



**SRINIVAS UNIVERSITY
INSTITUTE OF ENGINEERING AND
TECHNOLOGY
MUKKA, MANGALURU**

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NOTES

OBJECT ORIENTED CONCEPTS

SUBJECT CODE: 19SCS43

COMPILED BY:

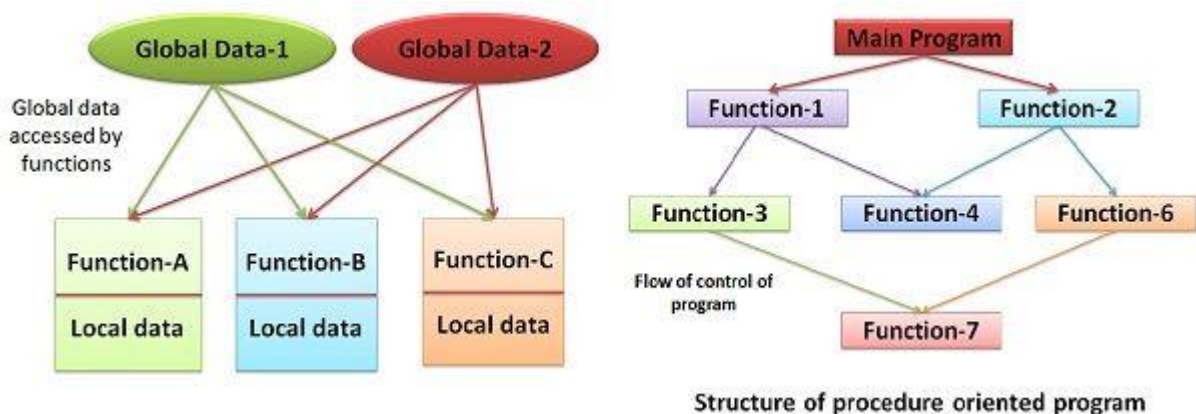
Mr. AKHILRAJ V. GADAGKAR
Assistant Professor

MODULE 1

INTRODUCTION TO OBJECT ORIENTED CONCEPTS

Procedure oriented programming (POP)

POP is a conventional way of programming. Procedural programming is where the primary focus is on getting the task done in sequential order. Flowchart organizes the flow of control of the program. If the program is extensive, it is structured in some small units called functions, which shares global data. Here, the concern of data security arises, as there is an unintentional change in the program by functions.



POP characteristics

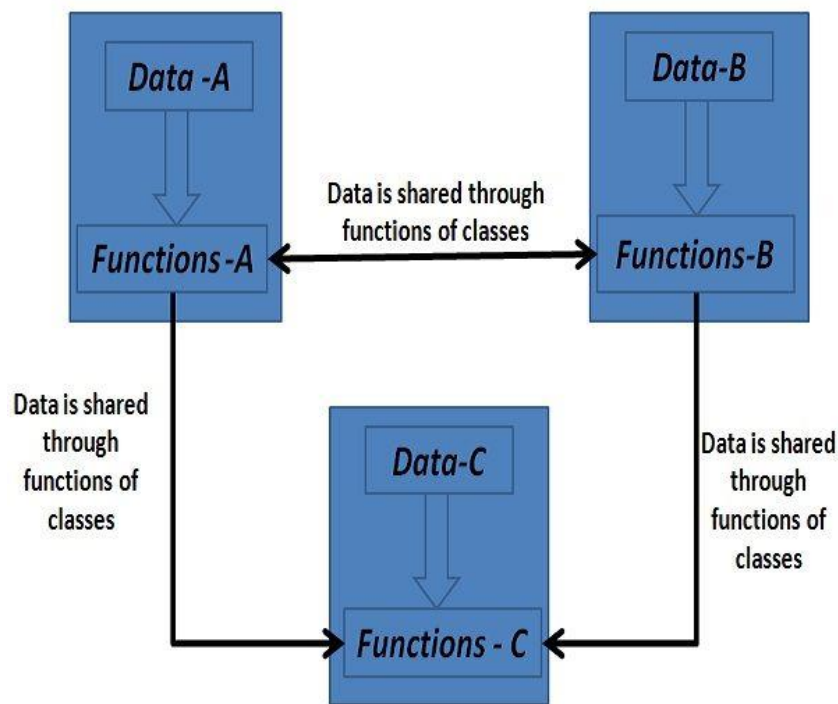
- While designing a program, POP follows a top-down programming approach.
- Majority of the functions allows global data to be shared.
- It also divides larger programs into smaller parts called as functions.
- It allows a free data movement around the system from functions to functions.
- Data is transformed by the functions from one form to another.
- It gives importance to the concept of the functions.

