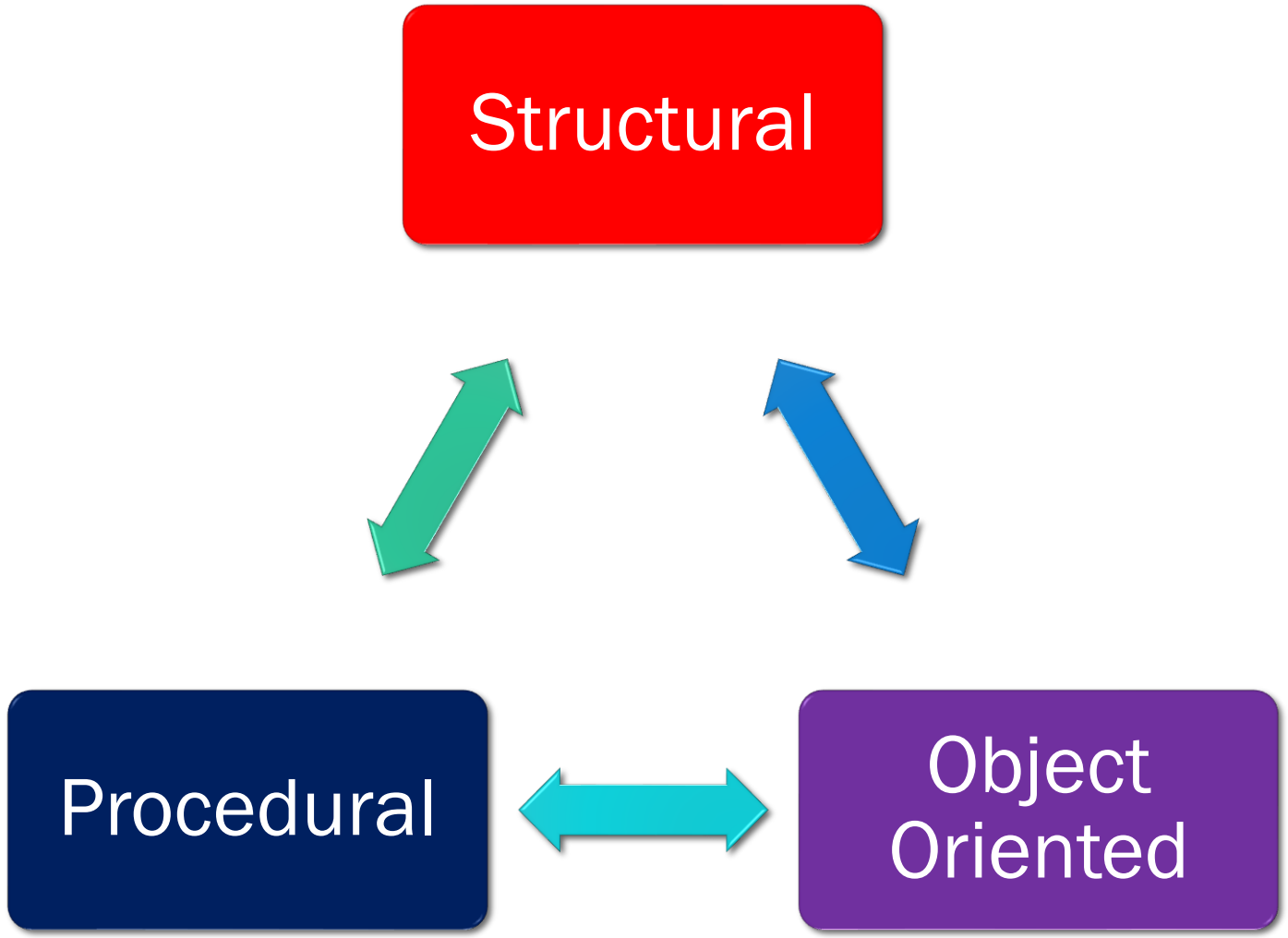
Unit 5

OBJECT ORIENTED PROGRAMMING (OOP)

Programming Techniques



Procedural Programming

- A procedural programming language consists of a set of method and a set of code for each method
- The program executes statement by statement, reading and modifying a shared memory
- It executes set of program sequentially in order
- These languages are based on procedure calls
- The statements are organized into procedures
- Procedures are also known as subroutines or functions

Procedural Programming

- Procedures are written in a step by step manner so that it is very easy to follow the smaller programs
- Procedural programming works fine for simple & smaller program
- We need structural and OOP concept for larger and complex program
- Examples of Procedural Programming Languages are PASCAL, FORTRAN, COBOL, C programming

