Object Oriented Programming and Design

Programming Task/Quizzes followed by Oral	40 Marks
End Semester Exam	60 Marks

Prof. Harish D.G.
Dept. of Computer and IT
College of Engineering,Pune
www.harishgadade.com

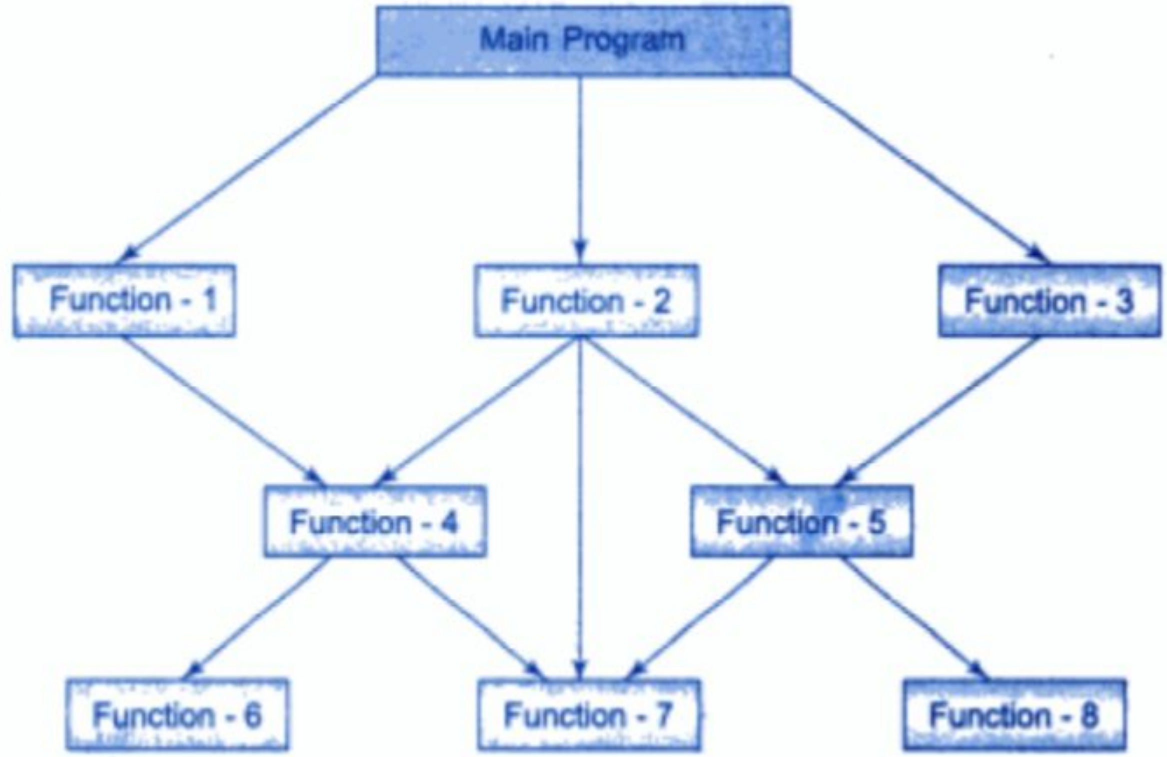
Programming Paradigm

1. Imperative Programming Paradigm
    a. Procedural Programming Paradigm
    b. Object Oriented Programming Paradigm
    c. Parallel Processing/Concurrent Programming Approach
2. Declarative Programming Paradigm
    a. Logic Programming Paradigm
    b. Functional Programming Paradigm
    c. Database Programming Approach

Procedural Programming

- In procedural-Oriented approach,the problem is viewed as a sequence of things to be done, such as reading, calculation and printing.
- A number of functions are written to accomplish these task.
- Primary focus is on function

