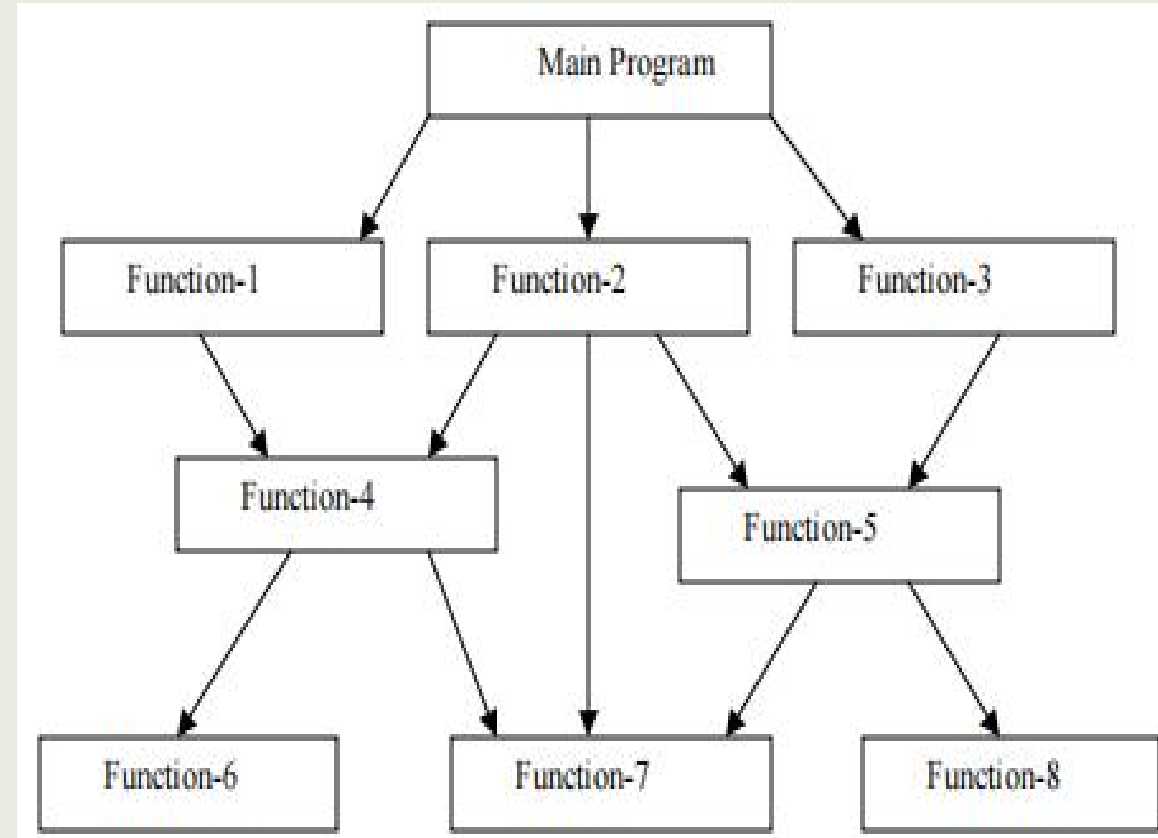


Chapter 1: Introduction to OOP

BIBHA STHAPIT
ASST LECTURER
IOE, PULCHOWK CAMPUS

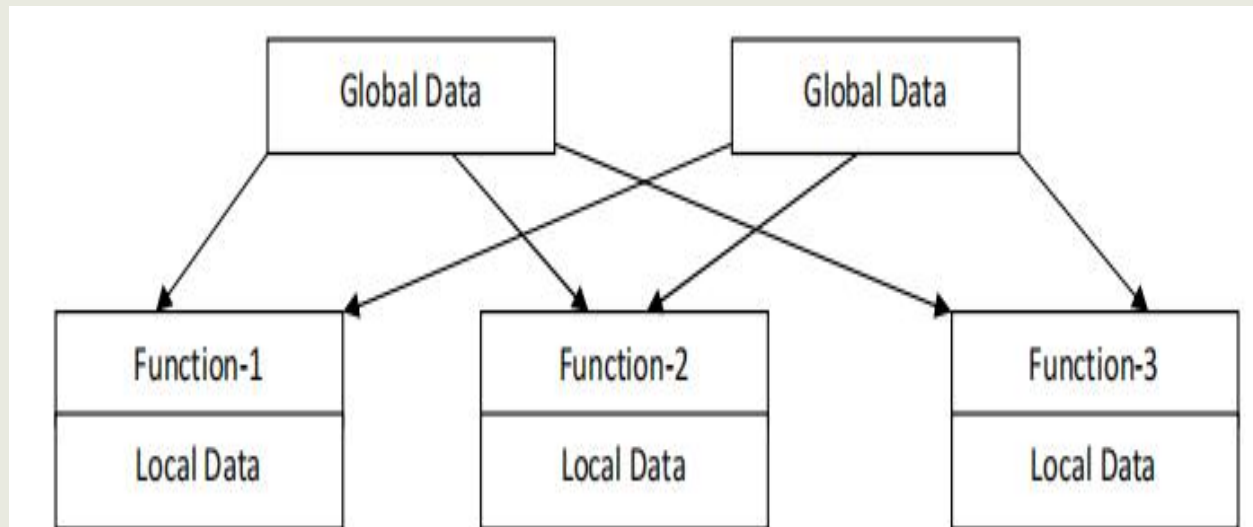
Procedure-Oriented Programming

- In the procedure oriented approach, the problem is viewed as the sequence of things to be done such as reading, calculating and printing such as cobol, fortran and c.
- The primary focus is on functions.
- The technique of hierarchical decomposition has been used to specify the tasks to be completed for solving a problem.



Procedure-Oriented Programming

- In a multi-function program, many important data items are placed as global so that they may be accessed by all the functions.
- Global data are more vulnerable to an accidental change by a function.
- Another drawback with the procedural approach is that we do not model real world problems very well. This is because functions are action-oriented and do not really corresponding to the element of the problem.

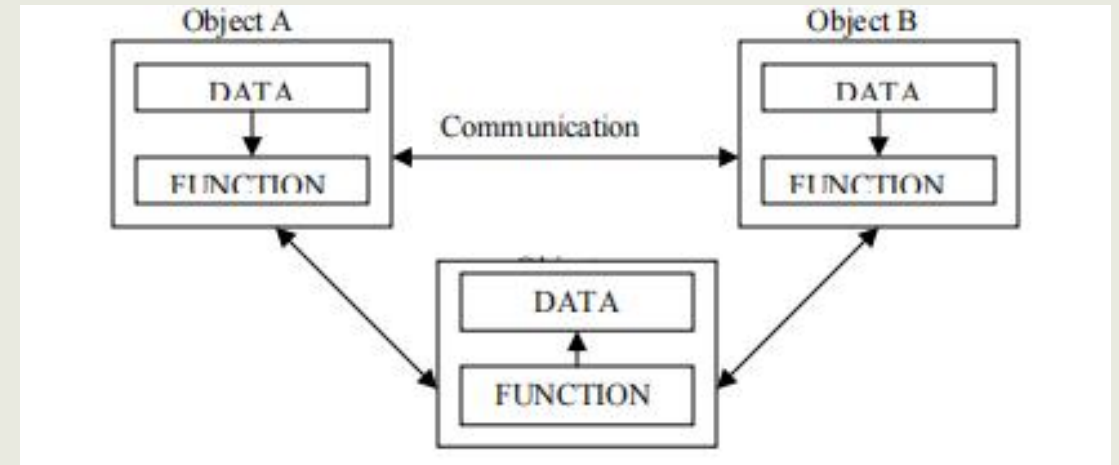


Procedure-Oriented Programming

- Some Characteristics exhibited by procedure-oriented programming are:
 - Emphasis is on doing things (algorithms).
 - Large programs are divided into smaller programs known as functions.
 - Most of the functions share global data.
 - Data move openly around the system from function to function.
 - Functions transform data from one form to another.
 - Employs top-down approach in program design.

Object Oriented Paradigm

- OOP 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 function that operate on it, and protects it from accidental modification from outside function.
- OOP allows decomposition of a problem into a number of entities called objects and then builds data and function around these objects.
- The data of an object can be accessed only by the function associated with that object. However, function of one object can access the function of other objects.



Object Oriented Paradigm

- Some of the features of object oriented programming are:
 - 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.