Structural Programming

- A structured programming language is also known as modular programming
- It is a programming model that uses the reusable codes
- It emphasizes on separating a program's data from its functionality
- The system is further divided into compositional subsystem
- It focuses on dividing the problem into sub parts and hence simplifies the problem solving
- Large programs are divided into smaller programs so that they can be easily solved
- Examples: ALGOL, PASCAL

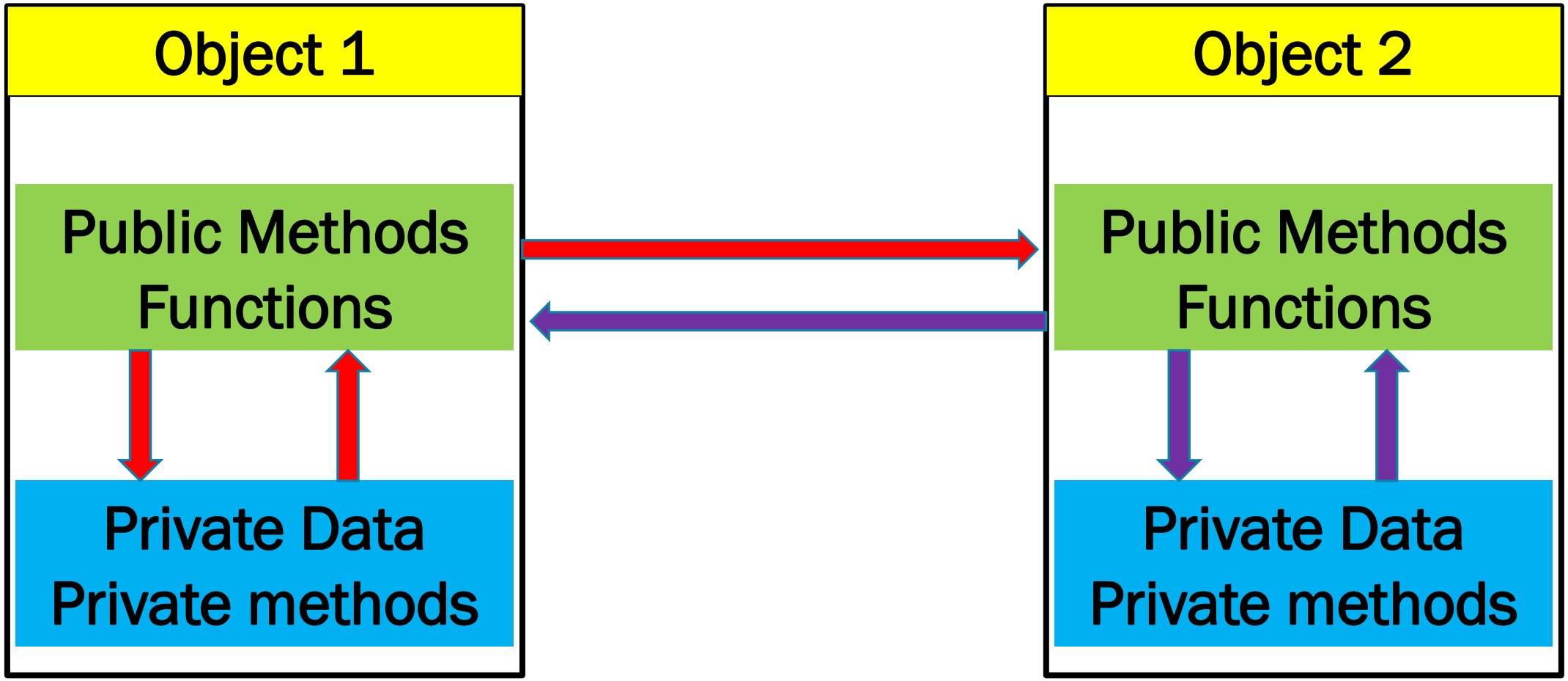
Object Oriented Programming (OOP)

- OOP attempts to make programs more closely as models the way people think about and deal with the world
- OOP approach was invented to overcome the drawbacks of structural and procedural oriented programming (POP)
- It uses bottom up approach
- OOP treats data as critical (very important) element in the program development and does not allow data to move freely around the program
- OOP ties the data more closely to the function

Object Oriented Programming (OOP)

- It protects the data from accidental modification from outside function
- OOP allows decomposition of a program into a number of entities called objects and then builds data and functions around these objects
- Data of an object can be accessed only by the function associated with the that function
- However function of one object can access the function of another object

Object Oriented Programming (OOP)



Benefits of OOP over POP

1. Eliminate Redundancy

   We can eliminate redundancy of code using Inheritance

2. Allows Developing Secure Program

   Data can be hidden and can only be accessed by authorized functions using Data Abstraction

