

✓ Object Oriented Programming Paradigm

The major motivating factor in the invention of object oriented approach is to remove some of the flaws encountered in the procedural approach. OPP treats data as a critical element in the program development and does not allow it to flow freely around the system. It ties data more closely to the functions that operate on it, and protects it from accidents modifications from outside functions. OOP allows decomposition of a problem into a number of entities called objects and then builds data and functions around these objects. The organization of data and functions is object oriented programs is showed below :

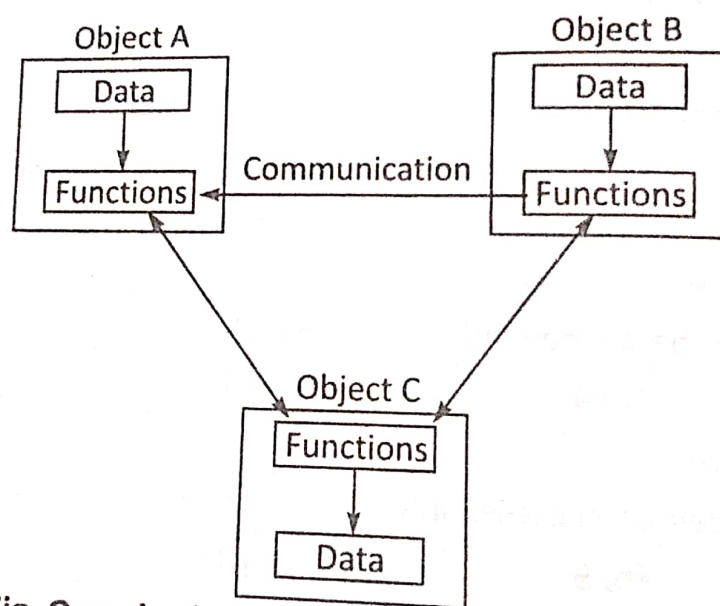


Fig. Organization of data and functions in OOP.

✓ Features of Object Oriented Programming

- (1) Emphasis is on data rather than procedure.
- (2) Programs are divided into what are known as objects.
- (3) Data structure are designed such that they characterize the objects.
- (4) Functions that operate on the data of an object are tied together in the data structure.

- (5) Data is hidden and cannot be accessed by external functions.
- (6) Object may communicate with each other through functions.
- (7) New data and functions can be easily added when ever necessary.
- (8) Follows bottom up approach in program design.

Advantages of Object Oriented Programming

- (1) Through inheritance, we can eliminate redundant code and extend the use of existing classes.
- (2) We can build programs from the standard working modules that communicate with one another, rather than having to start writing the code from scratch. This leads to saving of development time and higher productivity.
- (3) The principle of data hiding helps the programmer to build secure programs that cannot be invaded by code in other parts of the program.
- (4) It is possible to have multiple instances of an object to co-exist without any interference.
- (5) It is possible to map objects in the problem domain to those in the program.
- (6) It is easy to partition the work in a project based on objects.
- (7) Object oriented systems can be easily upgraded from small to large system.
- (8) Software complexity can be easily managed.
- (9) Message passing techniques for communication between objects makes the interface descriptions with external systems much simpler.

Applications of OOP

The following are the various applications of OOP.

