



Object Oriented Programming Paradigm

M.Sc. 2nd Semester

MPHYCC-5: Modelling and Simulation

Unit I (Part 1)

Dr. Ashok Kumar Jha

Assistant Professor

Department of Physics, Patna University

Mob:7903067108, Email: ashok.jha1984@gmail.com

Lecture Outline

- Introduction to Object Oriented Programming
- Its features
- Its basic concepts

Object Oriented Programming

2

- Object oriented Programming is defined as an approach that provides a way of modularizing programs by creating partitioned memory area for both data and functions that can be used as templates for creating copies of such modules on demand.
- Writing object-oriented programs involves creating classes, creating objects from those classes, and creating applications, which are stand-alone executable programs that use those objects.
- After being created, classes can be reused over and over again to develop new programs.

Features of object oriented programming

3

- Emphasis is on data rather than procedure.
- Programs are divided into what are known as objects.
- Data structures are designed such that they characterize the objects.
- Functions that operate on the data of an object are tied together in the data structure.
- Data is hidden and cannot be accessed by external function.
- Objects may communicate with each other through function.
- New data and functions can be easily added whenever necessary.
- Follows bottom up approach in program design.

Basic Concepts of Object Oriented Programming

4

It is necessary to understand some of the concepts used extensively in object-oriented programming. These include:

- Objects
- Classes
- Data abstraction and encapsulation
- Inheritance
- Polymorphism
- Dynamic binding
- Message passing

Objects

5

Objects are the basic run time entities in an object-oriented system. They may represent a person, a place, a bank account, a table of data or any item that the program has to handle. They may also represent user-defined data such as vectors, time and lists. Programming problem is analyzed in term of objects and the nature of communication between them. Program objects should be chosen such that they match closely with the real-world objects. Objects take up space in the memory and have an associated address like a record in Pascal, or a structure in c.

OBJECTS: STUDENT

DATA

Name
Date-of-birth
Marks

FUNCTIONS

Total
Average
Display
.....

Classes

6

The entire set of data and code of an object can be made a user-defined data type with the help of class. In fact, objects are variables of the type class. Once a class has been defined, we can create any number of objects belonging to that class. Each object is associated with the data of type class with which they are created. A class is thus a collection of objects similar types. For examples, Mango, Apple and orange members of class fruit. Classes are user-defined that types and behave like the built-in types of a programming language. The syntax used to create an object is not different then the syntax used to create an integer object in C. If fruit has been defines as a class, then the statement

Fruit Mango;

Will create an object mango belonging to the class fruit.

Data Encapsulation

7

The wrapping up of data and function into a single unit (called class) is known as **encapsulation**. Data and encapsulation is the most striking feature of a class. The data is not accessible to the outside world, and only those functions which are wrapped in the class can access it. These functions provide the interface between the object's data and the program. This insulation of the data from direct access by the program is called data hiding or information hiding.

Data Abstraction

8

- Abstraction refers to the act of representing essential features without including the background details or explanation. Classes use the concept of abstraction and are defined as a list of abstract attributes such as size, weight, and cost, and functions operate on these attributes. They encapsulate all the essential properties of the object that are to be created.
- The attributes are sometimes called data members because they hold information. The functions that operate on these data are sometimes called methods or member functions.