3. Easy Job Division

   The problem can be divided and assigned easily between group members

4. Provide Extensibility

   OOP programs can be easily upgraded from small to large system

Benefits of OOP over POP

5. Reduce Complexity

   Software complexity can be easily managed and reduced

6. Saves Development time

   Modular programs can be developed and integrated fast

7. Increases Productivity

   Solution can be built by using standard modules instead of writing code from scratch

8. Allows Designing Simpler Interface

   Message passing techniques between objects allows making simpler interface descriptions with external systems

Features of OOP

- Class
- Object
- Inheritance
- Polymorphism
- Abstraction/Data Hiding
- Encapsulation
- Overloading
- Reusability

Features of OOP

Class

- A class is the collection of similar objects which is defined as the template or prototype to define the common attributes and behavior for all the objects of the class.
- Objects are variable of user defined data type class.
- We can create any number of objects associated with the class defined.
- For example Crow, Parrot, Pigeon etc are the objects of class Bird.
- Classes are also known as Abstract Data Types (ADT) in OOP.

Features of OOP

1. Class

- Class declaration is similar to structure declaration

Class Structure

```
class <class_name>{
private:
variable declaration;
function declaration;

public:
variable declaration;
function declaration;
}
```

Class Example

```
class player{
private:
int score;
int key;

public:
void setScore(int n);
int getScore();
}
```

Features of OOP

Object

- Object is the basic run time entities, things or organizations that exists in real world that consists of two fundamental characteristics: its attributes and behaviors.
- They may represent a person, a place, a vehicle etc.
- It is user defined data. OOP attributes are represented by data and behaviors are represented by the functions.
- An object can communicate with others by using message passing mechanism.

Features of OOP

Object

- Object is a variable consisting both routine/method and data that is treated as discrete entity within program
- Objects contains data and function to manipulate the data
- Different objects can interact by sending and receiving message through function to one another
- Examples: Book (class) => "Saya" (object)

  Birds (class) => "Parrot" (object)

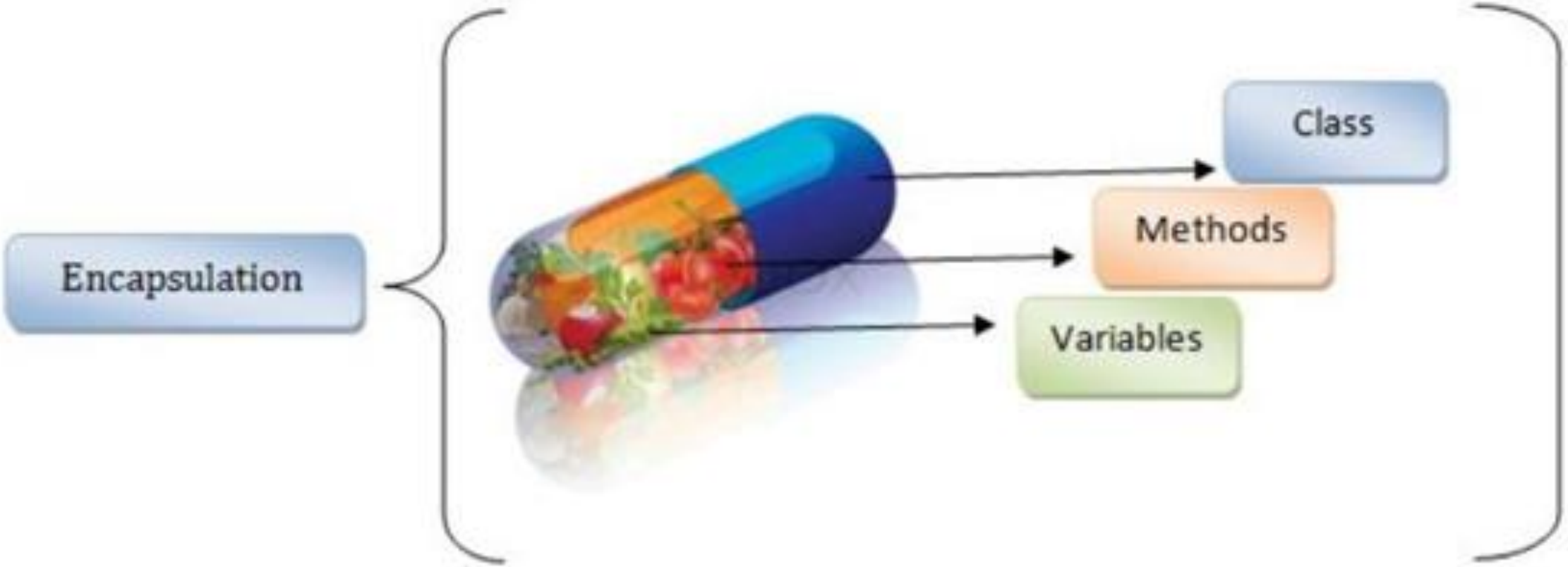
Features of OOP

Encapsulation

- The wrapping up of data and function into a single unit called class is known as encapsulation
- The data is hidden inside the object. It is not accessible to the outside world and only those functions which are wrapped in the class can access the data.
- It prevents the data from accidental or unnecessary modification
- Function provide interface between object data and program