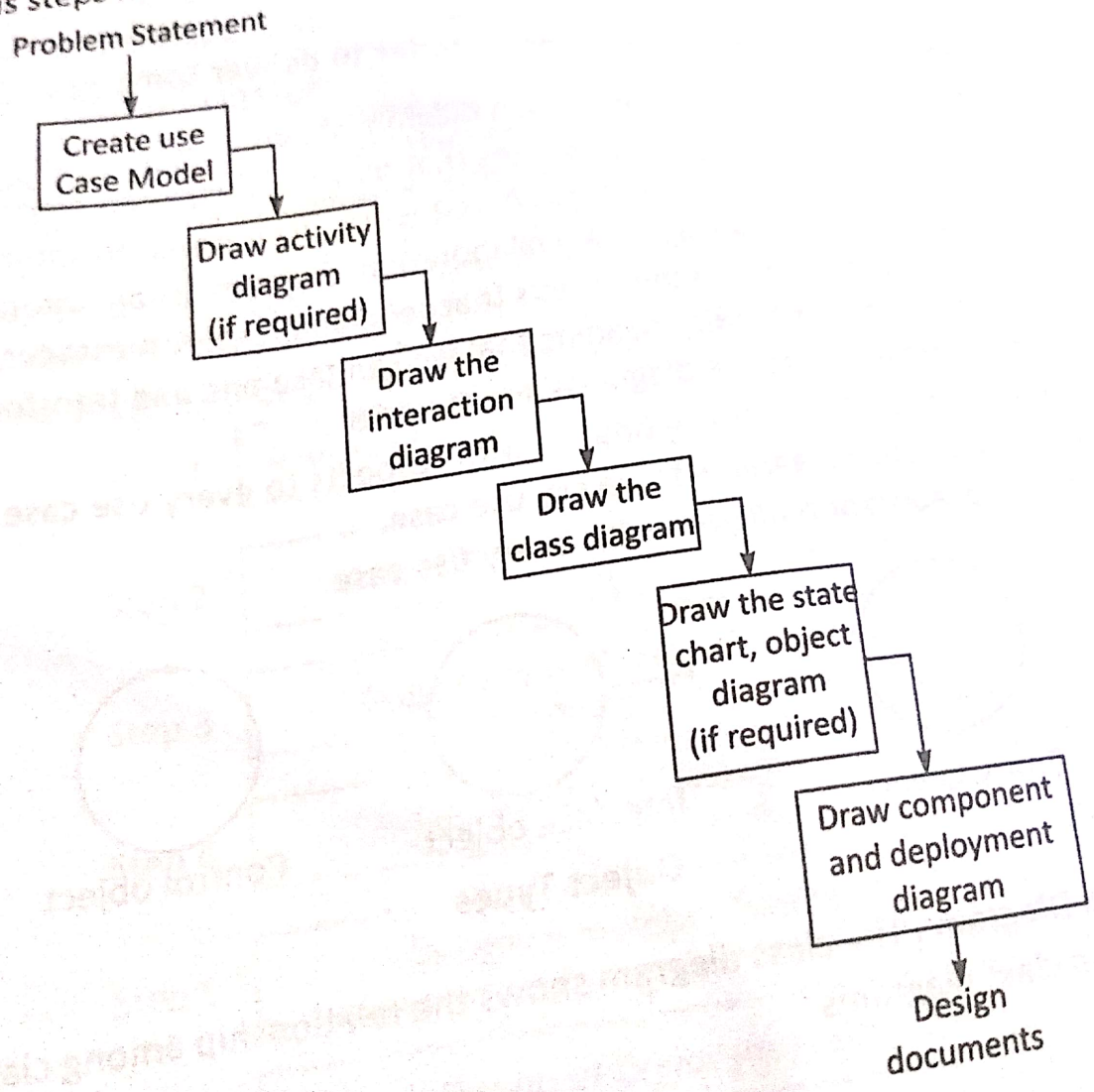
- (a) Real time systems
- (b) Simulation and modeling
- (c) Object oriented database
- (d) Hypertext, Hypermedia and Expertext
- (e) Artificial Intelligence
- (f) Expert System
- (g) Neural Networks and Parallel Programming
- (h) Decision Support Systems
- (i) Office Automation Systems
- (j) CIM/CAM/CAD Systems.

✓ Object Oriented Design

Object Oriented design begins with an examination of the real world 'things' that are part of the problem to be solved. These things (which we can call objects) are characterized individually in terms of their attributes (transient state information) and behaviour (functional process information). Each object maintains its state, and offers a set of services to other objects. Shared data areas are eliminated and objects communicate by message passing. Objects are independent entities that may readily be changed because all state and representation information is held with in the object itself.