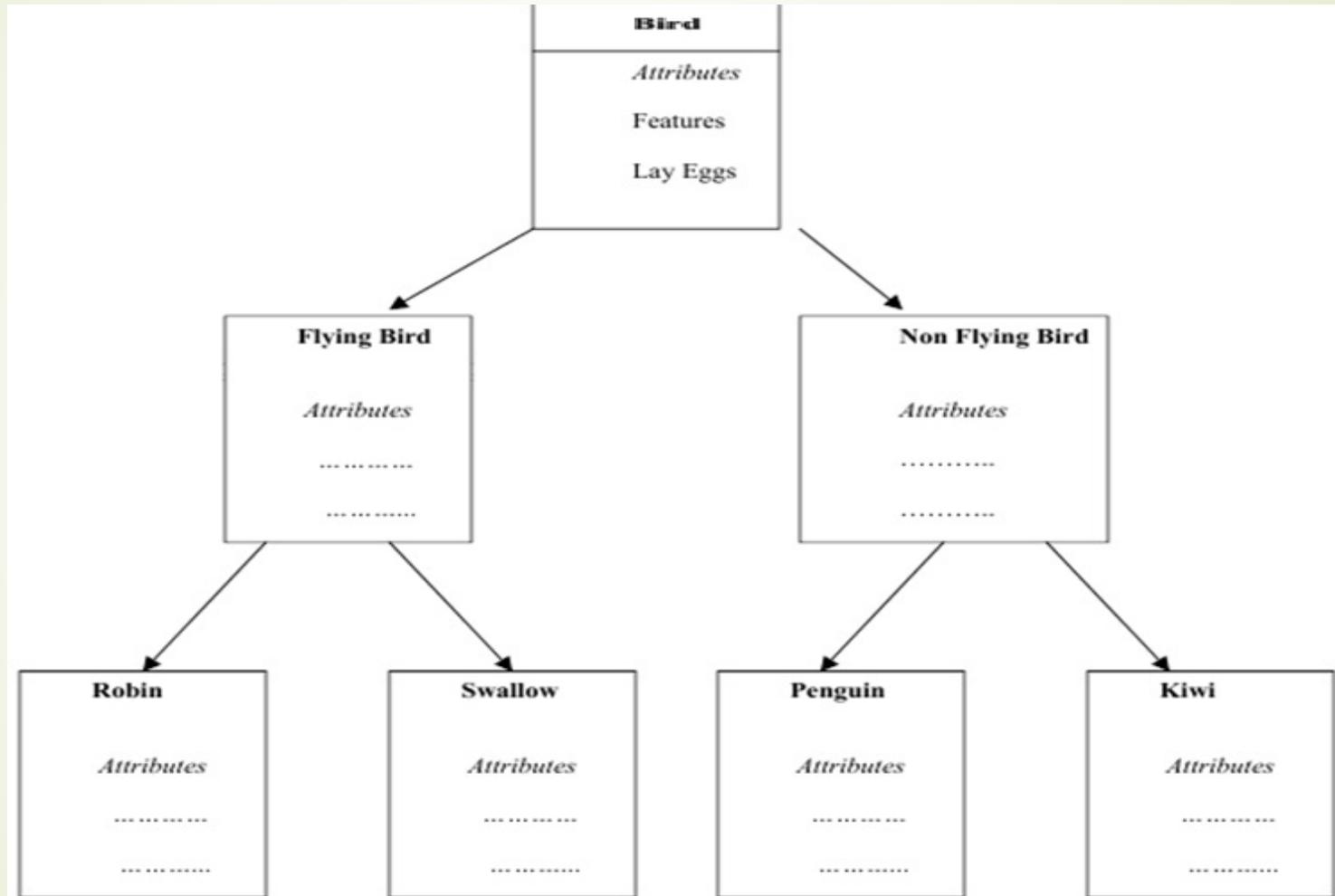
Inheritance

9

- Inheritance is the process by which objects of one class acquired the properties of objects of another classes. It supports the concept of hierarchical classification.
- The concept of inheritance 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 new class from the existing one. The new class will have the combined feature of both the classes.
- The real appeal and power of the inheritance mechanism is that it allows the programmer to reuse a class i.e. almost, but not exactly, what he wants, and to tailor the class in such a way that it does not introduced any undesirable side-effects into the rest of classes.