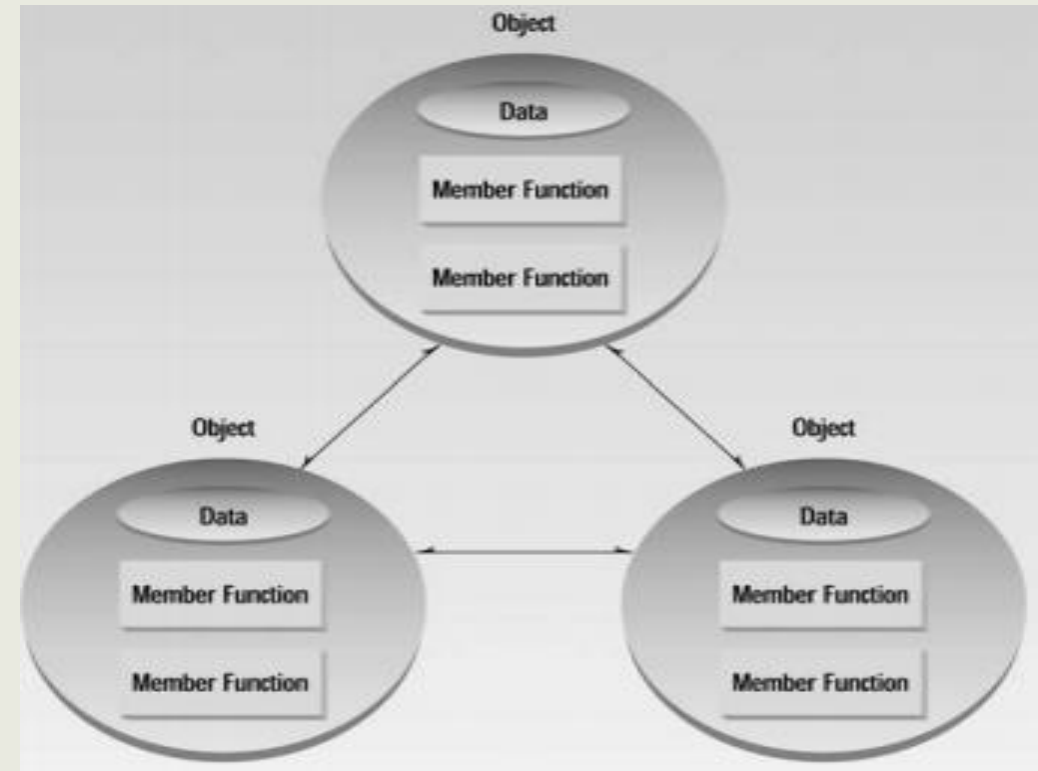
Basic Concepts of Object Oriented Programming

- It is necessary to understand some of the concepts used in object-oriented programming. These include:
 - Objects
 - Classes
 - Data encapsulation
 - Data abstraction
 - Inheritance
 - Polymorphism
 - Dynamic binding
 - Message passing

Basic Concepts of Object Oriented Programming

- **Objects**

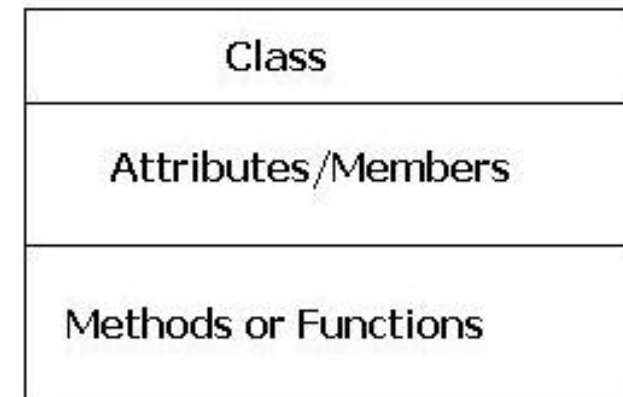
- Objects are the basic *runtime entities* in an object oriented system.
- They may represent a person, a place, a bank account, a table of data or any item that the program has to handle such as vectors, time and lists.
- Objects take up space in the memory and have an associated address like a structure in c.



Basic Concepts of Object Oriented Programming

- **Class**

- Object contains data, and code to manipulate that data.
- The entire set of data and code of an object can be made a ***user-defined data type*** with the help of a 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.
- Classes are user-defined that types and behave like the built-in types of a programming language.



