



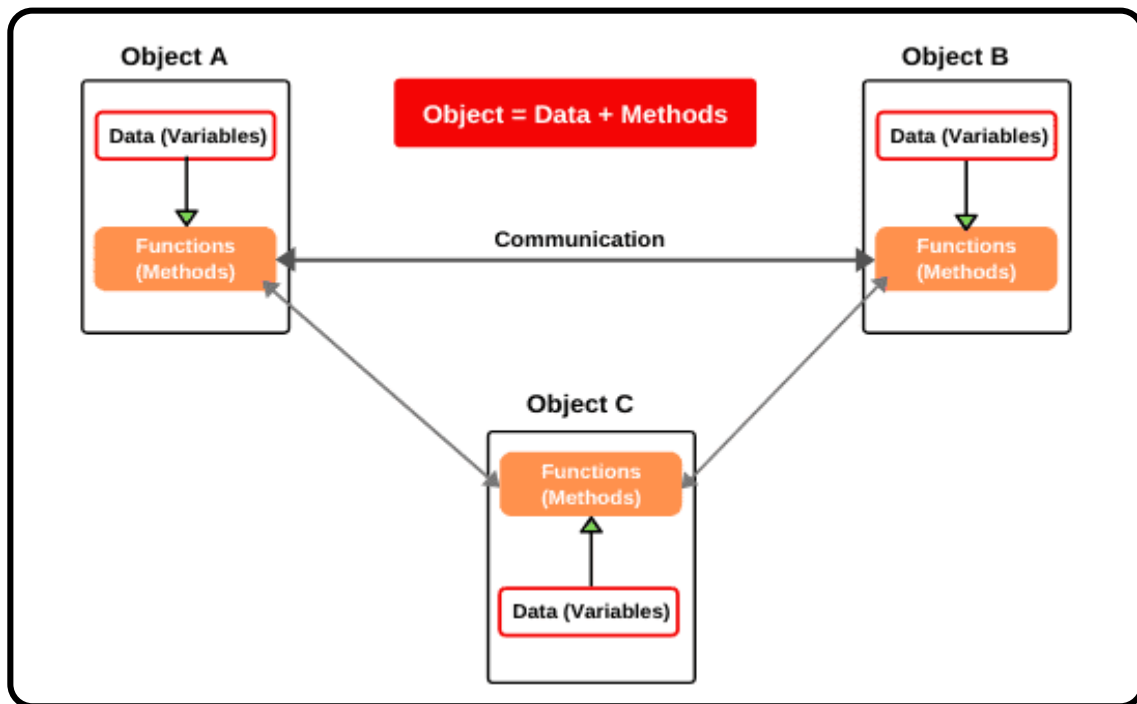
Introduction to Object Oriented Programming in Java

What is OOPs?

- ☐ Object-oriented programming (OOP) in Java is a programming methodology or paradigm (model) to design a computer program using **classes and objects**.
- ☐ It is the most widely programming approach used in the software industry today. It is an extension of procedural programming.
- ☐ Procedural programming means writing code without objects.
- ☐ Whereas, object-oriented programming means writing code with objects. It contains data in the form of fields or variables (often known as state or attributes or properties in java) and functionality code in the form of methods (often known as behaviour in java).
- ☐ Java OOP concept uses variables and methods as procedural programs do, but it focuses on the objects that contain variables and methods.
- ☐ That's why this new approach is called **object-oriented approach** in which we write programs using classes and objects.
- ☐ It comes with four main features like encapsulation, abstraction, inheritance, and polymorphism.
- ☐ When we write a program using these features, it is called **Object-Oriented Programming System (OOPs)**.
- ☐ The main goal of the OOPs concept in java programming is that everything you want to do, do through objects.

Object-Oriented Programming Paradigm (Model/Design)

- ☐ Object oriented programming is a new approach to overcome the drawbacks of a procedural oriented approach.
- ☐ It divides programs into the number of entities called objects that contain data (variables) and functions/tasks (known as methods in java).
- ☐ In an application, objects communicate with each other, share data among them, and solve problems. The combination of data and functions (methods) in OOP makes an object that is shown in the below figure.



- ☐ The data of an object is tied more closely with a function that operates on it. In the OOPs concept, data cannot move freely around the system.
- ☐ The data of an object can be accessed and modified only by the function of that object. An external function cannot access the data of an object but the functions of one object can access the function of other objects.

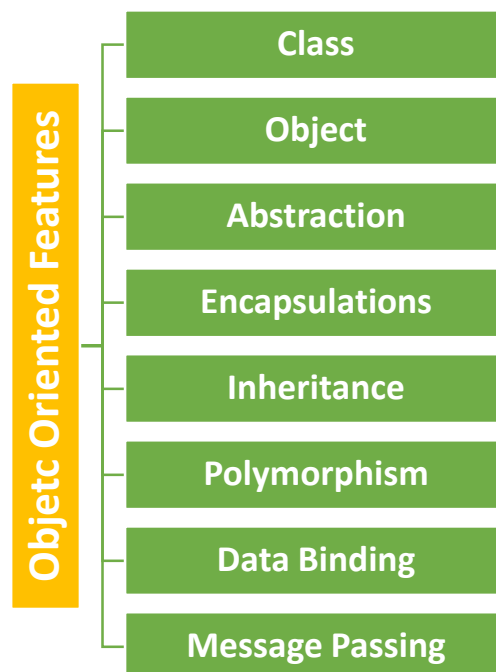
What is the difference between Procedural programming and OOPS?