Object-Oriented Programming (OOP)

The object-oriented programming is basically a computer programming design philosophy or methodology that organizes/ models software design around data, or objects rather than functions and logic.

OOP's chief concern is to hide the data from non-member functions of a class, which it treats like "critical information". Data is closely tied to the member functions of a class, which operates on it. It doesn't allow any non-member function to modify the data inside it. Objects interact with each other through member functions to access their data.

An object is referred to as a data field that has unique attributes and behavior. Everything in OOP is grouped as self-sustainable objects.



Data flow in object-oriented programming

OOP is the most popular programming model among developers. It is well suited for programs that are large, complex, and actively updated or maintained. It simplifies software development and maintenance by providing major concepts such as abstraction, inheritance, polymorphism, and encapsulation. These core concepts support OOP.

OOP characteristics

- Everything is an object
- Developer manipulates objects that uses message passing.
- Every object is an instance of a class.
- The class contains the attribute and behaviour associated with an object.

Advantage of OOPs over Procedure-oriented programming language

1. OOPs makes development and maintenance easier, whereas, in a procedure-oriented programming language, it is not easy to manage if code grows as project size increases.
2. OOPs provides data hiding, whereas, in a procedure-oriented programming language, global data can be accessed from anywhere.
3. OOPs provides the ability to simulate real-world event much more effectively. We can provide the solution of real word problem if we are using the Object-Oriented Programming language.

Benefits of POP (Procedure Oriented Programming)

- Provides an ability to reuse the same code at various places.
- Facilitates in tracking the program flow.
- Capable of constructing modules.

Limitations of POP (Procedure Oriented Programming)

- Global data are vulnerable.
- Data can move freely within a program
- It is tough to verify the data position.
- Functions are action-oriented.
- Functions are not capable of relating to the elements of the problem.
- Real-world problems cannot be modelled.
- Parts of code are interdependent.
- One application code cannot be used in other application.
- Data is transferred by using the functions.

Benefits of OOP (Object Oriented Programming)

- Modular, scalable, extensible, reusable, and maintainable.
- It models the complex problem in a simple structure.
- Object can be used across the program.
- Code can be reused.
- We can easily modify, append code without affecting the other code blocks.
- Provides security through encapsulation and data hiding features.
- Beneficial to collaborative development in which a large project is divided into groups.
- Debugging is easy.

Limitations of OOP (Object Oriented Programming)

- Requires intensive testing processes.
- Solving problems takes more time as compared to Procedure Oriented Programming.
- The size of the programs created using this approach may become larger than the programs written using the procedure-oriented programming approach.
- Software developed using this approach requires a substantial amount of pre-work and planning.
- OOP code is difficult to understand if you do not have the corresponding class documentation.
- In certain scenarios, these programs can consume a large amount of memory.
- Not suitable for small problems.
- Takes more time to solve problems.

List of Object-Oriented Programming Languages

There are various object-oriented programming languages are present. But we have enlisted some popular and widely used OOP languages. According to the **TIOBE index**, the top twenty OOP languages are Java, C++, C#, Python, R, PHP, Visual Basic.NET, JavaScript, Ruby, Perl, Object Pascal, Objective-C, Dart, Swift, Scala, Kotlin, Common Lisp, MATLAB, and Smalltalk.

