbols to indicate the program structures.

Object Oriented Paradigm

- Draw on general systems theory as a conceptual background
- A system can be viewed as a collection of entities that interact together to accomplish certain objectives.
- The entities are called objects.
- The object that encapsulate data and procedures.
- They play central role in all the stages of software development and high degree of overlap and iteration between the stages.

A System showing inter – relationship of entities

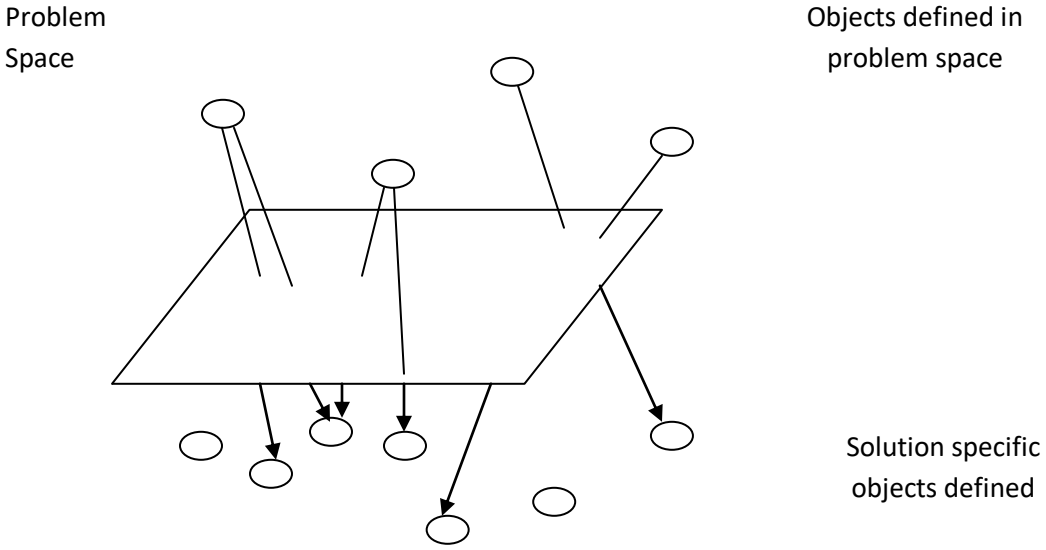


OOA - Object Oriented Analysis – refers to the methods of specifying requirement of the software in terms of real world objects their behavior and their interactions.

OOD - Object Oriented Design – turns the software requirement into specification for objects and derives class hierarchies from which objects can be created.

OOP - Object Oriented Programming – refers to implementation of program using objects.

In one phase, the problem domain objects are identified, while in the next phase additional objects required for a particular solution are specified. The design process is repeated for these implementation – level objects.



Object Oriented Notations and Graphs

Graphical notations are essential part of any design and development process, and object oriented design is no exception. Notation represents classes, objects, subclasses, and their inter – relationships. Some of the commonly used notations to represent the following:

1. Classes and objects.
2. Instances of objects.
3. Message communication between objects.
4. Inheritance relationship.

5. Classification relationship.
6. Composition relationship.
7. Hierarchical chart.
8. Client server relationship
9. Process layering.