Features of OOP

Encapsulation



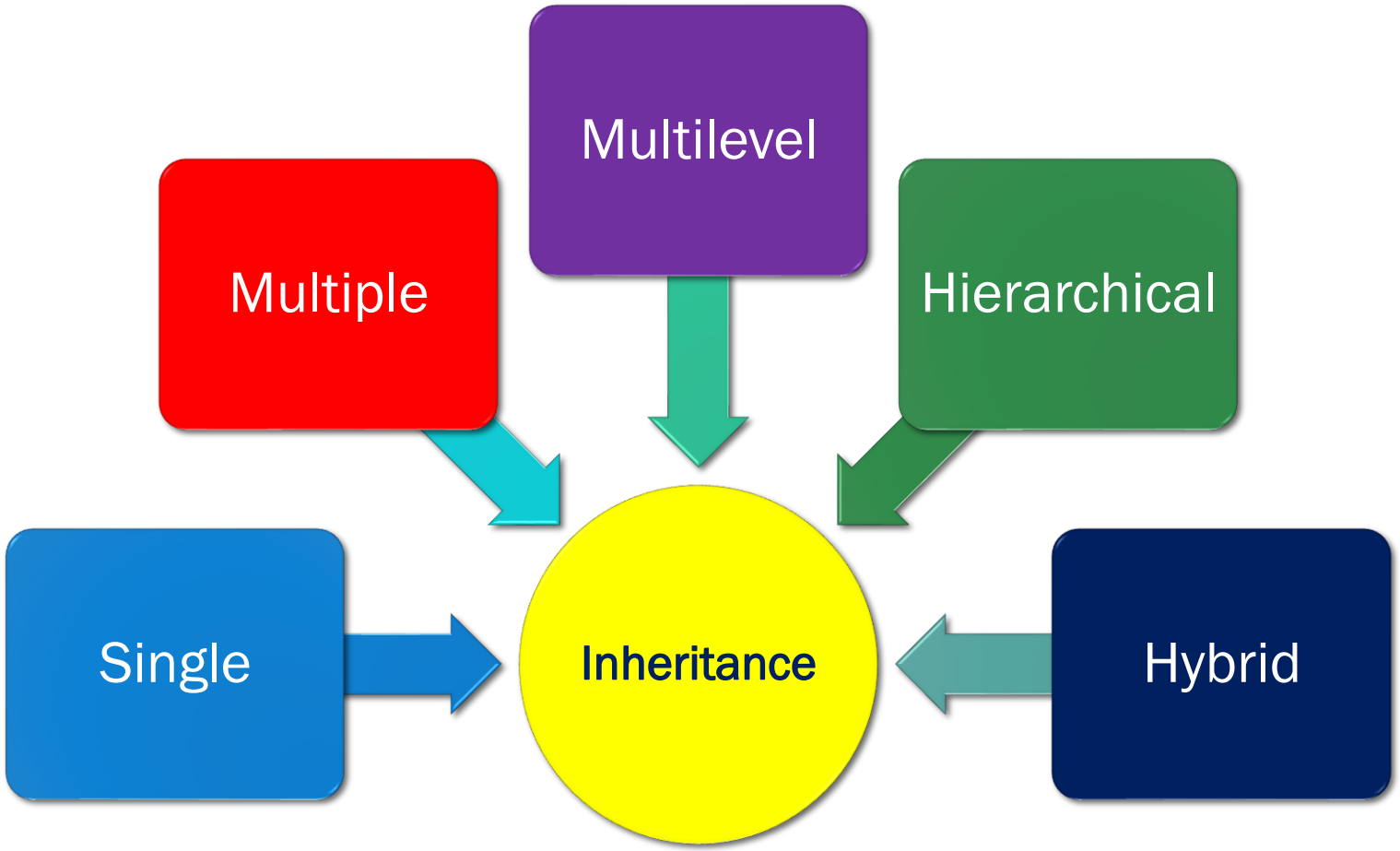
Features of OOP

Inheritance

- Inheritance is the process by which objects of one class acquires the similar properties of objects of another class from which they are derived.
- The concept of inheritance provides the idea of reusability
- We can add additional features to an existing class without modifying it.
- It is possible by deriving a new class from the existing one.
- New classes will have combined features of both the classes