Applications of OOPs

- Computer graphics applications
- Object-oriented database
- User-interface design such as windows
- Real-time systems
- Simulation and modelling
- Client-Server System
- Artificial Intelligence System
- CAD/CAM Software
- Office automation system

Comparison / Differentiation between POP and OOP

BASIS FOR COMPARISON	Procedure Oriented Programming (POP)	Object Oriented Programming (OOP)
Basic	Procedure/Structure oriented .	Object oriented.
Approach	Top-down.	Bottom-up.
Basis	Main focus is on "how to get the task done" i.e. on the procedure or structure of a program .	Main focus is on 'data security'. Hence, only objects are permitted to access the entities of a class.
Division	Large program is divided into units called functions.	Entire program is divided into objects.
Entity accessing mode	No access specifier available.	Access specifier are "public", "private", "protected".
Inheritance	There is no provision of inheritance.	Inheritance achieved in three modes public private and protected.
Data hiding & security	There is no proper way of hiding the data, so data is insecure	Data is hidden in three modes public, private, and protected. hence data security increases.
Data sharing	Global data is shared among the functions in the program.	Data is shared among the objects through the member functions.
Overloading or Polymorphism	Neither it overload functions nor operators.	It overloads functions, constructors, and operators.
Example	C, Visual Basic (VB), FORTRAN, Pascal	C++, JAVA, Python, C#, Visual Basic.NET.

Principles of OOP

The following are the principles of OOP, that a object oriented programming language should support:

1. Class
2. Objects
3. Data Abstraction
4. Encapsulation
5. Inheritance
6. Polymorphism
7. Dynamic Binding
8. Message Passing

1. Class:

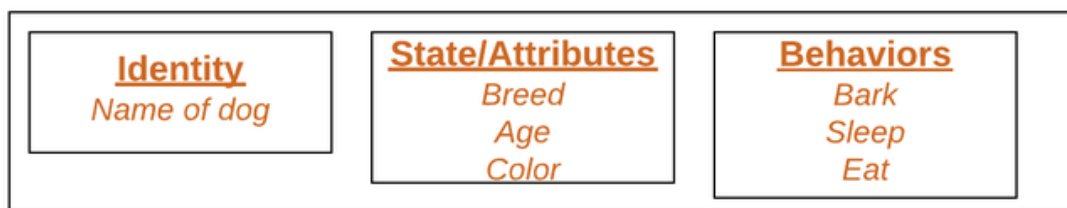
A class is a user-defined data type. It consists of data members and member functions, which can be accessed and used by creating an instance of that class. It represents the set of properties or methods that are common to all objects of one type. A class is like a blueprint for an object.

For Example: Consider the Class of Cars. There may be many cars with different names and brands but all of them will share some common properties like all of them will have 4 wheels, Speed Limit, Mileage range, etc. So here, Car is the class, and wheels, speed limits, mileage are their properties.

2. Object:

It is a basic unit of Object-Oriented Programming and represents the real-life entities. An Object is an instance of a Class. When a class is defined, no memory is allocated but when it is instantiated (i.e. an object is created) memory is allocated. An object has an identity, state, and behavior. Each object contains data and code to manipulate the data. Objects can interact without having to know details of each other's data or code, it is sufficient to know the type of message accepted and type of response returned by the objects.

For example "Dog" is a real-life Object, which has some characteristics like color, Breed, Bark, Sleep, and Eats.



3. Data Abstraction:

Data abstraction is one of the most essential and important features of object-oriented programming. Data abstraction refers to providing only essential information about the data to the outside world, hiding the background details or implementation.

Consider a real-life example of a man driving a car. The man only knows that pressing the accelerators will increase the speed of the car or applying brakes will stop the car, but he does not know about how on pressing the accelerator the speed is increasing, he does not know about the inner mechanism of the car or the implementation of the accelerator, brakes, etc in the car. This is what abstraction is.

4. Encapsulation:

Encapsulation is defined as the wrapping up of data under a single unit. It is the mechanism that binds together code and the data it manipulates. In Encapsulation, the variables or data of a class are hidden from any other class and can be accessed only through any member function of their class in which they are declared. As in encapsulation, the data in a class is hidden from other classes, so it is also known as data-hiding.

