Unit I: Introduction to OPP

Programming paradigms:

Procedure oriented programming (POP):

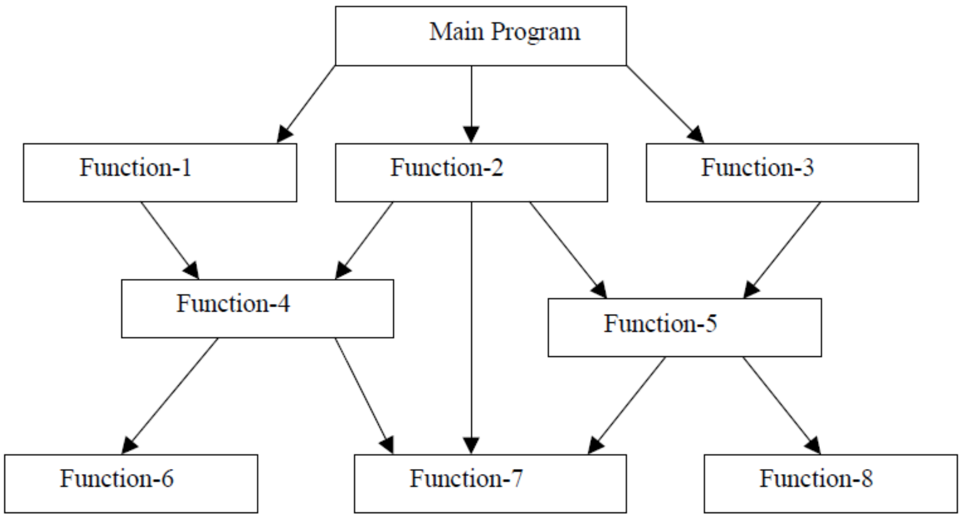
Procedure oriented programming exhibit following characteristics

- Procedure is nothing but function or method.
- 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.

Note

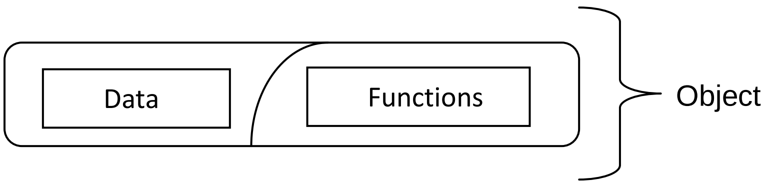
The drawback of procedure oriented programming is that “It is not suitable for to handle the real world problems”.

Diagram to represent the procedure oriented programming



Object oriented programming (OOP):

Object is a smallest unit of program which consists of a data and functions.
Object is a representation of real world entities like student, principal, employee etc.



Object is a capsule which consists of data and functions in it.

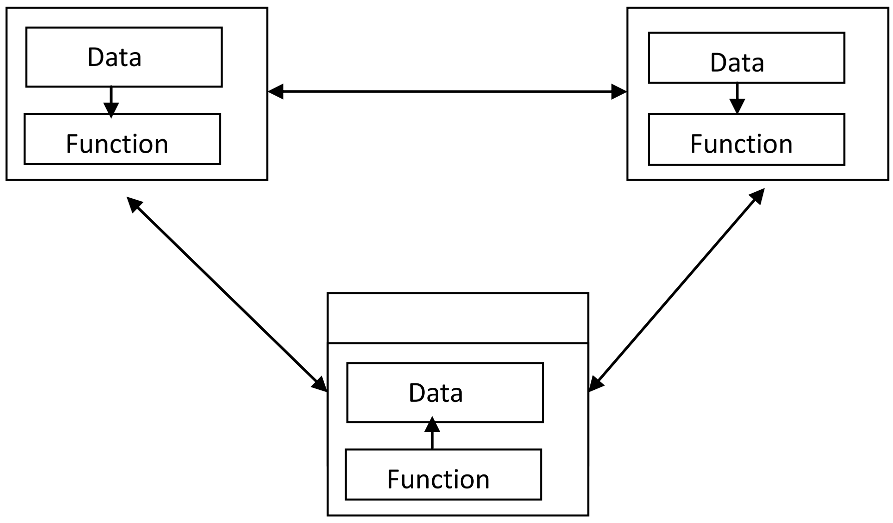


Diagram to represent the object oriented programming

Object oriented programming exhibit following characteristics

- 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 ties 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 OOP and Features:

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

We shall discuss these concepts in some detail in this section.

