

OBJECT ORIENTED PROGRAMMING USING JAVA

Paper Code: CSCS231

MODULE - I

Concepts of OOP: Introduction OOP, Procedural Vs Object Oriented Programming, Principles of OOP, Benefits and applications of OOPS

OOPs Concepts (Introduction)

OOP is said to be the most popular programming model among developers. It is well suited for programs that are large, complex, and actively updated or maintained. It makes the development and maintenance of software easy by providing major concepts such as abstraction, inheritance, polymorphism, and encapsulation. These four are also the four pillars of an object-oriented programming system.

OOPs, provide the ability to simulate real-world events much more effectively. We can provide the solution to real-world problems if we are using the Object-Oriented Programming language. OOPs, provide data hiding, whereas, in Procedure-oriented programming language, global data can be accessed from anywhere.

Object-Oriented Programming is a paradigm that provides many concepts, such as object and class, encapsulation, inheritance, data binding, polymorphism, abstraction, etc.

Components of Object-oriented Programming

Objects – Object is the entity that makes the classes to be implemented into the program. It makes the features, properties, and behaviors of the class to be implemented. **Example** – A car is an object that has property color, model, brand name, fuel type, etc, and behaviors like, it runs. So these properties and behavior make an object (CAR).

Classes – A class can be stated as the blueprint of the object. This states what an object can do. It contains all the features, attributes, and behavior of what the model can do. We can say that class contains the definition of the object. **Example** – Car color, engine type, etc. And with the definition, we can create any number of objects.

Methods – Methods are the attributes of the class which are defined for the specified behavior of the object. It can also modify the state of the class. **Example** –

Method to drive a car, It changes the state of the car from parking state to running state.

Instances – It is the members of the class who holds some values related to the **OBJECTS OF THE CLASS**.

To understand better about these components better, let's consider an example – Consider the same example of television.

So according to the Object-Oriented Programming approach, you will perform certain steps-

- Define object name as a Television.
- Define the properties of television like channel, volume, on and off, etc.
- Create a function that will handle the functionality of tree-like changing the channel adjusting volume, or other things.
- Now implement all features into a program.

Now when you define an object as Television then it is the **class**. And then you declare the properties of that Television -like channel, volume, etc is the **instances** of the class Television. You create functions to change the properties, which are the **methods** of Television. And finally when you implement this in your program then you will create an **object** of class Television to implement into the program.

Programmatically it should be like –

```
class Television{
    int channel;    // Member variable
    int volume;    // Member variable
    boolean on = false; // Member variable
    .
    .
    void changeChannel(){ } //Methods
    void adjustVolume(){ } //Methods
    void switchOnOff(){ } //Methods
    .
    .
}
```



Instance

```
Television tv = new Television(); //Television Object
```

Explanation – In the above snippet of code. Defined Television is a class that contains the instances that are the property of the class. And methods that help to change the behavior of the television class.

And we are creating one object of television called tv, which will have all the properties and behavior of the class.

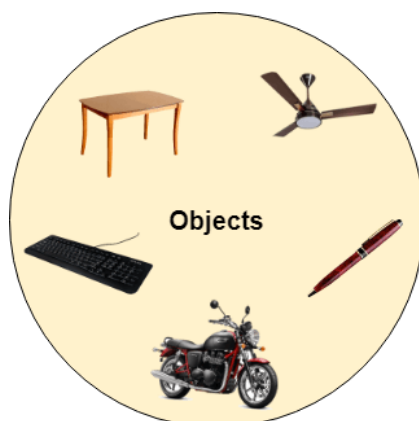
Procedural vs. Object Oriented Programming