Features of OOP

Inheritance

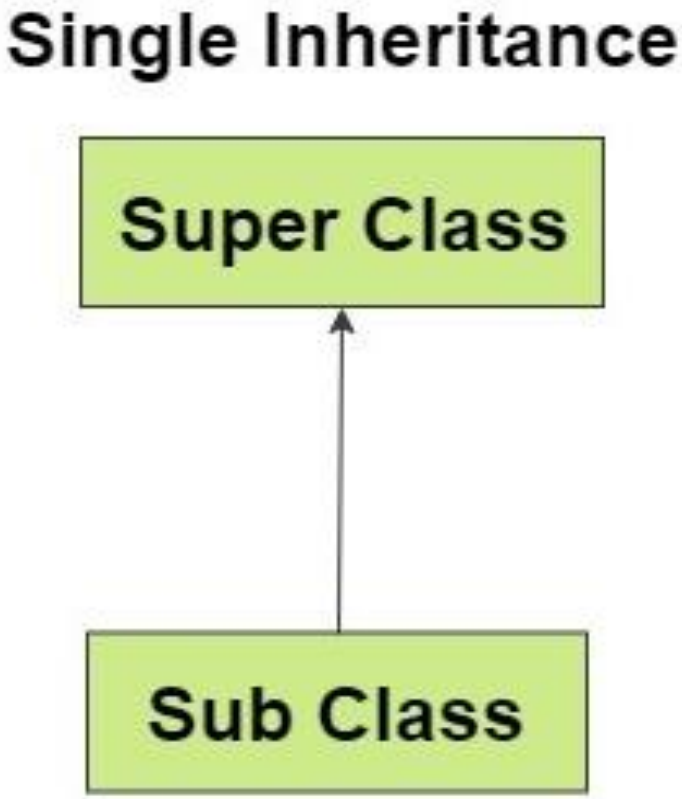


Features of OOP

Single Inheritance

The process of creating a new class from an existing class is called single inheritance.

In single inheritance, there is only one base class and only one derived class.

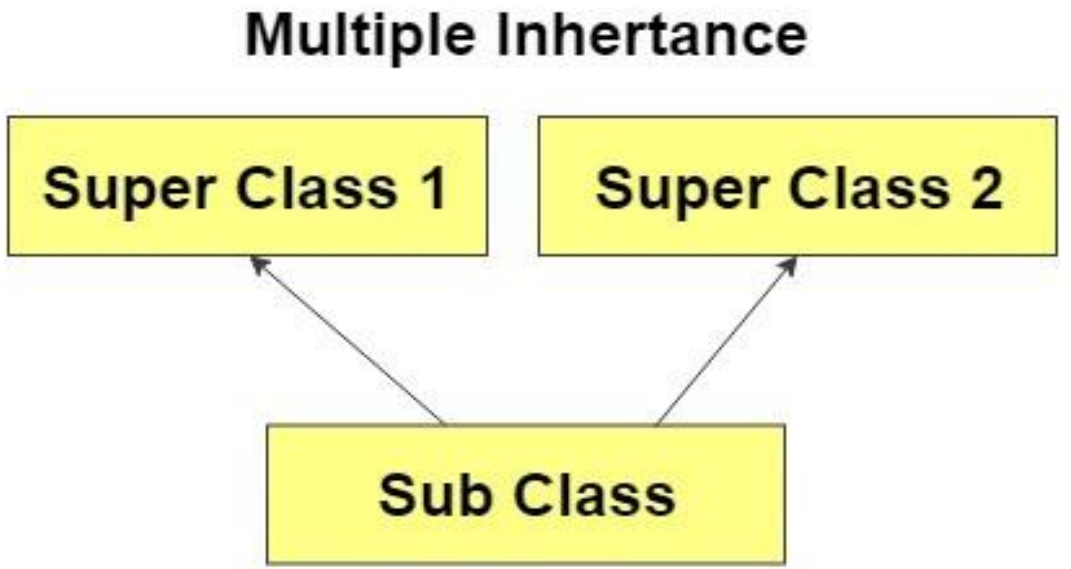


Features of OOP

Multiple Inheritance

The process of creating only one new class from several existing classes is called multiple inheritance.

In multiple inheritance, there is only one derived class and two or more base class.