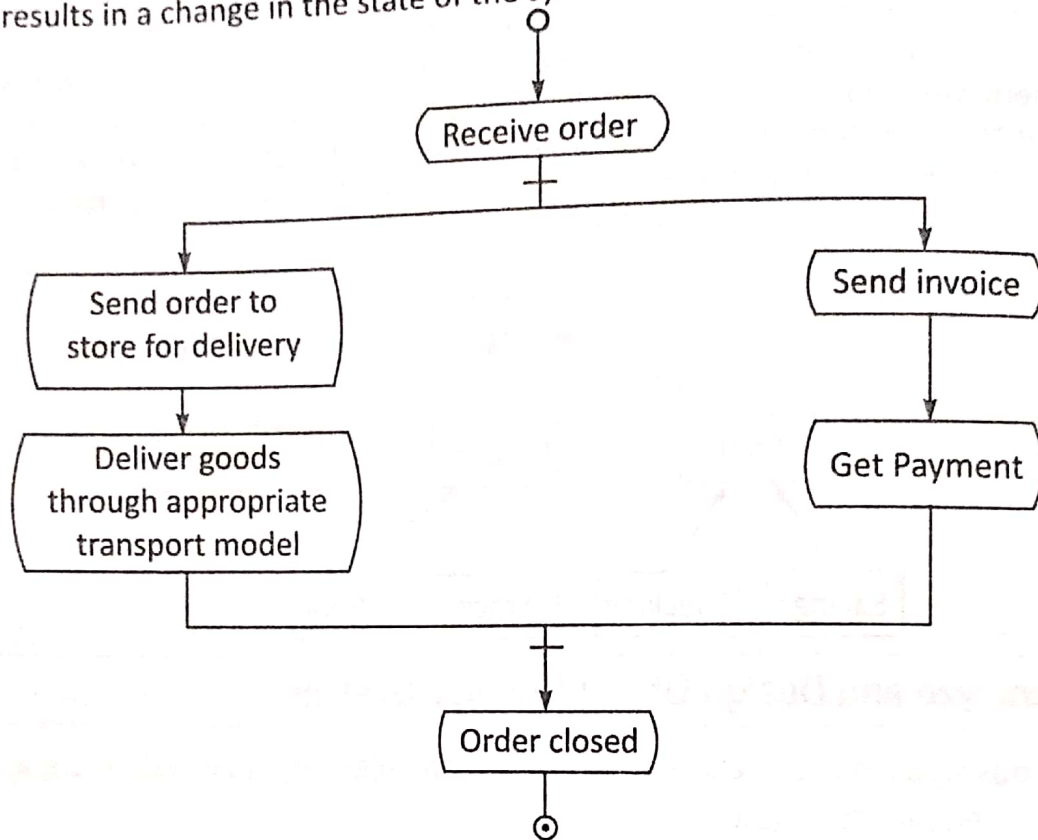
✓ Steps and Analyze and Design Object Oriented System

There are various steps in the analysis and design of an object oriented system are as follows :



(1) **Create use Case Model** : First step is to identify the actors interacting with the system. We should then write the use case and draw the use case diagram.

(2) **Draw Activity Diagram (If required)** : Activity diagram illustrate the dynamic nature of a system by modeling the flow of control from activity to activity. An activity represents an operation on some class in the system that results in a change in the state of the system. It is essentially like a flow chart.

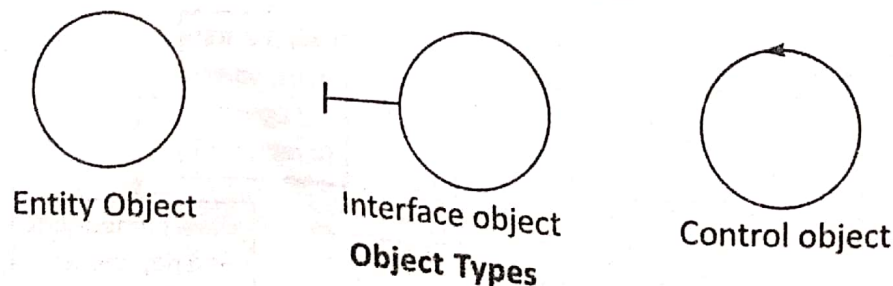


Activity diagram of processing an order to deliver some goods.

(3) **Draw the Interaction Diagram** : An interaction diagram shows an interaction, consisting of a set of objects and their relationships, including the messages that may be dispatched among them. Interaction diagrams address the dynamic view of a system. A sequence diagram is an interaction diagram that emphasizes the time ordering of messages, a collaboration diagram is an interaction diagram that emphasizes the structural organisations of the objects that send and receive messages. Sequence diagrams and collaboration diagrams are isomorphic, meaning that we can take one and transform it into the other.

The various steps to draw interaction diagrams are as under :

- (1) Firstly, we should identify that the objects with respects to every use case.
- (2) We draw the sequence diagrams for every use case.
- (3) We draw the collaboration diagrams for every use case.



(4) **Draw the Class Diagram** : The class diagram shows the relationship among classes. There are four types of relationships in class diagrams.