5. Inheritance:

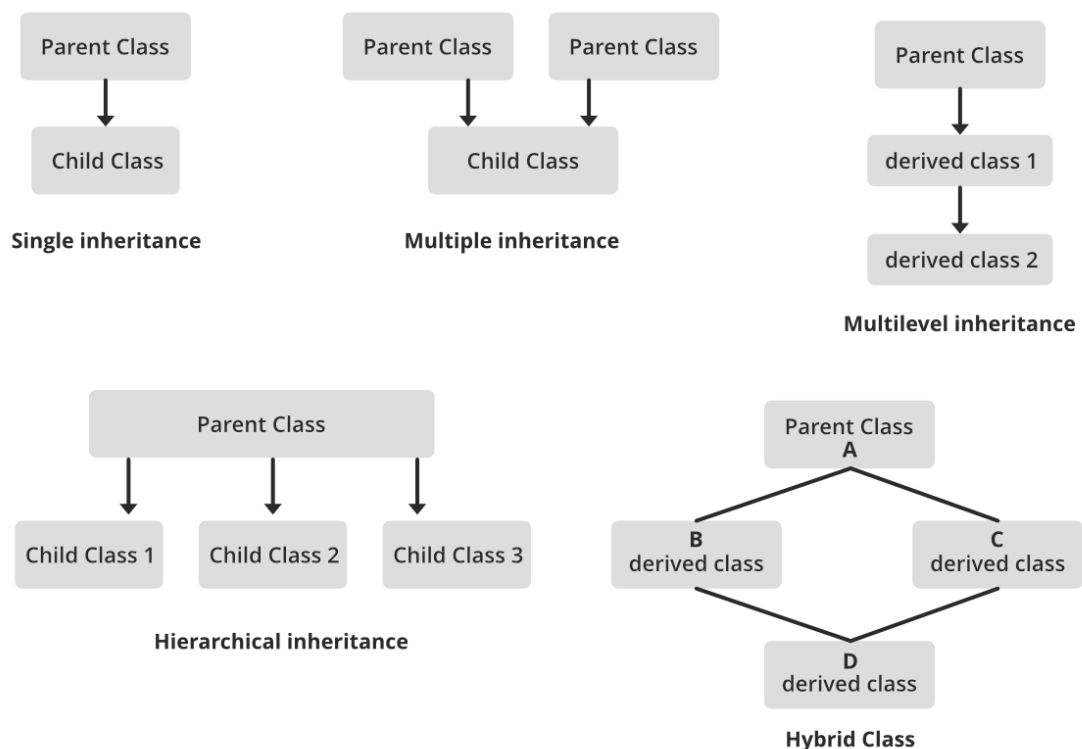
Inheritance is an important pillar of OOP (Object-Oriented Programming). The capability of a class to derive properties and characteristics from another class is called Inheritance. When we write a class, we inherit properties from other classes. So when we create a class, we do not need to write all the properties and functions again and again, as these can be inherited from another class that possesses it. Inheritance allows the user to reuse the code whenever possible and reduce its redundancy. **(Refer to the figure in next page, for types of inheritance)**

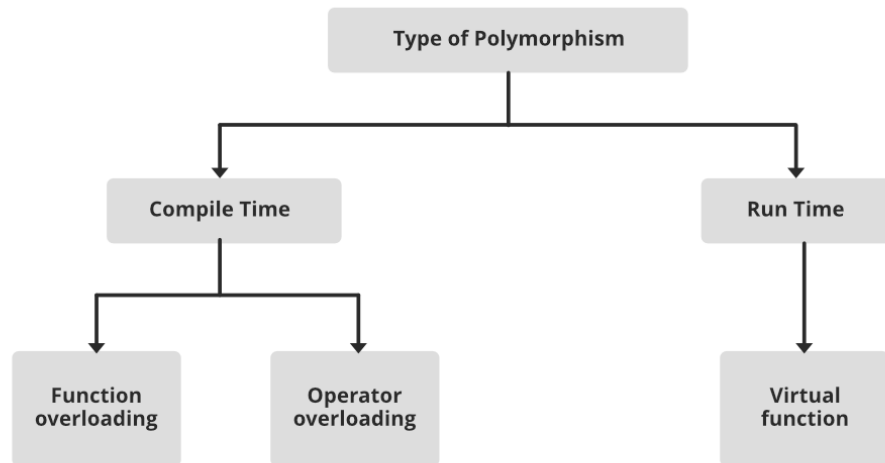
6. Polymorphism:

The word polymorphism means having many forms. In simple words, we can define polymorphism as the ability of a message to be displayed in more than one form. For example, A person at the same time can have different characteristics. Like a man at the same time is a father, a husband, an employee. So the same person possesses different behaviour in different situations. This is called polymorphism. **(Refer to the figure, for types of polymorphism)**

7. Dynamic Binding:

In dynamic binding, the code to be executed in response to the function call is decided at runtime. Dynamic binding means that the code associated with a given procedure call is not known until the time of the call at run time. Dynamic Method Binding One of the main advantages of inheritance is that some derived class D has all the members of its base class B. Once D is not hiding any of the public members of B, then an object of D can represent B in any context where a B could be used. This feature is known as subtype polymorphism.