Fig 1: Various forms of representation of classes/ objects

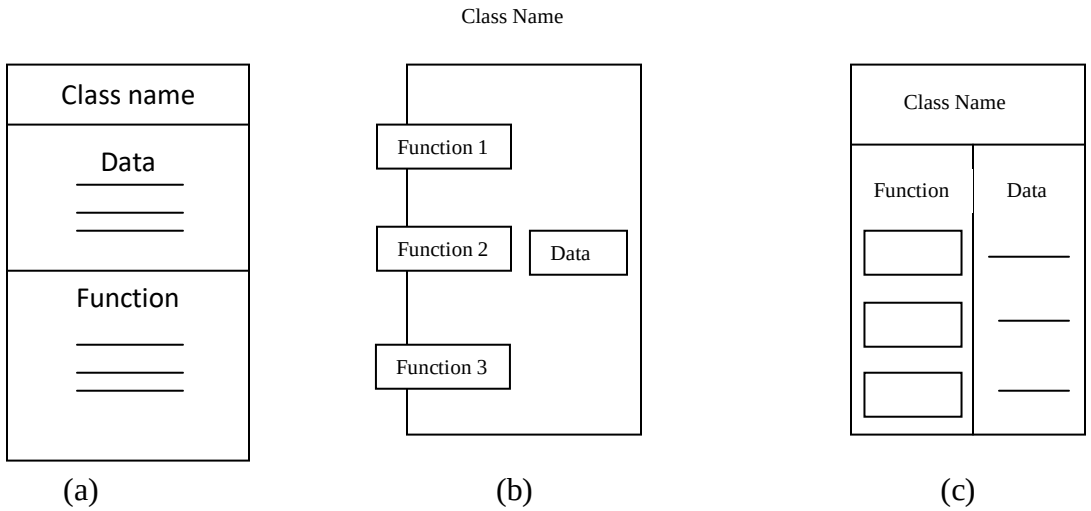


Fig 2: Instances of Objects

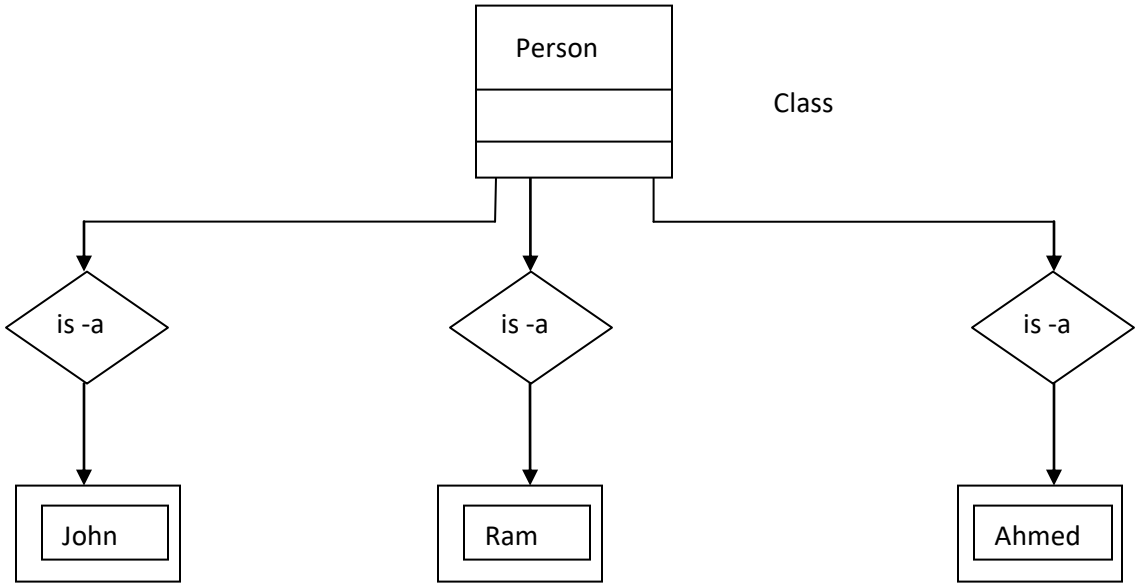


Fig 3: Message communication between Objects

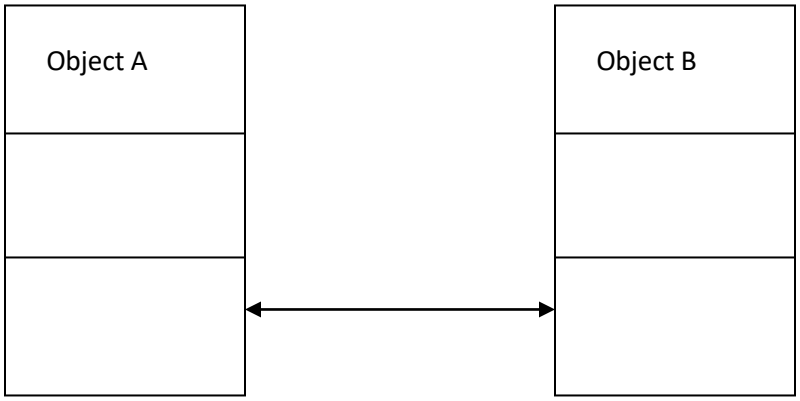