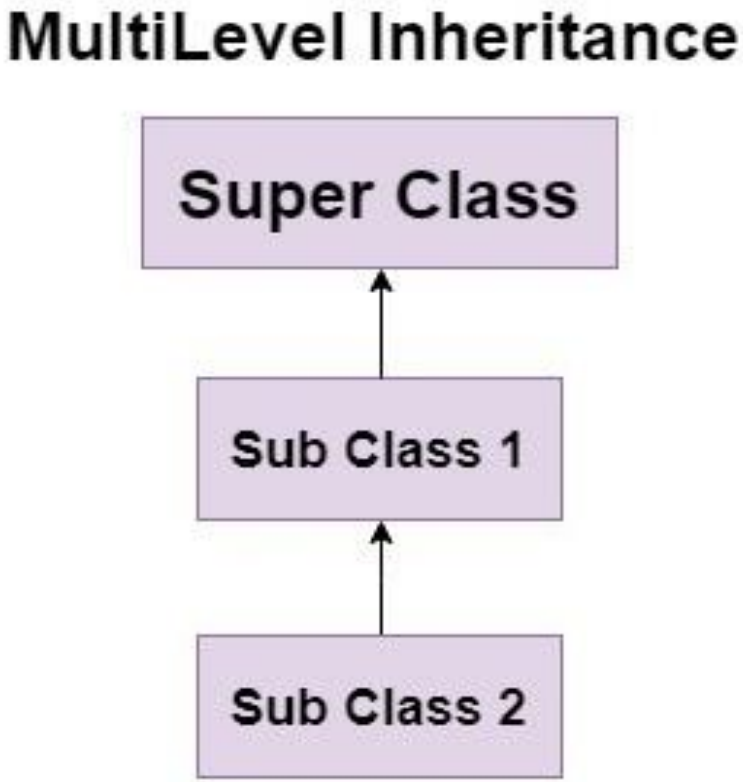


Features of OOP

Multilevel Inheritance

The process of creating a new class from another derived class is called multilevel inheritance.

A new class can be derived from the derived class derived from another base class.

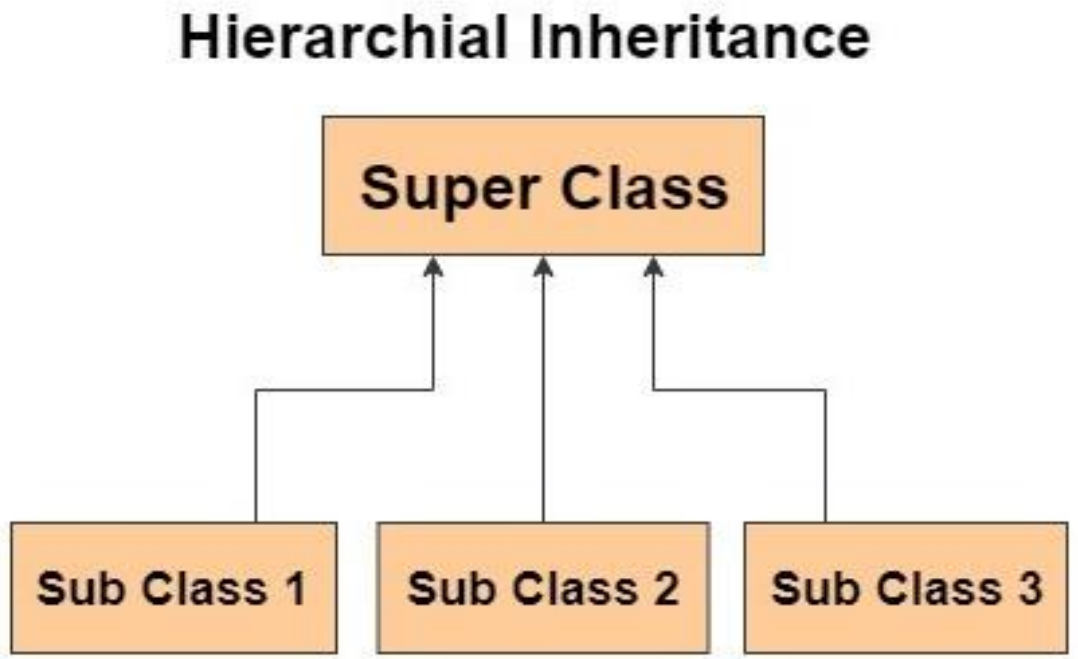


Features of OOP

Hierarchical Inheritance

The process of creating several classes from only one class is called hierarchical inheritance.

There are two or more derived classes an only one base class in hierarchical inheritance.



