Object Oriented Programming Concepts

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

School Of Computer Engineering

Mr. Abhaya Kumar Sahoo
Assistant Professor [II]
School of Computer Engineering,
Kalinga Institute of Industrial Technology (KIIT),
Deemed to be University,Odisha

3 Credit

Lecture Note 02

Chapter Contents

- **Object oriented programming concepts**
  - ✓ Classes
  - ✓ Objects
  - ✓ Encapsulation &Abstraction
  - ✓ Inheritance
  - ✓ Polymorphism
  - ✓ Dynamic binding
  - ✓ Message passing

Classes

✓ OOP, being specifically designed to solve real world problems, allows its users to create user defined data types in the form of classes.

✓ A class provides a template or a blueprint that describes the structure and behavior of a set of similar objects.

✓ Class does not take any space.

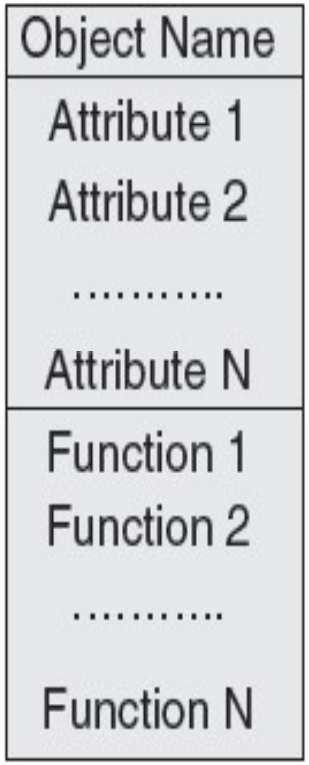
✓ Objects are nothing but variables of type class.

✓ Once a class has been defined , we can create any number of objects belonging to that class.

```
class student
{
      private:
         int roll_no;
         char name[20];
         float marks;
      public:
         get_details();
         show_details();
};
```

Objects

- ✓ Objects are basic run-time entities in an object-oriented system. Objects are instances of a class.
- ✓ Each object is associated with the data of type class with which they are created.
- ✓ A class is thus a collection of object of similar type. Example: mango, apple and orange are members of class fruit.
- ✓ Another example of student class and all students such as Aditya, Chaitanya, Deepti, and Esha are objects of the class.
- ✓ Every object contains some data and procedures. They are also called methods.
- ✓ When a program is executed the objects interact by sending messages to one another.
- ✓ Object take up space in memory and have an associated address like a record in pascal or structure or union in C.
- ✓ 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.



OOP Concepts

- Data Abstraction and Encapsulation
- Inheritance
- Polymorphism
- Dynamic Binding
- Message Passing

Data Abstraction and Encapsulation

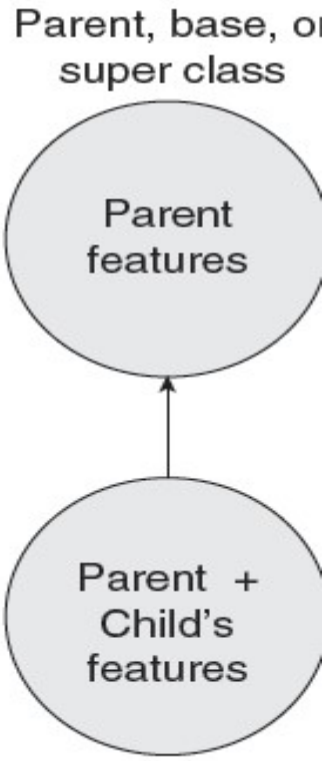
- ✓ The wrapping up of data and functions into a single unit (called class) is known as **encapsulation**.
- ✓ Data encapsulation is the most striking feature of a class.
- ✓ The data is not accessible to the outside world and only those functions which are wrapped in the class can access it.
- ✓ This 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 explanations.
- ✓ Data abstraction refers to the process by which data and functions are defined in such a way that only essential details are revealed and the implementation details are hidden.

Example of Encapsulation: Class (A class defines the structure of data and functions, which is common to all its objects.)