Procedure Oriented Programming	Object Oriented Programming
A procedural language is based on functions	Object-oriented language is based on real-world objects.
Procedural language gives importance to the sequence of function execution	Object-oriented language gives importance on states and behaviours of the objects.
Procedural language exposes the data to the entire program	Object-oriented language encapsulates the data.
Procedural language follows a top-down programming paradigm	Object-oriented language follows a bottom-up programming paradigm.
A procedural language is complex in nature so it is difficult to modify, extend and maintain	Object-oriented language is less complex in nature so it is easier to modify, extend and maintain.
Procedural language provides less scope of code reuse	Object-oriented language provides more scope of code reuse.

Some important features or points to remember when we develop application using object oriented programming in java.

1. Higher priority is focused on Data rather than functions.
2. Programs are divided into the number of entities known as objects.
3. Objects communicate with each other through functions (methods).
4. Methods that operate on Data of an object are closely bound together in the data structure.
5. An object is a group of data and methods.
6. Data is hidden in the OOP and cannot be accessed by external methods. Hence, it is safe from accidental modification.
7. Data cannot move freely out of the object.
8. New data and methods can be easily added whenever needs.
9. A bottom-up approach is adopted in programming design.

Object Oriented Programming Features or Properties:



1. Class:

A class is basically user-defined data types that act as a template for creating objects of the identical type. It represents the common properties and actions (functions) of an object.

A class contains a skeleton of the object and does not take any space in the memory. It means Class does not exist physically but an object exists physically.

2. Objects:

An object is a basic unit in object-oriented programing. An object contains data and methods or functions that operate on that data. Objects take up space in memory.

3. Abstraction:

Abstraction is a technique by which we can hide the unnecessary data that is not needed from a user and expose only that data is of interest to the user. It hides all unwanted data so that users can work only with the required data.

Abstraction is one of the fundamental principles of object-oriented programming that permits the user to use an object without knowing its internal details. It helps to reduce complexity by not including background details.

Example 1: when we are driving the car, first we start the engine by inserting a key. We are not aware of the process that goes on in the background for starting the engine.

Example 2:

A real-time example of abstraction is “sending SMS”. When you need to send SMS from your mobile, you only type the text and send the message. But you don’t know the internal processing of the message delivery.

4. Encapsulation:

The process of binding data (variables) and corresponding functions (methods) together into a single unit (called class) is called encapsulation in Java. It is one of the striking features to achieve data security in an object-oriented program.

By doing this, data is not easily accessible to the outside world. In OOP we achieve encapsulation by making data members as private and having public methods or functions to access these data members. Only member methods defined in a class will have access to the data.

5. Inheritance

Inheritance is one of the main pillars of the OOP concept. It provides a mechanism for the users to reuse the existing code within the new applications.

We can do this by acquiring the properties of one class object into another class of object. Hence Inheritance provides reusability of code.

As such we can design a new class by acquiring the properties and functionality of another class and in this process, we need not modify the functionality of the parent class. We only add new functionality to the class.

6. Polymorphism

Polymorphism means one thing in multiple form. Polymorphism is an important feature of OOP and is usually implemented as operator overloading or function overloading. Operator overloading is a process in which an operator behaves differently in different situations. Similarly, in function overloading, the same function behaves differently in different situations.

Note: Java does not support operator overloading concept and this is also one of the reasons Java is not a pure object-oriented programming language. It means Java can support only function overloading.

7. Dynamic Binding

Dynamic binding (also known as late binding) is a way of connecting one program to another that is to be executed whenever it is called at runtime.

It is associated with inheritance and polymorphism.

8. Message Passing

In OOP, objects communicate with each other through member methods of the class. To establish communication between two objects, the following steps take place:

1. Creating classes that declare variables and methods.
2. Creating objects of classes that already declared.
3. Calling methods through suitable data to establish communication between objects.

Message passing involves three elements: name of object, name of method, and information to be sent.

Example:

```
Employee.salary(name);
```

Here, Employee is the name of object, salary is the name of method and name is parameter that contains information.

Advantage/Benefit of OOPs Concepts in Java

1. **Security:** In OOP, Data is encapsulated with methods in the class so that data is protected and secured from accidental modification by other external non-member methods.
2. **Reusability:** Through inheritance, we can use the features of an existing class in a new class without repeating existing code that saves a lot of time for developers, and also increases productivity.
3. **Effective communication:** In OOP, objects can communicate via message passing technique that makes interface descriptions with outside systems much simpler.
4. **Developing complex software:** OOPs is the most suitable approach for developing complex software because it minimizes the complexity through the feature of inheritance.
5. **Easily upgraded:** Object-oriented system can be easily upgraded from small to large systems because OOP uses bottom-up approach.
6. **Easy partition of work:** It is easy to partition complicated work in a project based on objects.
7. **Maintenance:** The maintenance of object-oriented code is easier.
8. **Efficiency:** The concepts of OOP provide better efficiency and an easy development process.