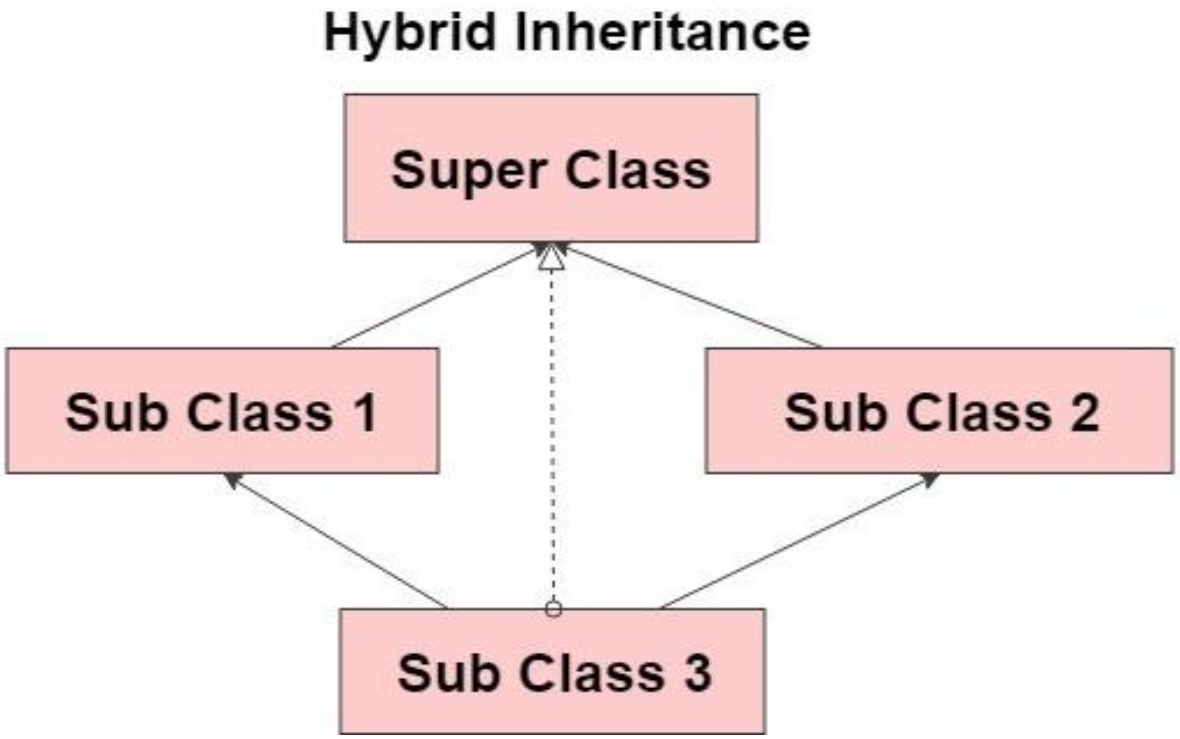
Features of OOP

Hybrid Inheritance

It is the combination of two or more types of inheritance.

So it may contain more than one type of inheritance.

