**Course: Advanced Bioinformatics**

**Module title: Object Oriented Data Modeling**

**Module no. : 13**

Object-Oriented Data Modeling (OODM) is state of the art in data modeling these days. Basic idea behind OODM is objects and classes. OODM centers on objects and classes. Objects are encapsulated where data and behavior are put together in the same object. Another interesting feature is Inheritance, which depicts relationship between classes and objects.

OODM has several advantages as compared to other design methodologies.

* Solve challenging problems quite easily and smoothly.
* Improved communication between users, analysts, designers, programmers
* Increased consistency in analysis, design and programs
* Explicit representation of commonality among system components
* System robustness
* Reusability of analysis, design and programming results

**OODM- Concepts**

**Class:** An entity that has a well-defined role in the application domain, state, behavior and identity.

It has several properties which help us to better understand classes.

* Tangible: person, place or thing
* Concept or Event: department, performance, registration, cell division
* Process: user interface, controller, scheduler

**Object:** A particular instance of a class. Objects exhibit behavior as well as attributes

They are different from entities as they have state and behavior.

State: attribute types and values

Behavior: how an object acts and reacts. Behavior is expressed through operations that can be performed on it

**Identity:** Every object has unique identity, even If all of its attribute values are the same

Inheritance is one of the main concepts in OODM since it provides relationship between objects.

Concepts of subclass and superclass are necessary for inheritance. Superclass is the main class from where subclass inherits information. For example vehicle is a superclass, with car, bike, truck are subclasses.

Common attributes, relationships, and operations can be inherited from one class to the other class.

**Abstract Class:** A class which has no direct instances, i.e. which cannot have any objects such as vehicle class. But subclasses may have direct instances such as car, bike etc.

**Concrete Class:** A class which can have direct instances