	Procedural Oriented Programming	Object-Oriented Programming
1	In procedural programming, the program is divided into small parts called <i>functions</i> .	In object-oriented programming, the program is divided into small parts called <i>objects</i> .
2	Procedural programming follows a <i>top-down approach</i> .	Object-oriented programming follows a <i>bottom-up approach</i> .
3	There is no access specifier in procedural programming.	Object-oriented programming has access specifiers like private, public, protected, etc.
4	Adding new data and functions is not easy.	Adding new data and function is easy.
5	Procedural programming does not have any proper way of hiding data so it is <i>less secure</i> .	Object-oriented programming provides data hiding so it is <i>more secure</i> .
6	In procedural programming, overloading is not possible.	Overloading is possible in object-oriented programming.
7	In procedural programming, there is no concept of data hiding and inheritance.	In object-oriented programming, the concept of data hiding and inheritance is used.
8	In procedural programming, the function is more important than the data.	In object-oriented programming, data is more important than function.
9	Procedural programming is based on the <i>unreal world</i> .	Object-oriented programming is based on the <i>real world</i> .
10	Procedural programming is used for designing medium-sized programs.	Object-oriented programming is used for designing large and complex programs.
11	Procedural programming uses the concept of procedure abstraction.	Object-oriented programming uses the concept of data abstraction.
12	Code reusability absent in procedural programming,	Code reusability present in object-oriented programming.
13	Examples: C, FORTRAN, Pascal, Basic, etc.	Examples: C++, Java, Python, C#, etc.

Principles of OOPs



Object : Any entity that has state and behavior is known as an object. For example, a chair, pen, table, keyboard, bike, etc. It can be physical or logical. Object-Oriented Programming is a methodology or paradigm to design a program using classes and objects.



An Object can be defined as an instance of a class. An object contains an address and takes up some space in memory. Objects can communicate without knowing the details of each other's data or code. The only necessary thing is the type of message accepted and the type of response returned by the objects.

Class : It is a keyword in Java, through this class we can describe object in program. A class can also be defined as a blueprint from which you can create an individual object. Class doesn't consume any space.

Inheritance: *When one object acquires all the properties and behaviors of a parent object*, it is known as inheritance. It provides code reusability. It is used to achieve runtime polymorphism.

Polymorphism : *If one task is performed in different ways*, it is known as polymorphism. For example: to convince the customer differently, to draw something, for example, shape, triangle, rectangle, etc.

In Java, we use method overloading and method overriding to achieve polymorphism.

Abstraction: *Hiding internal details and showing functionality* is known as abstraction. For example phone call, we don't know the internal processing.

Encapsulation : It can be defined as the binding of data and attributes or methods and data members in a single unit. In classes, we have Data and attributes that perform operations on that data. So according to the OOPs principle of Encapsulation, that data can be merged into a single unit. Encapsulation enhances more security of the data as everything related to a single task must be grouped and access to the data is given as per need.

And this can be achieved using the concept of Data Hiding.

Encapsulation = Data Hiding + Abstraction.

Data Hiding – It means hiding the data of the class and restricting access to the outside world. Example – Using the access specifier keywords like private that restricts the data to only accessible and modifiable in the same class. Outside users can not access the data.

Let's understand with the help of an example-

Example 1- In the market, Capsules are available to cure different medical problems. Consider capsule for fever. Now in that capsules, there are different compositions are grouped to make a complete medicine that cures fever. So the grouping of that compositions into a single unit in the capsule is a form of encapsulation. Here, consider fever as data, and the capsule as the operations on the data. So everything is grouped here.

Explanation-

In the above image, all the dots represent the compositions for curing the fever (as per the example). And these are grouped into a capsule. Similarly in programming, when there are multiple operations related to a particular data are present, then grouping that into a simple unit is Encapsulation.

Benefits/Advantages of Object-Oriented Programming

1. Modularity for easier troubleshooting

When working with object-oriented programming languages, you know exactly where to look when something goes wrong. “Oh, the car object broke down? The problem must be in the Car class!” You don’t have to go line-by-line through all your code.

That’s the beauty of encapsulation. Objects are self-contained, and each bit of functionality does its own thing while leaving the other bits alone. Also, this modularity allows an IT team to work on multiple objects simultaneously while minimizing the chance that one person might duplicate someone else’s functionality.

2. Reuse of code through inheritance

The existing classes properties can be acquired by subclasses through the concept of inheritance.

- Reusability of Code adds essence to the Object Oriented Paradigm.
- Several types of Inheritances like single level, multilevel , hierarchical etc., help in reusability of code.