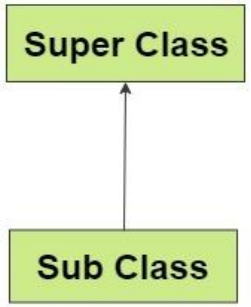
It is also known as Virtual Inheritance



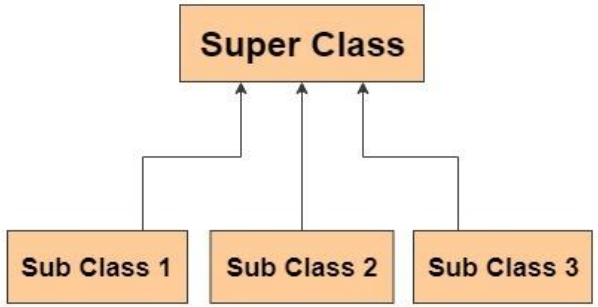
Features of OOP

Types of Inheritance

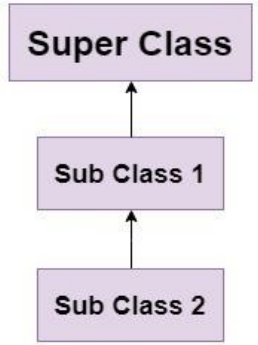
Single Inheritance



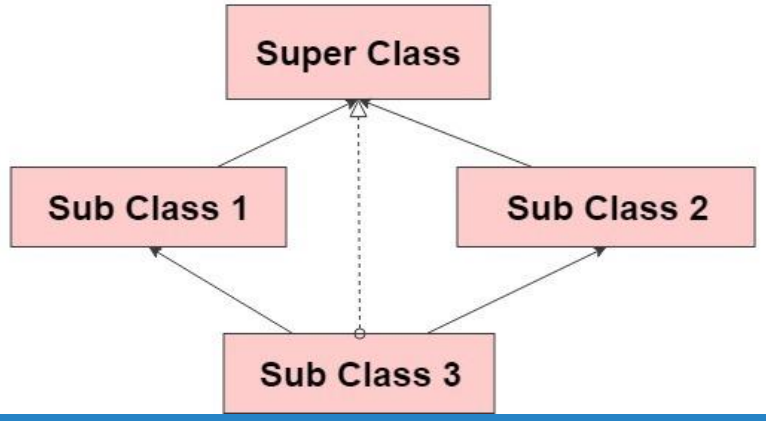
Hierarchial Inheritance



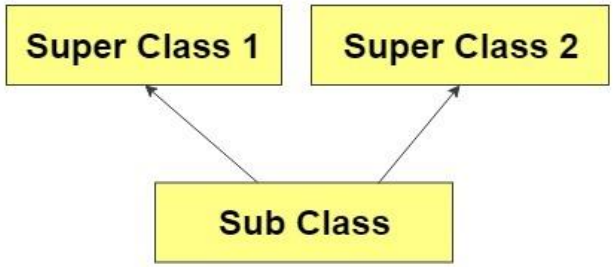
MultiLevel Inheritance



Hybrid Inheritance



Multiple Inhertance



Features of OOP

Polymorphism

- Polymorphism refers the ability of an object to take a different forms depending upon situations.
- The term Polymorphism is formed by the combination of two Greek words poly and morph which means many and forms respectively.
- It simply means one name many forms.
- It is and important feature of OOP which refers to the ability of an object to take on different forms depending upon situations.
- The behavior depends upon the types of data used in the operation.
- It simplifies coding and reduces the rework involved in modifying and developing applications.

Features of OOP

Polymorphism

There are two types of polymorphism:

1. Operator overloading
2. Function overloading

Features of OOP

1. Operator Overloading

- The process of making an operator to exhibit different behaviors in different instances is known as operator overloading.
- For example (int + int), (string + string)
- The '+' symbol for two integers will generate sum but for string it will generate a concatenated third string
- Eg. 20 + 30 = 50 "wel"+"come" = "welcome"

Features of OOP

2. Function Overloading

- The process of using a single function name to perform different types of tasks is known as function overloading.
- Function overloading is performed by making the number of arguments or type of arguments different.
- float calcArea(float r); //returns area of circle
- int calcArea(int l); //returns area of square
- int calcArea(int l, int b); //returns area of rectangle

Application of OOP

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.
- OCSI consist of three major technologies:
  - The Client Server
  - Object-Oriented Programming
  - The Internet

Application of OOP

2. Object-Oriented Databases

- They are also called Object Database Management Systems (ODBMS).
- These databases store objects instead of data, such as real numbers and integers.
- Objects consist of the following:
  - Attributes:
    - Attributes are data that defines the traits of an object. This data can be as simple as integers and real numbers. It can also be a reference to a complex object.
  - Methods:
    - They define the behavior and are also called functions or procedures.

Application of OOP

3. AI Expert Systems

- These are computer applications which 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

Application of OOP

4. Real-Time System Design

- Real time systems inherit complexities that makes 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 schedulable analysis and behavioral specifications.

Application of OOP

5. Simulation and Modelling 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 modelling and understanding interactions explicitly.
- Object-oriented Programming provides an alternative approach for simplifying these complex modelling systems.

Application of OOP

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.

Application of OOP

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.

Application of OOP

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 as well as outside the organization.
- Some examples are:
  - Email
  - Word processing
  - Web calendars
  - Desktop publishing

Application of OOP

9. 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, flowcharts, etc.
- OOP makes it possible for the designers and engineers to produce these flowcharts and blueprints accurately.

OOP Vs POP

SN	OOP	POP
1	Most recent programming concept with bottom-up approach	Old programming concept with top-down approach
2	Programs are divided into number of entities called objects	Programs are divided into number of functions
3	Emphasis is on data rather than function	Emphasis is on functions rather than on data
4	Does not allow data to move freely around program from one object to another	Allows data to move freely around the program from one function to another
5	Data is hidden inside the object and can not be accessed by external functions	data is open and can be freely accessed by all programs

OOP Vs POP

SN	OOP	POP
6	Problems are viewed as real world entity	Problems are not viewed as real world entity
7	OOP is written by using High Level Language (HLL) such as C++, Java, ASP.NET etc	POP is written by using High & Middle Level Language (HLL & MLL) such as C, Pascal, FORTRAN etc
8	Concept of encapsulation provides high level of security	Security can not be maintained due to lack of such feature
9	Easy to reuse existing program because of inheritance feature	No proper mechanism for reuse of existing code
10	New data and functions can be easily added whenever required	Adding new data and function is difficult and time consuming

Thank You

END OF UNIT 5