Procedural Programming



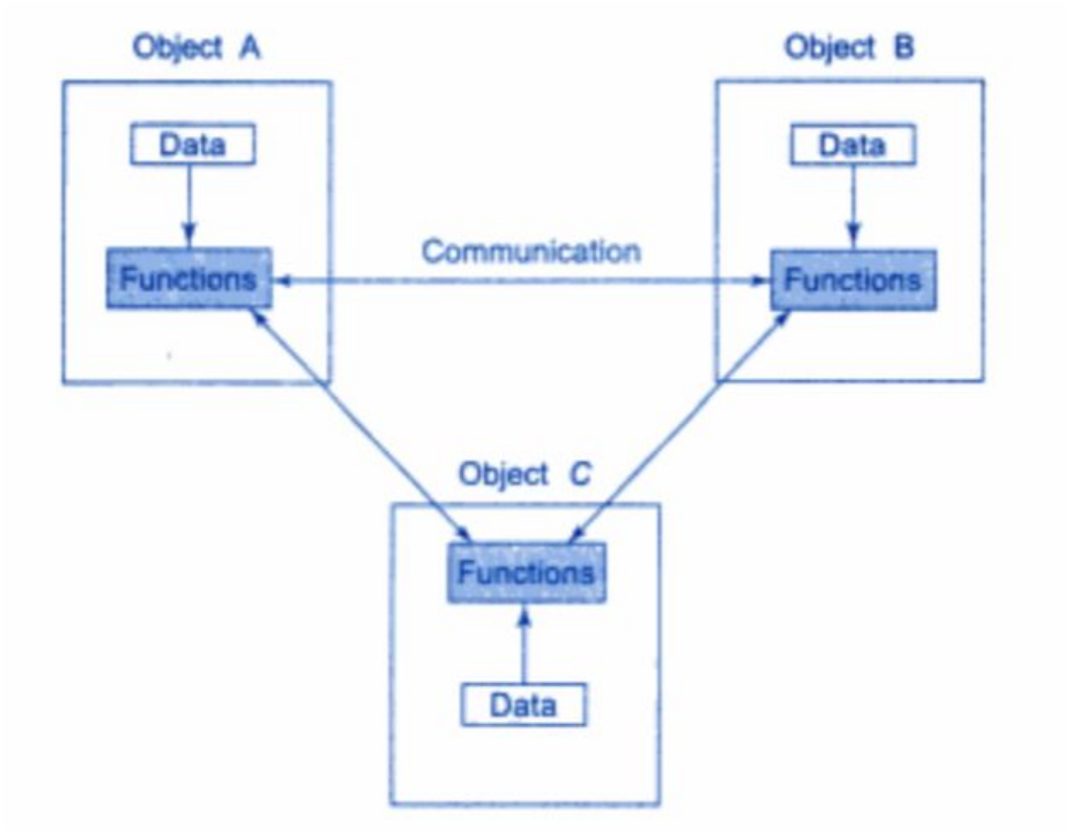
Procedural Programming

- The code reusability feature is not present in the procedure oriented programming.
- we have to write the same programming code to many times .
- We can not perform encapsulation,inheritance etc in the procedure oriented programming.
- Difficult to relate with real-world objects.
- Global data is more vulnerable.

