OBJECT ORIENTED PROGRAMMING COCEPTS

PROGRAMMING APPROACHES

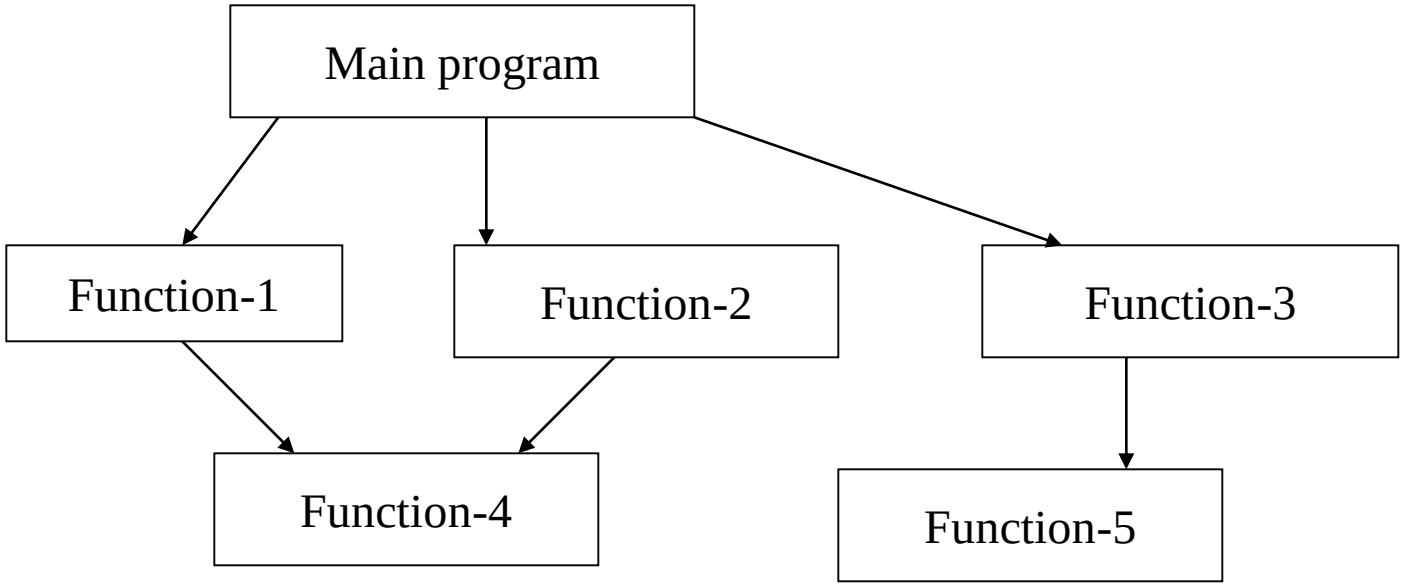
- All computer programs consist of two elements code and data.
- A program can be conceptually organized around its code or around its data.
- Two programming Approaches
  - Procedure oriented programming
  - Object oriented programming.

Procedure Oriented Programming

- The procedure oriented model can be thought of as code acting on data.
- Problems with this approach appear as programs grow larger and more complex.

