OOP : how it differs from structured programming

1. Compare & contrast between features of Structured / Procedure Oriented Programming with Object Oriented Programming :

   Basic features of s Structured / Procedure Oriented Programming are

   i. Emphasis on doing algorithms.
   ii. Large Programs are divided into smaller programs known as Functions
   iii. Most of the function shares global data
   iv. Data move around the system globally from function to function.
   v. Function transfers the data from one form to another.
   vi. Employs top-down approach of Programming.

   Object oriented Programming is an approach that provides a way of modularizing the program by creating partitioned memory area for both data & function that access data for creating copies of such modules in demand.

   An object is considered to be a partitioned area in memory that stores data & set of operations that can access the data. Since the memory partitions are independent, the object can be used in variety of different program without modifications.

   The data of an object can be accessed only by the functions associated with that object. However functions of one object can access the function of other object & even some normal functions can access object data provided these normal functions must be qualified enough (friend).

   The striking features of Object Oriented Programming are

   i. Emphasis on data rather than the procedure.
   ii. Programs are divided into objects.
   iii. Functions that operate on the data of an object are tied together In the data organization.
   iv. Data are hidden and can't be accessed by external functions (ordinarily)
   v. Object may communicate with each other through functions.
   vi. New data & functions can be easily added whenever necessary (concept of reusability, inheritance).
   vii. Followed bottom-up approach in programming.

CLASSES

Object contain data & code to manipulate data. The entire set of data & code of an object can be made user defined data-type with the help of a class. Actually object are variables of the type class. Once a class has been identified we can create any number of objects belonging to that class . Each object is associated with the data-type Class with reference to which they are created.

A class is thus a collection of objects of similar type. Classes are user defined data-type and behave like a built –in type in PL. e.g.

If PL is defined as a class then

PL C, COBOL,PASCAL;

will create objects C, COBOL,PASCAL belonging to the class PL.

OBJECTS

Objects are the basic run-time entities in an object oriented system. They may represent various kind of items depending upon problem aspect. e.g. *employee* or *salary* in a pay-roll system or *student* or *examiner* in an Examination system. Programming problem is analyzed in terms of object & Nature of communication between them. Program objects should be chosen In such a way that they represent closely the real-world entity. Each object contain data & code to manipulate data.

Relating objects to other paradigm

There are five main kind of programming style expressing which we can relate them to other paradigm.

Procedure Oriented Programming	algorithms.
Object Oriented Programming	classes & objects.
Logic Oriented	predicate calculus.
Constraint oriented	invariant relationship.

2. Comparison & contrast between object & classes:

1. Class is a concept, while object is physical representation of class.
2. Class is user-defined data-type. Objects are the variable of the type Class.
3. Class definition doesn't require memory space. Space is allocated during declaration of objects.

DATA ABSTRACTION & ENCAPSULATION

The wrapping- of data into a single unit is called class is known as encapsulation. The data aren't accessible to the outside world and only those function which are wrapped in the class can access it. These function provides an interface between objects data & the program.

These insulation of data from direct access by the program is called Data hiding / Information hiding.

Abstraction refers to the act of representing essential features without including the background details or explanations. Classes use the concept of abstraction & are defined as a list of abstract attributes. They encapsulate all the essential features of the object that are to be created. The attributes are called data members for holding information.

INHERITANCE

Inheritance is the property by which object of one class acquire the properties of another class. It supports the concept of hierarchical classification. The basic concept is that derived class shares common characteristics with the class from which it is derived.

It provides the idea of reusability. This means that we can add additional features to an existing class without modifying it. This is possible by deriving a class from the existing class which will have combined features.

POLYMORPHISM

Polymorphism means ability to take more than one form. E.g. an operation may exhibit different behavior in different instances. The behavior depends upon the type of the data used in the operation. Polymorphism plays an important role in allowing objects having different internal structures to share the same external interface. This means that a general class of operation may be accessed in the same manner even though specific action associated with each operation may differ.

DYNAMIC BINDING

Binding refers to the linking of a procedure call to the code to be executed in response to the call. Code associated with a given procedure call is not known until all the time of the call at run time i.e. why it is called late binding.

MESSAGE COMMUNICATION

Objects communicate with one another by sending & receiving information. A message for an object is a request for execution of a procedure and therefore will invoke a function in the receiving object that generate the desired result. Message passing involves specifying object name, function name & the information to be sent.