Basic Concepts of Object Oriented Programming

- **Data Encapsulation**

- The wrapping up of data and functions into a single unit is known as encapsulation.
- The data is not accessible to the outside world, only those function which are wrapped in them 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*

Basic Concepts of Object Oriented Programming

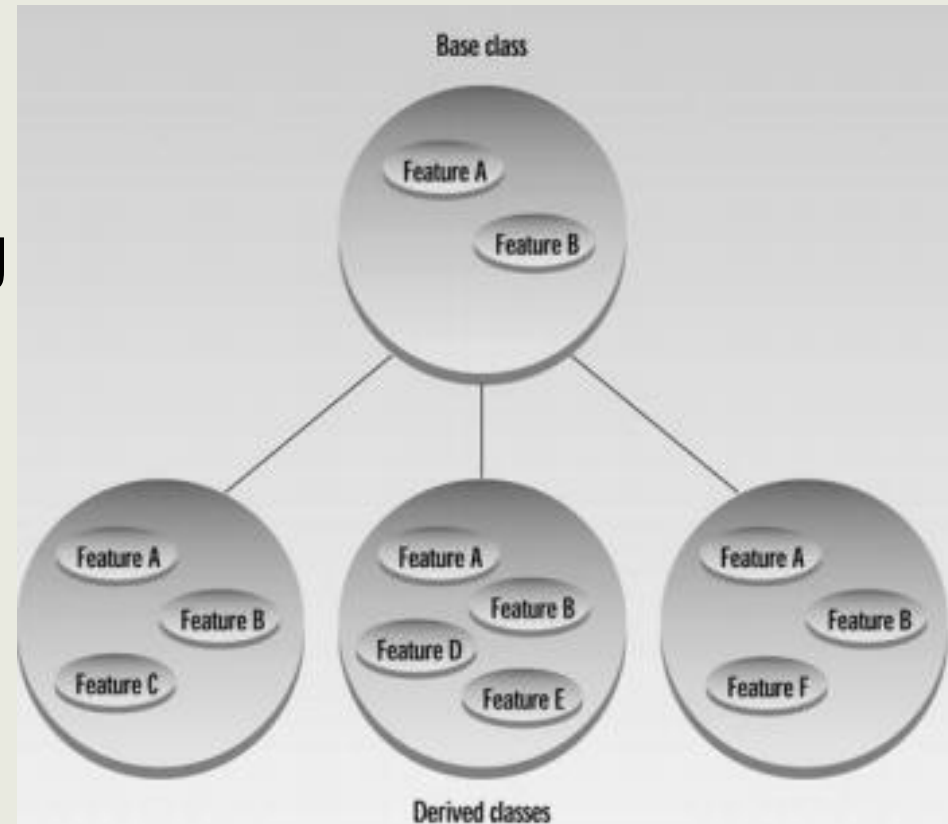
- **Data Abstraction**

- Abstraction refers to the act of representing essential features without including the background details or explanations.
- Since classes use the concept of data abstraction, they are known as ***Abstract Data Types*** (ADT)
- To build an ADT, one must be able to:
 - Support a type definition.
 - Make available a set of operations that can be used to manipulate instance of the type.
 - Protects the data associated with the type so that can be used to manipulate instances of the type.
 - Making multiple instance of the type.