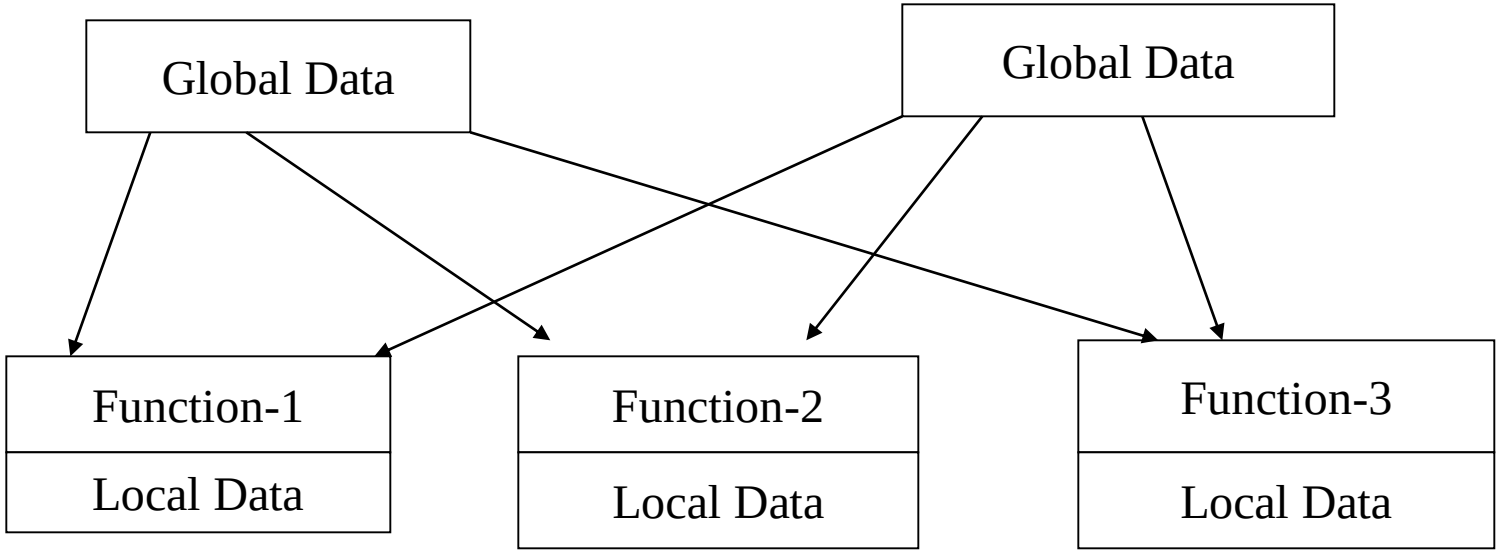
Procedure Oriented Programming

- The problem is viewed as a sequence of actions and a number of functions are written to solve these actions.
- Large problem is divided into smaller programs known as modular programming.



Procedure Oriented Programming

- Data move openly around the system from function to function.