Fig 4: Inheritance relationship

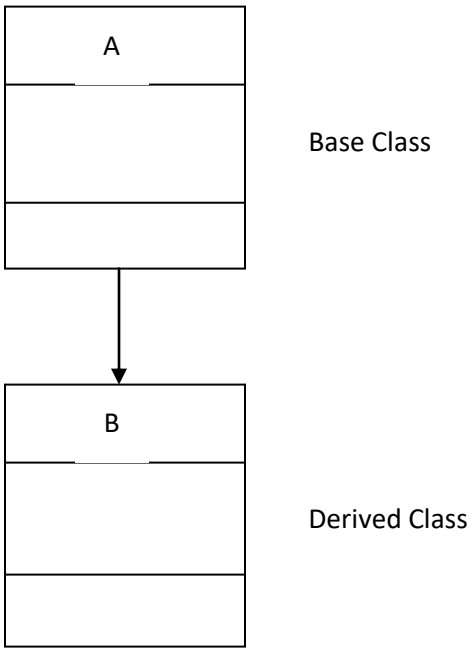
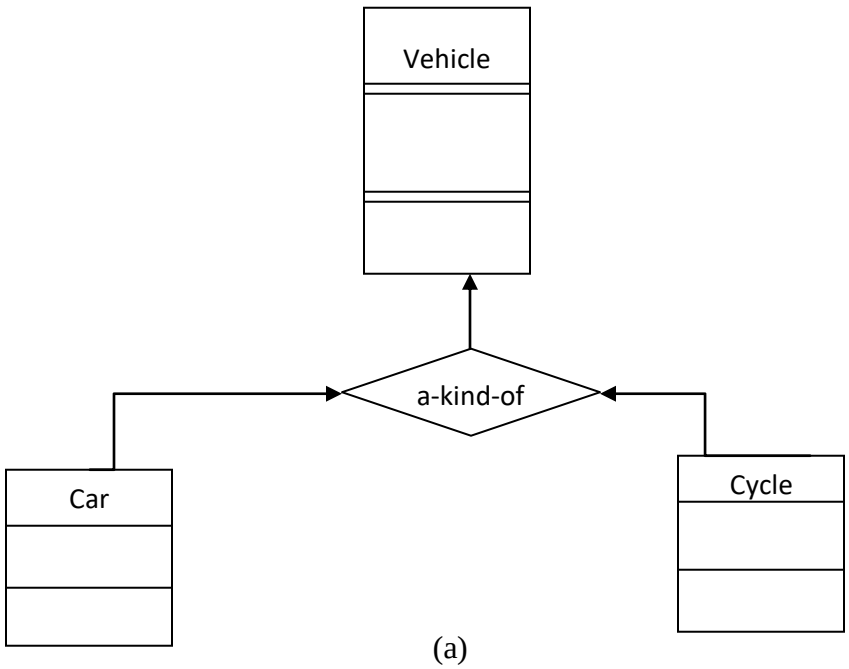
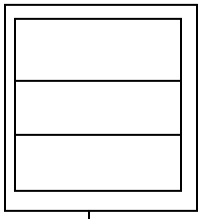


Fig 5: Classification Relationship

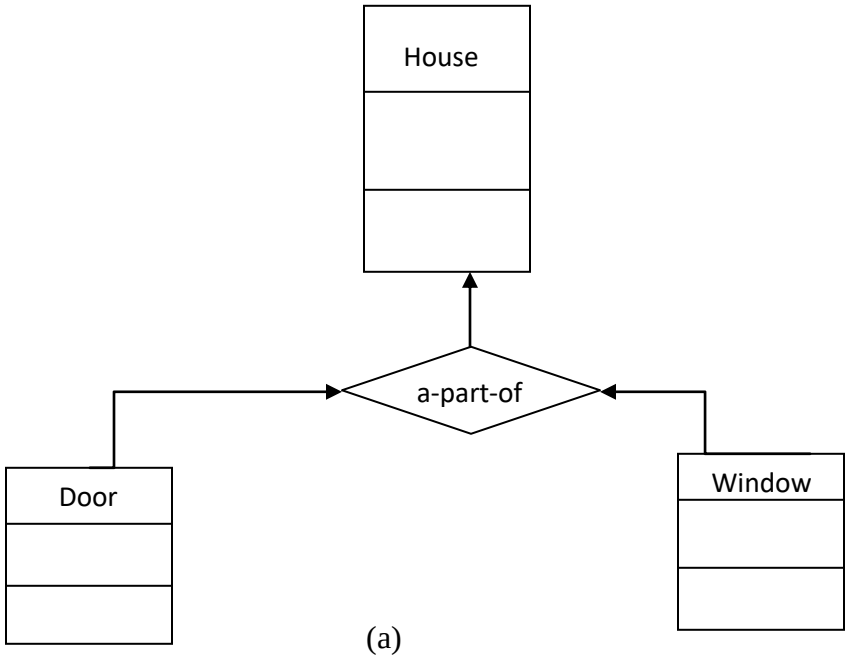


(a)