3. Flexibility through polymorphism

The flexibility is offered through the concept of Polymorphism. A scenario can be considered for a better understanding of the concept. A person can behave differently whenever the surroundings change. For example, if the person is in a market, the person will behave like a customer, or the behavior might get changed to a student when the person is in a school or any institution.

In this example, it can be observed that different behaviors are shown by the same person whenever the surroundings around the person get changed. This could explain the concept of Polymorphism and its flexibility. The developers benefit through Polymorphism in the following ways: simplicity and extensibility. Polymorphism is one of the benefits of oop as it gives scope to a code to be in more than one form.

4. Security

Object Oriented Programming provides security.

- The concept of Encapsulation bundles variables and methods into a component.
- The code can be protected from further modification, enhancement if required.

For example:

final keyword is used for not letting the class to be used for inheritance.

5. Easy Upgradation of Code

- The existing code can be modified to the changing needs easily.
- The reusability concept makes to easily upgrade the code by using the existing code.
- Polymorphism allows the programmer to implement the same method in different ways different parameters.

APPLICATIONS OF OBJECT ORIENTED PROGRAMMING

1. Client-Server Systems

Object-oriented client-server systems provide the IT infrastructure, creating Object-Oriented Client-Server Internet (OCSI) applications. Here, infrastructure refers to operating systems, networks, and hardware. OSCI consist of three major technologies:

- The Client Server
- Object-Oriented Programming
- The Internet

3. Object-Oriented Databases

These databases try to maintain a direct correspondence between the real-world and database objects in order to let the object retain its identity and integrity. They can then be identified and operated upon.

4. Real-Time System Design

Real-time systems inherent complexities that make it difficult to build them. Object-oriented techniques make it easier to handle those complexities. These techniques present ways of dealing with these complexities by providing an integrated framework, which includes schedulability analysis and behavioral specifications.

5. Simulation and Modeling System

It's difficult to model complex systems due to the varying specification of variables. These are prevalent in medicine and in other areas of natural science, such as ecology, zoology, and agronomic systems. Simulating complex systems requires modeling and understanding interactions explicitly. Object-oriented programming provides an alternative approach for simplifying these complex modeling systems.

6. Hypertext and Hypermedia

OOP also helps in laying out a framework for hypertext. Basically, hypertext is similar to regular text, as it can be stored, searched, and edited easily. The only difference is that hypertext is text with pointers to other text as well.

Hypermedia, on the other hand, is a superset of hypertext. Documents having hypermedia not only contain links to other pieces of text and information but also to numerous other forms of media, ranging from images to sound.

7. Neural Networking and Parallel Programming

It addresses the problem of prediction and approximation of complex time-varying systems. Firstly, the entire time-varying process is split into several time intervals or slots. Then, neural networks are developed in a particular time interval to disperse the load of various networks. OOP simplifies the entire process by simplifying the approximation and prediction ability of networks.

8. Office Automation Systems

These include formal as well as informal electronic systems primarily concerned with information sharing and communication to and from people inside and outside the organization. Some examples are:

- Email
- Word processing
- Web calendars
- Desktop publishing

9. CIM/CAD/CAM Systems

OOP can also be used in manufacturing and design applications, as it allows people to reduce the effort involved. For instance, it can be used while designing blueprints and flowcharts. OOP makes it possible for the designers and engineers to produce these flowcharts and blueprints accurately.

10. AI Expert Systems

These are computer applications that are developed to solve complex problems pertaining to a specific domain, which is at a level far beyond the reach of a human brain.

It has the following characteristics:

- Reliable
- Highly responsive
- Understandable
- High-performance

Other Applications

- Desktop applications
- Web applications
- CAD/CAM Software
- Web servers and application servers
- Mobile applications (particularly Android based)
- Games
- Graphical User Interface Design
- Big Data applications
- Parallel Programming
- Machine Learning Applications
- Artificial Intelligence
- Expert Systems
- Real Time Systems
- e-Business Applications
- Database connection
- Object Oriented Database