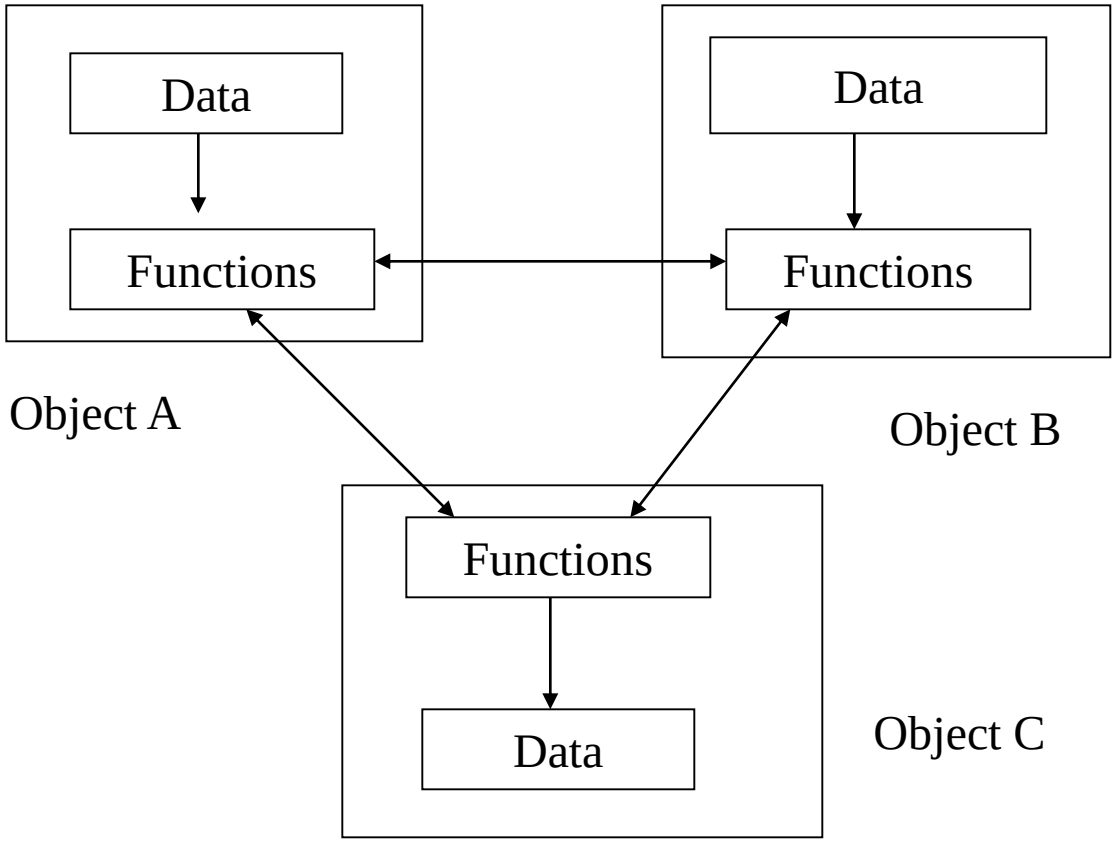


OBJECT ORIENTED PROGRAMMING

- To manage increasing complexity, the second approach called object oriented programming was conceived.
- OOP organizes a program around its data (that is, objects) and a set of well defined interfaces to that data.
- Data controlling access to code.
- OOP is the core of Java.

Object –Oriented Programming

- Emphasis is on data rather than procedure.
- Programs are divided into objects.



Differences between Procedural and OO Programing

Procedural	OO programming
Code is placed into totally distinct functions or procedures	Everything treated as an Object
Data placed in separate structures and is manipulated by these functions or procedures	Every object consist of attributes(data) and behaviors (methods)
Code maintenance and reuse is difficult	Code maintenance and reuse is easy
Data is uncontrolled and unpredictable (i.e. multiple functions may have access to the global data)	The data of an object can be accessed only by the methods associated with the object
You have no control over who has access to the data	Good control over data access
Testing and debugging are much more difficult	Testing and debugging are much easy
Not easy to upgrade	Easy to upgrade

OBJECT ORIENTED PROGRAMMING

- OOP is an approach to program organization and development, which attempts to eliminate some of the drawbacks of conventional programming methods.

- ***OOP allows us to decompose a problem into number of entities called objects and then build data and methods (functions) around these entities.***

- ***The data of an object can be accessed only by the methods associated with the object.***