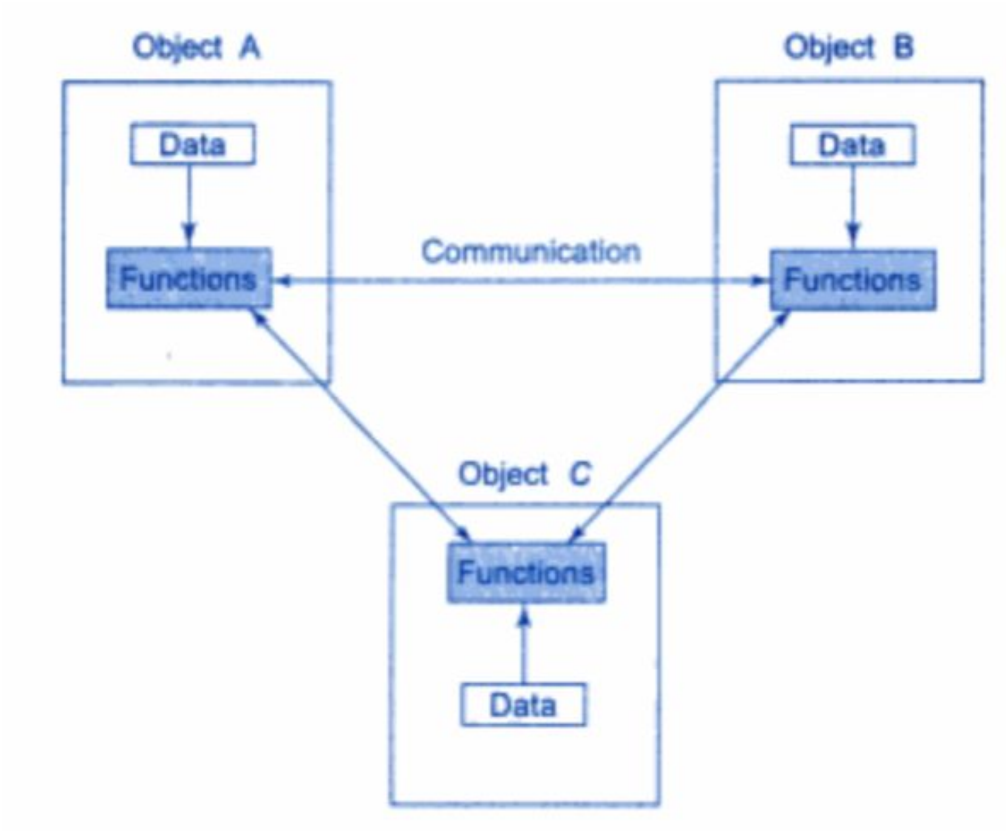
Object-Oriented Programming

- 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 functions that operate on it and protect it from accidental modifications from outside function.
- OOP allow us to decompose a problem into a number of entities called objects and then build data and functions around these entities.

Object-Oriented Programming



Object-Oriented Programming



Features:

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

Basic Concepts of OOP

In OOPs following concepts are used

1. Objects
2. Classes
3. Data Encapsulations
4. Inheritance
5. Polymorphisms
6. Dynamic Binding
7. Message Passing

1. Objects

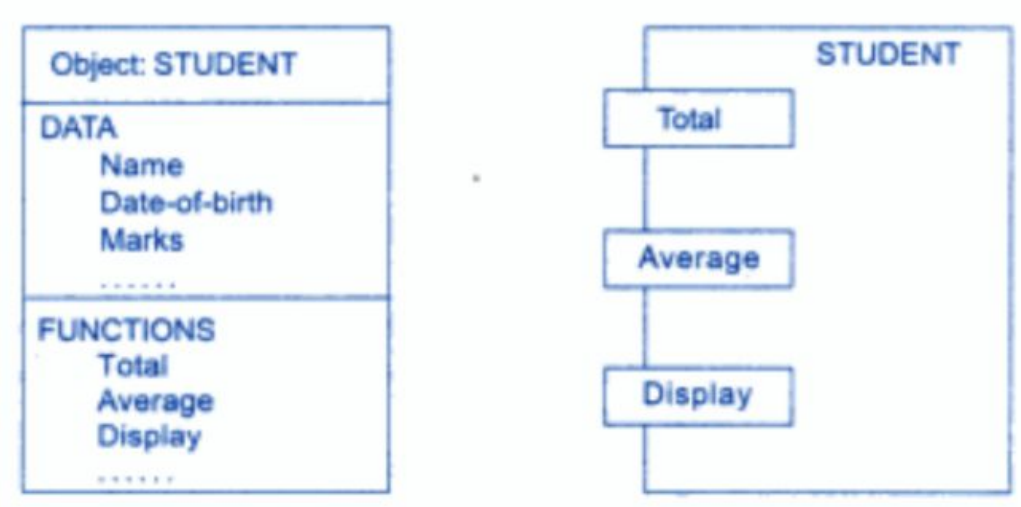


Fig. Two ways of representing an object

- Objects are basic runtime entities in OOPs
- They may represents a person, a place,a bank account, a table of data or any item that the program has to handle.
- When program is executed, the objects can interact with each other by sending messages
- E.g. if ***"customer"*** and ***"account"*** are two objects,then a ***"customer"*** object may send a message to ***"account"*** object requesting the bank balance.