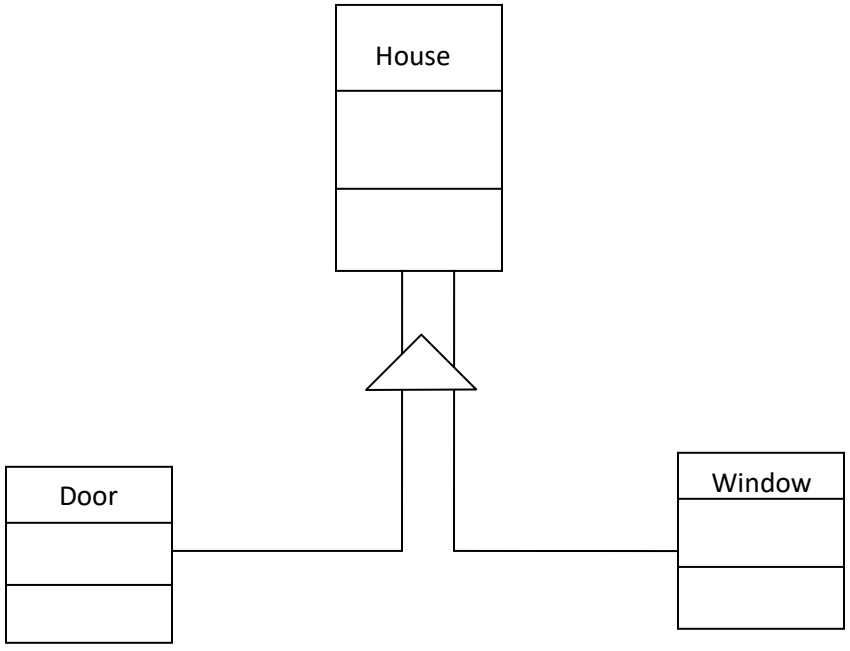


(b)

Fig 6: Composition Relationship



(a)



(b)

Fig 7: Hierarchical chart

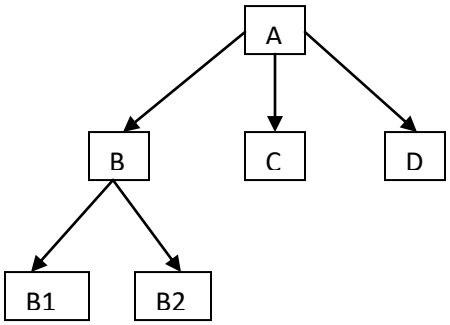


Fig 8: Client Server relationship

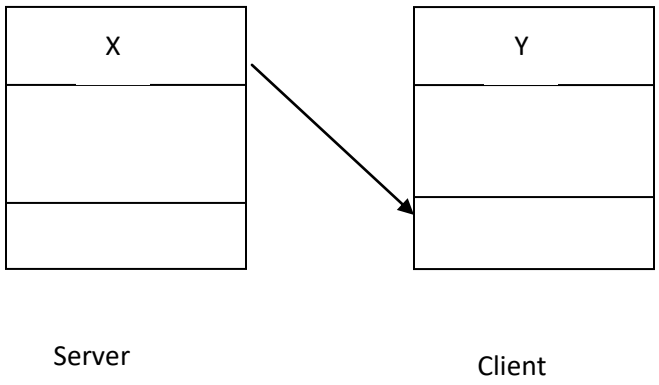
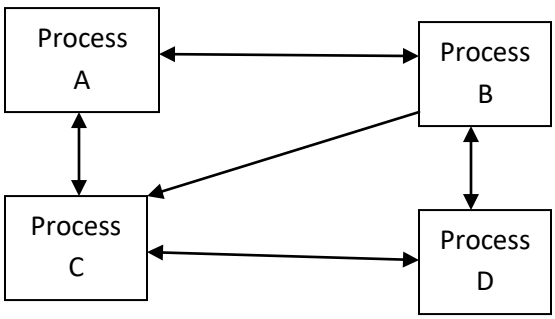


Fig 9: Process layering (A process may have typically five to seven objects)



The object oriented analysis (OOA) approach consists of the following steps:

1. Defining the problem.
2. Estimating requirements of the user and the software.
3. Identifying the objects and their attributes.
4. Identifying the interface services that each object is supposed to provide.
5. Establishing interconnections between the objects in terms of services required and services rendered.

The object oriented design (OOD) approach consists of the following steps:

1. Review of objects created in the analysis phase.
2. Specification of class dependencies.
3. Organization of class hierarchies.
4. Design of classes.
5. Design of member functions.
6. Design of driver program.

One way of understanding the system design and its ramifications before a complete system is built is to build and test a working model of the proposed system. This model system is called the prototype, and the process is called prototyping.

The benefits of using the prototype approach are:

1. You can produce understandable specification which is correct and complete as far as possible.
2. The user can understand what is being offered.
3. Maintenance changes that are required when a system is installed are minimized.
4. Development engineers can work from a set of specifications, which have been tested and approved.