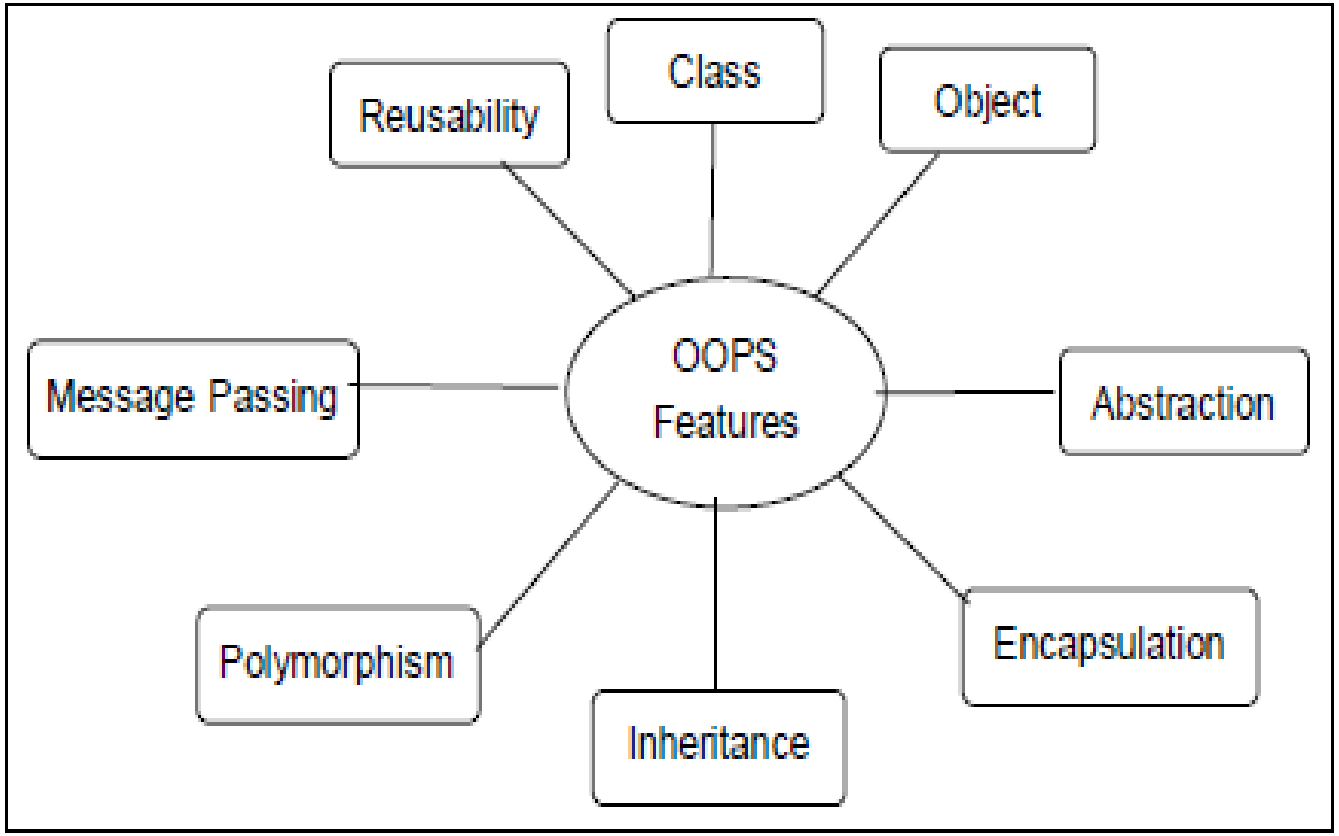
OBJECT ORIENTED PROGRAMMING

- Object-oriented programming (OOP) is a programming paradigm that uses “Objects “and their interactions to design applications.

- It simplifies the software development and maintenance by providing some concepts:
  - Object
  - Class
  - Data Abstraction & Encapsulation
  - Inheritance
  - Polymorphism
  - Dynamic Binding
  - Message Passing

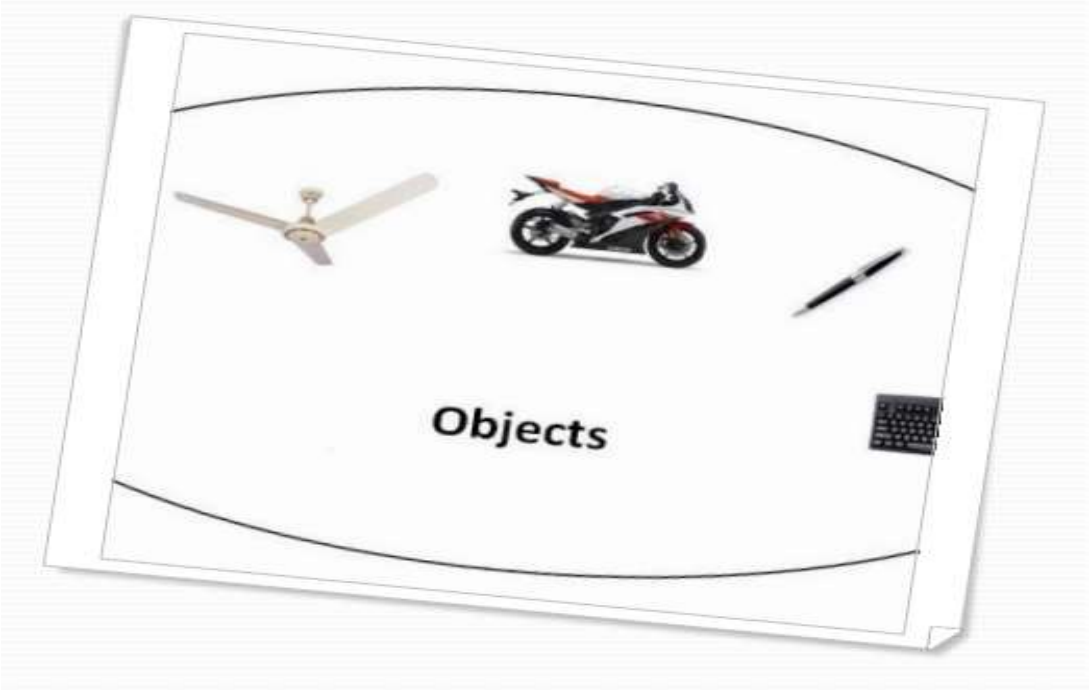
Basic Concepts of Oop

The general concepts of OOP which forms the heart of Java language are:



Object

- Object : 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



Class

- Class : The entire set of data and code of an object can be made of a user defined data type with the help of a class.
- Class is the template or blueprint from which objects are made.
- 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.
-

Class

- Characteristics of an object are represented in a class as Properties.
- The actions that can be performed by objects become functions of the class and is referred to as Methods.
- A class is thus a collection of objects of similar type.
- For example: mango, apple, and orange are members of the class fruit.
- Example :

fruit mango;

will create an object mango belonging to the class fruit.