2. Class

- A class is a collection of objects of similar type.
- As we mentioned that objects contain 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 variable of type class.
- For example, mango, apple, orange are the members of class fruit.

```
fruit mango;
```

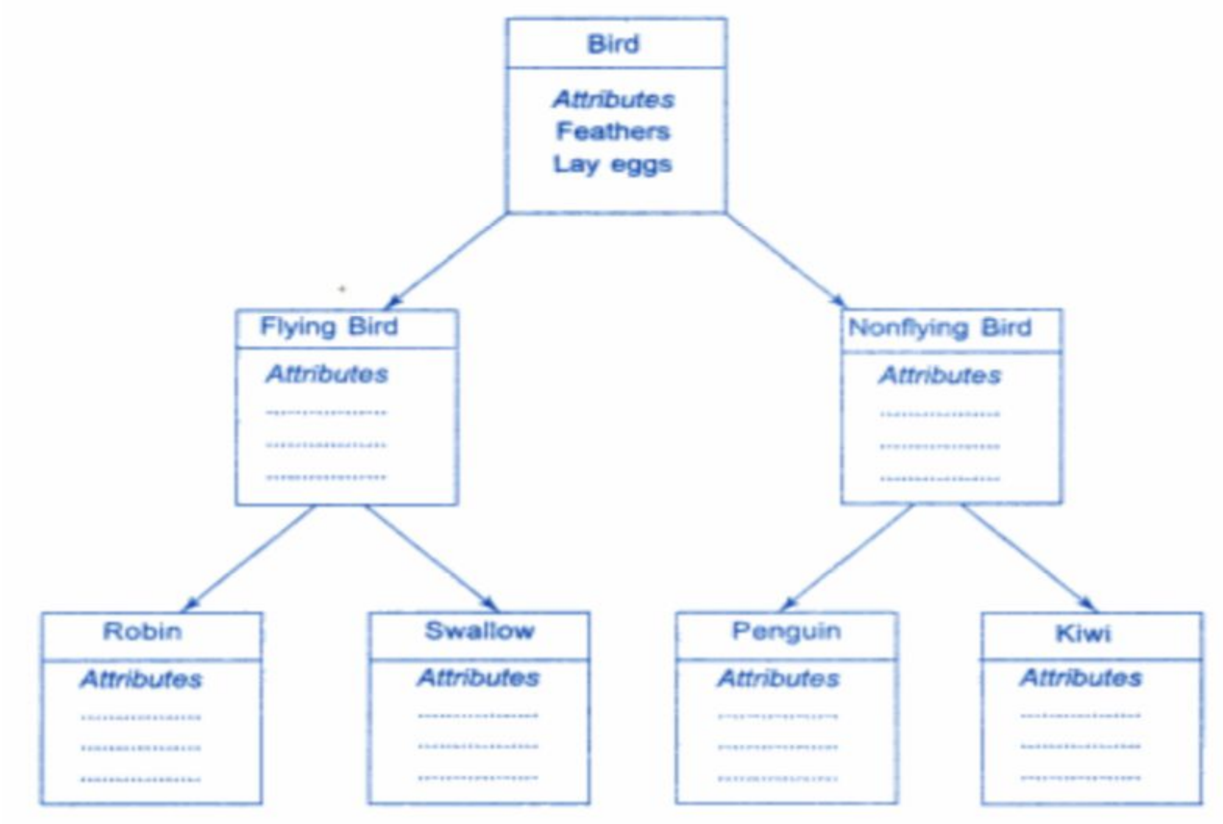
3. Data Abstraction and Encapsulation

- The wrapping up of data and functions into a single unit(called class) is known as Encapsulation.
- Data encapsulation is the most striking feature of a class because data is not accessible to the outside world and only those functions which are wrapped in the class can access it.
- Abstraction means displaying only essential information and hiding the details.
- Data abstraction refers to providing only essential information about the data to the outside world, hiding the background details or implementation.