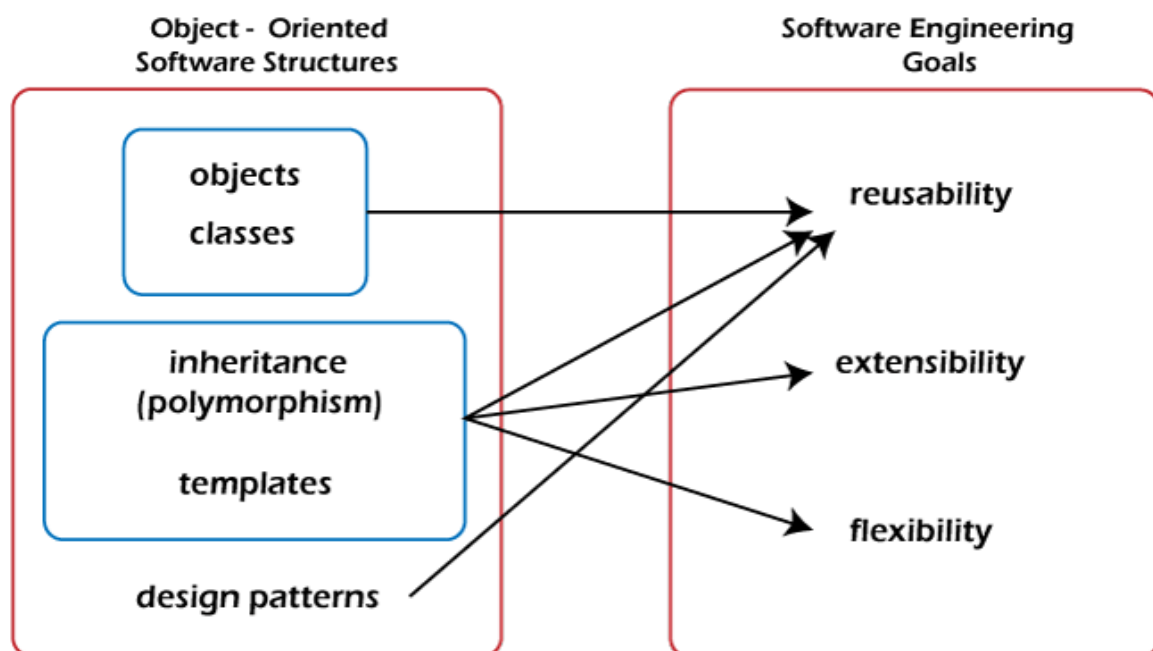


8. Message Passing:

It is a form of communication used in object-oriented programming as well as parallel programming. Objects communicate with one another by sending and receiving information to each other. A message for an object is a request for execution of a procedure and therefore will invoke a function in the receiving object that generates the desired results. Message passing involves specifying the name of the object, the name of the function, and the information to be sent.

Why should OOP be used?

Object-oriented programming is an evolutionary development in software engineering. Using OOP in software development is a good habit because it accomplishes the three major software engineering goals, as we have shown in the following figure.



Where is it used?

OOP is often the best use when we are dealing with manufacturing and designing applications. It provides modularity in programming. It allows us to break down the software into chunks of small problems that we then can solve one object at a time.

It should be used where the reusability of code and maintenance is a major concern. Because it makes development easy and we can easily append code without affecting other code blocks. It should be used where complex programming is a challenge.

What is the difference between an object-oriented programming language and object-based programming language?

Object-based programming language is one that supports the first four OOP principles (class, objects, data abstraction, and encapsulation) sometimes support polymorphism, but not inheritance are termed as object-based languages. JavaScript and VBScript are examples of object-based programming languages.