Abstraction

- Abstraction refers to the act of representing essential features without including the background details or explanations. (hiding internal implementation is called Abstraction)
- Since the classes use the concept of data abstraction ,they are known as abstract data type(ADT).
- For example, when we apply brake to our two wheeler, bike stops. But, we don’t know the internal mechanism of how brake works. Still, we use break.
- Classes use the concept abstraction

Abstraction

- We can enhance the internal implementation without effecting outside world.
- Abstraction provides security.
- A class contains lot of data and the user does not need the entire data.
- The advantage of abstraction is that every user will get his own view of the data according to his requirements and will not get confused with unnecessary data.

ENCAPSULATION

- **Data Encapsulation: The wrapping (combining) up of data and methods into a single unit is** called as encapsulation.

- The data is not accessible at outside of the world.
- Data is accessed by only those methods, which are wrapped in class.

- Data of one object cannot be accessible to other objects.
- The insulation of data from direct access by the program is known as data hiding.

ENCAPSULATION

- Class is an example for encapsulation.
- Encapsulation can be described as a protective barrier that prevents the code and data being randomly accessed by other code defined outside the class

INHERITANCE

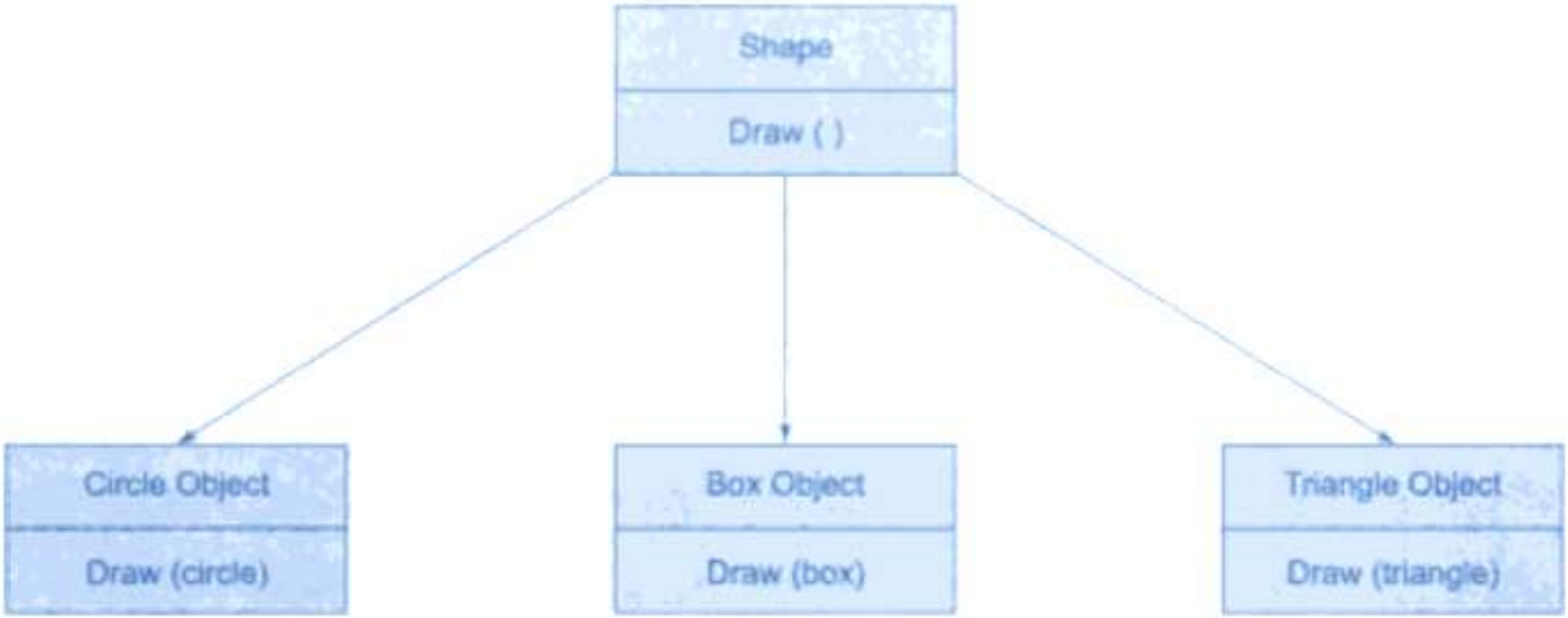
- **Inheritance: The process of deriving a new class from an existing class is known as inheritance.**
- In this process, an object of one class acquires the properties of objects of another class.
- The new class is called derived class or subclass or child class and the existing class is called as base class or super class or parent class.
- It provides reusability, by adding additional features to an existing class without modifying it.
- The child class inherits characteristics of the parent class(i.e the child class inherits the methods and data defined for the parent class.