Basic Concepts of Object Oriented Programming

- **Inheritance**

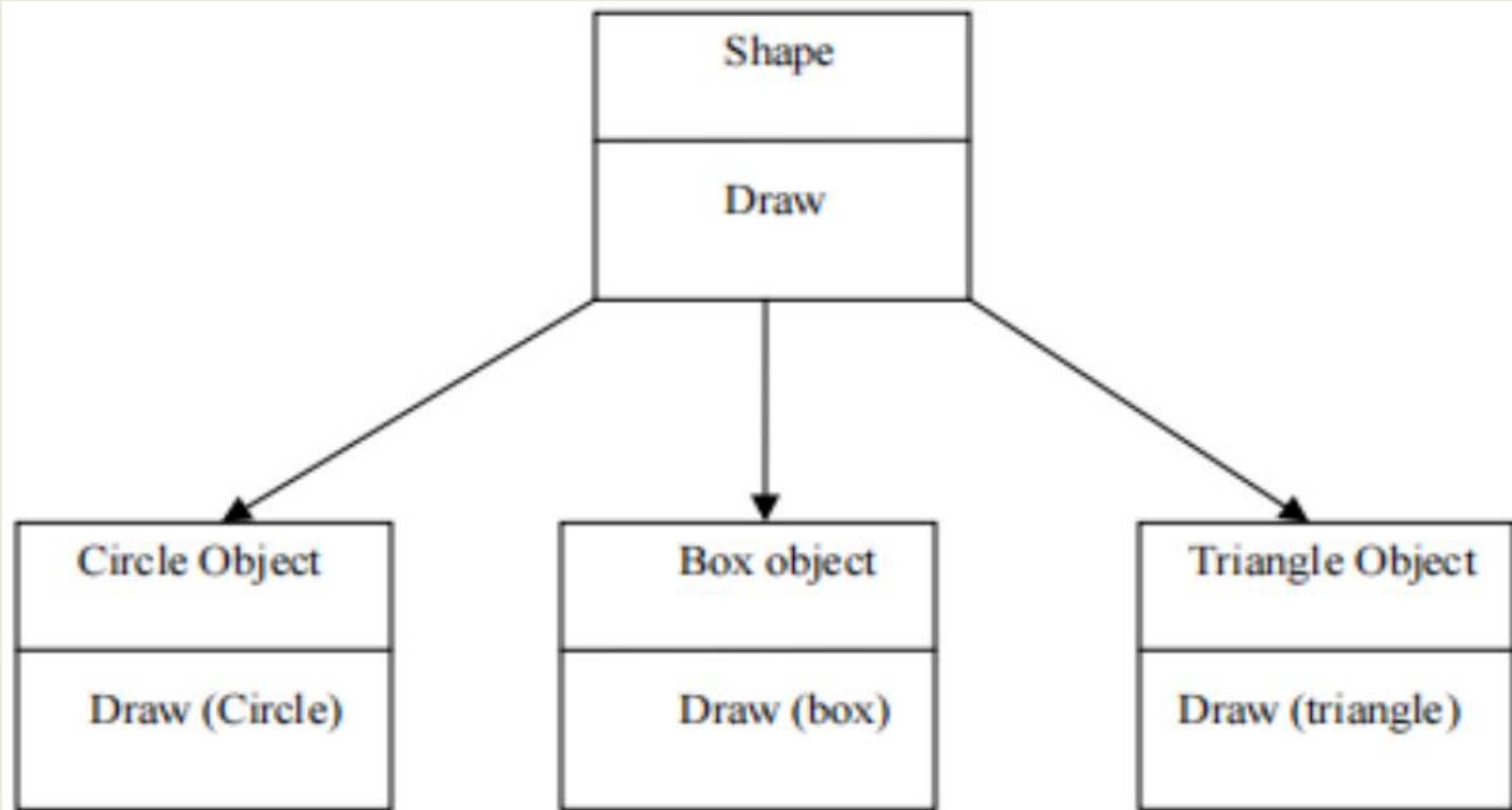
- Inheritance is the process by which objects of one class acquire the properties of objects of another class.
- In OOP, the concept of inheritance provides the idea of *reusability*. This means we can add additional features to an existing class without modifying it.
- The original class is called the *base class*; other classes can be defined that share its characteristics, but add their own as well. These are called *derived classes*.



Basic Concepts of Object Oriented Programming

- **Polymorphism**

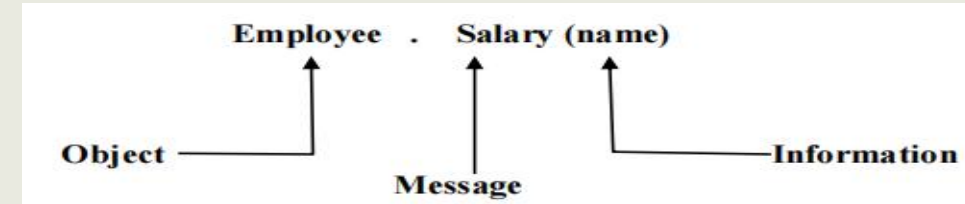
- Polymorphism, a Greek term means to ability to take more than one form.
- An operation may exhibits different behaviors in different instances. The behavior depends upon the type of data used in the operation.
- For example consider the operation of addition for two numbers; the operation will generate a sum. If the operands are string then the operation would produce a third string by concatenation.
- The process of making an operator to exhibit different behavior in different instances is known as ***operator overloading***.
- Similarly, in ***function overloading***, the types of the arguments with which the function is called determines which definition will be used



Basic Concepts of Object Oriented Programming

- **Message Passing**

- An object oriented program consists of a set of objects that communicate with each other.
- The process of programming in an object-oriented language, involves the following basic steps:
 1. Creating classes that define object and their behavior,
 2. Creating objects from class definitions, and
 3. Establishing communication among objects.
- 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 result. Message passing involves specifying the name of the object, the name of the function (message) and information to be sent.



Benefits of Object Oriented Programming

- 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 an 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.

Benefits of Object Oriented Programming

- 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.

End of Chapter 1