4. Inheritance

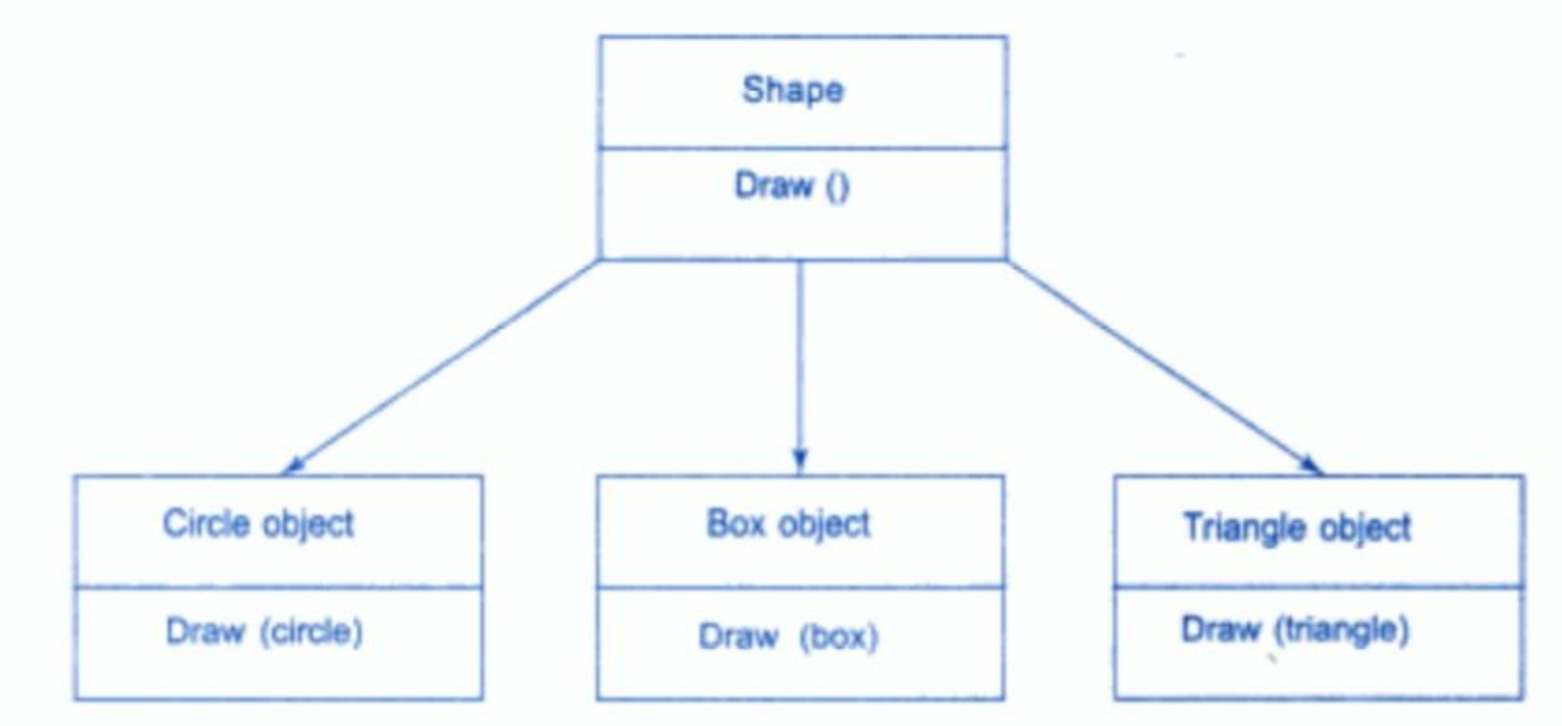
- Inheritance is the process by which objects of one class can acquire the properties of objects of another class.
- E.g. The bird robin is a part of the class *"flying bird"* which is again a part of the class *"bird".*
- In OOPs, the concept of inheritance provides the idea of reusability, means we can add additional features to an existing class without modifying it.
- This is possible by deriving a new class from the existing class. New class will have the combined features of both classes.

4. Inheritance



5. Polymorphism

- Polymorphism means the ability to take more than one form



6. 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 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.

7. Message Communication

- An OOP consists of a set of objects that communicates with each other. It consists of following steps
    - Creating classes that defines objects and their behaviour.
    - Creating objects from class definitions.
    - Establishing communication among objects.