Property inheritance

POLYMORPHISM

- **Polymorphism: Polymorphism means the ability to take more than one form.**
- It plays an important role in allowing objects having different internal structure to share the same external interfaces.
- This operation exhibits different behavior in different times.

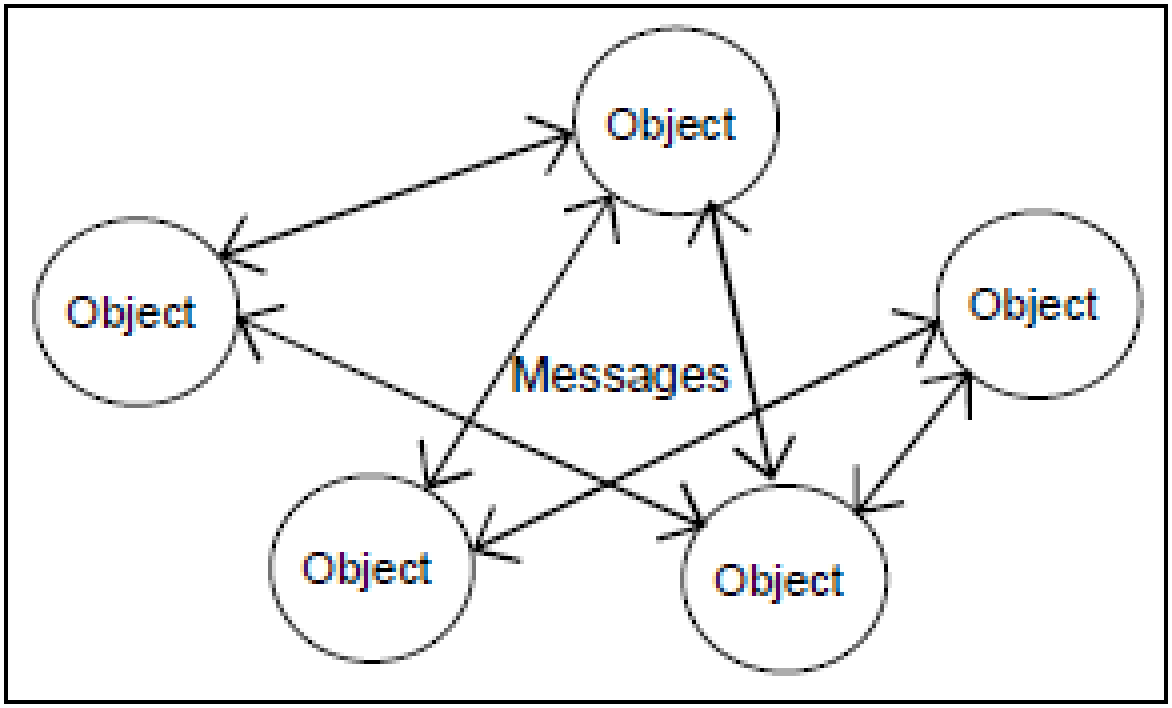
POLYMORPHISM



MESSAGE PASSING

- Message Passing: Calling a method in a class is called message passing.
- Message Passing: Objects communicate with each other by sending and receiving information as people pass messages to one another.
- A message for an object is a request for execution of a procedure, and therefore will invoke a method in receiving the object that generates the desired result.

MESSAGE PASSING



Interaction of objects via message passing

Dynamic Binding

- **Dynamic Binding: 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 call at runtime.

Reusability

- **Reusability: The term reusability refers to the ability for multiple programs to use the same** written and debugged existing class of data.
- This is time saving and adds code efficiency to the language.
- It optimizes code, helps in gaining secured applications and facilitates easier maintenance on the application.