Objects

- The object represents real world entity.
- 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.
- When a program is executed, the objects interact by sending messages to one another.

For example, if “customer” and “account” are to object in a program, then the customer object may send a message to the count object requesting for the bank balance. Each object contain data, and code to manipulate data. Objects can interact without having to know details of each other’s data or code. It is a sufficient to know the type of message accepted, and the type of response returned by the objects.

Classes

- Class is blue print for action.
- The entire set of data and code of an object can be made a user-defined data type with the help of class.
- 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 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;
- It creates an object **mango** belonging to the class **fruit.**

Data Abstraction and Encapsulation

- Data and encapsulation is the most striking feature of a class.
- The wrapping up of data and function into a single unit (called class) is known as encapsulation.
- 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.

- 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, wait, and cost, and function operate on these attributes.
- Abstraction acts as a shield between end user and code.
- Through abstraction user is unaware of internal complexity of the software system.

Inheritance

- Inheritance is the process by which objects of one class acquired the properties of objects of another classes.
  Or
- Inheritance is the flow of properties from one class to another class.
- It supports the concept of hierarchical classification.
- Through inheritance reusing of the code can be possible.
- Inheritance increases the performance of the program, speed of delivery and assures the less chances of errors.

Polymorphism

- Polymorphism, a Greek term, means the ability to take more than on form.
- An operation may exhibit different behaviour is different instances.
- If an operator to exhibit different behaviours in different instances is known as operator overloading.
- If a function exhibit different behaviours in different instances is known as function overloading.

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 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

- An object-oriented program consists of a set of objects that communicate with each other.
- Objects communicate with one another by sending and receiving information much the same way as people pass messages to one another.
- The concept of message passing makes it easier to talk about building systems that directly model or simulate their real-world counterparts.
- 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.

- Object has a life cycle. They can be created and destroyed. Communication with an object is feasible as long as it is alive.

- Example:

student.getResult(int rollNo);

Here:
student : Object
getResult:Message
rollNo: Information.

Benefits of OOPS

OOP offers several benefits to both the program designer and the user. Object-Orientation contributes to the solution of many problems associated with the development and quality of software products. The new technology promises greater programmer productivity, better quality of software and lesser maintenance cost.

The principal advantages are:

- 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 cannot 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-cantered 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.

Object Oriented Language

- Object-oriented programming is not the right of any particular languages.
- OOP concepts can be implemented using languages such as C and Pascal but programming becomes clumsy and may generate confusion when the programs grow large. A language that is specially designed to support the OOP concepts makes it easier to implement them.
- The languages should support several of the OOP concepts to claim that they are object oriented.
- Depending upon the features they support, they can be classified into the following two categories:

1. Object-based programming languages, and
2. Object-oriented programming languages.

Object-based programming is the style of programming that primarily supports encapsulation and object identity. Major feature that are required for object based programming are:

- Data encapsulation
- Data hiding and access mechanisms
- Automatic initialization and clear-up of objects
- Operator overloading

- Languages that support programming with objects are said to the objects-based programming languages.
- They do not support inheritance and dynamic binding. Ada is a typical object-based programming language.
- Object-oriented programming language incorporates all of object-based programming features along with two additional features, namely, inheritance and dynamic binding.
- Object-oriented programming can therefore be characterized by the following statements:

    - Object-based features + inheritance + dynamic binding

Application of OOP

OOP has become one of the programming buzzwords today.

The applications of the OOP are as follows

- Real-time system
- Simulation and modelling
- Object-oriented data bases
- Hypertext, Hypermedia, and expertext
- AI and expert systems
- Neural networks and parallel programming
- Decision support and office automation systems
- CIM/CAM/CAD systems

Sign Up

And

Download

Full

Notes

☺☺☺