Example of Inheritance

10



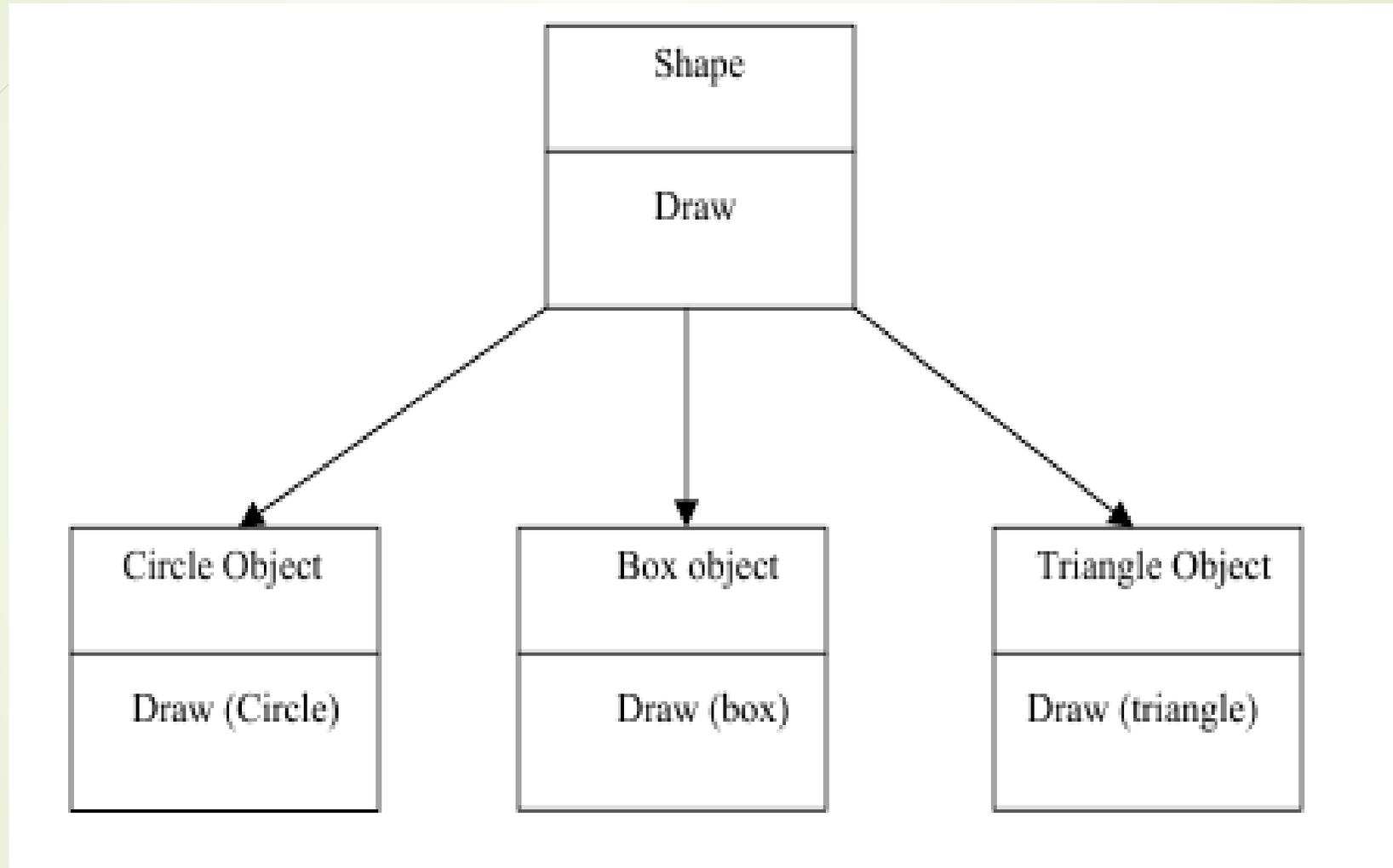
Polymorphism

11

- Polymorphism, a Greek term, means the ability to take more than one form. An operation may exhibit different behavior in different instances. The behavior depends upon the types of data used in the operation. For example, consider the operation of addition. For two numbers, the operation will generate a sum. If the operands are strings, then the operation would produce a third string by concatenation. The process of making an operator to exhibit different behaviors in different instances is known as **operator overloading**. Using a single function name to perform different type of task is known as **function overloading**.
- 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 operations may be accessed in the same manner even though specific action associated with each operation may differ. Polymorphism is extensively used in implementing inheritance.

Example of Polymorphism

12



Dynamic Binding

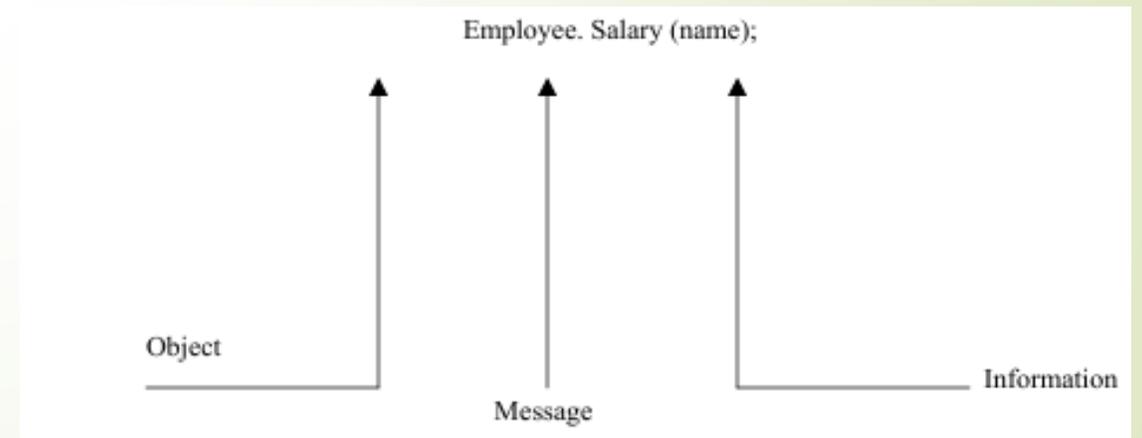
13

Binding refers to the linking of a procedure call to the code to be executed in response to the call. Dynamic binding means that the code associated with a given procedure call is not known until the time of the call at run time. It is associated with polymorphism and inheritance. A function call associated with a polymorphic reference depends on the dynamic type of that reference.

Message Passing

14

A Message for an object is a request for execution of a procedure, and therefore will invoke a function (procedure) in the receiving object that generates the desired results. Message passing involves specifying the name of object, the name of the function (message) and the information to be sent. Example:



Advantages of Object Oriented Programming

15

- Through inheritance, we can eliminate redundant code extend the use of existing Classes.
- 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.
- The principle of data hiding helps the programmer to build secure program that can not be invaded by code in other parts of a programs.
- It is possible to have multiple instances of an object to co-exist without any interference.
- It is possible to map object in the problem domain to those in the program.
- It is easy to partition the work in a project based on objects.
- The data-centered design approach enables us to capture more detail of a model can implemental form.
- Object-oriented system can be easily upgraded from small to large system.
- Message passing techniques for communication between objects makes to interface descriptions with external systems much simpler.
- Software complexity can be easily managed.

Assignment

16

1. How are data and functions organized in an object-oriented program?
2. What are the unique advantages of an object-oriented programming paradigm?
3. Distinguish between the following terms:
 - (a) Object and classes
 - (b) Data abstraction and data encapsulation
 - (c) Inheritance and polymorphism
 - (d) Dynamic binding and message passing.
4. Describe inheritance as applied to object-oriented program.
5. What do you mean by dynamic binding? How it is useful in object-oriented program?



Thank You