Classes use the concept of abstraction and are defined as a list of abstract attributes and functions. Thus, classes are known as **Abstract Data Types (ADT).**

Inheritance

- ✓ Inheritance is the process by which objects of one class acquire the properties of objects of another class.
- ✓ It supports the concept of hierarchical classification.
- ✓ Inheritance provides reusability. This means that we can add additional features to an existing class without modifying it.
- ✓ **Example:** Child and Parent



Polymorphism

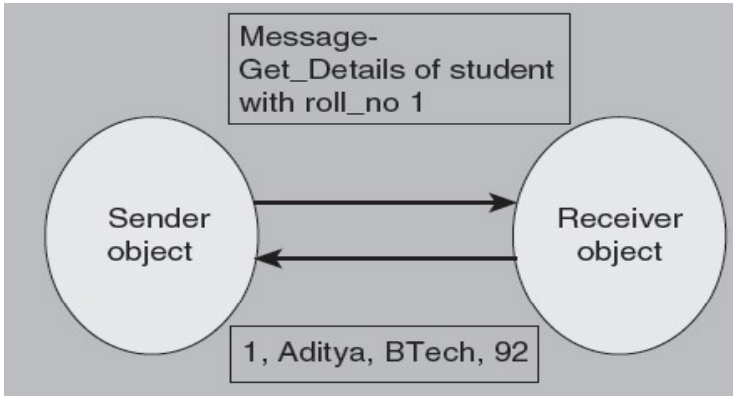
- ✓ Polymorphism means ability to take more than one form.
- ✓ An operation may exhibit different behaviours in different instances. The behavior depends upon the types of data used in the operation.
- ✓ Polymorphism is a Greek term, means the ability to take more than one form.
- ✓ Polymorphism= **poly (many) + morphism(forms)**
- ✓ C++ supports operator overloading and function overloading.
- ✓ The process of making an operator to exhibit different behaviors in different instances is known as **operator overloading**.
- ✓ **Function overloading** is using a single function name to perform different types of tasks.
- ✓ Example : Calculate Area

Dynamic Binding

- ✓ Binding refers to the linking of a procedure call to the code to be executed in response to the call.
- ✓ Dynamic binding (also known as late 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.
- ✓ In dynamic binding, the code to be executed in response to function call is decided at runtime. C++ has virtual functions to support this.
- ✓ For **example**, if we have a function print_result() in class student, then both the inherited classes, undergraduate and postgraduate students will also have the same function implemented in their respective classes. Now, when there is a call to print_result(), ptr_to_student-> print_result(); Then, the decision regarding which version to call—the one in undergraduate class or the one in postgraduate class—will be taken at the execution time.

Message Passing

- ✓ 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 required data elements.
- ✓ Two objects can communicate with each other through messages. An object asks another object to invoke one of its methods by sending it a message.



Benefits of OOP

Benefits of OOP

- Through inheritance, we can eliminate redundant code and extend the use of existing classes
- We can build programs from the standard modules that communicate with one another, rather than having to start writing the code from scratch
- Data hiding helps the programmer to develop secure programs that do not disturb code in other parts of the program
- Multiple instances of an object co-exist without any interference
- It is easy to partition the work in a project based on objects
- Object oriented systems can be easily upgraded from small to large systems
- Message passing technique for communication makes the interface descriptions with external systems much simpler
- Software complexity can be easily managed

Applications of OOP

- ✓ **Client-Server Systems**
- ✓ **Object-Oriented Databases**
- ✓ **Real-Time System Design**
- ✓ **Simulation And Modelling System**
- ✓ **Hypertext And Hypermedia**
- ✓ **Neural Networking And Parallel Programming**
- ✓ **Office Automation Systems**
- ✓ **CIM/CAD/CAM Systems**
- ✓ **AI Expert Systems**

More Details: https://www.quickstart.com/blog/10-applications-of-object-oriented-programming/

Object Oriented Languages

- ✓ The languages should support several of oop concepts to claim that they are Object-Oriented. Depending upon the features they support , they can be classified in to the folloowing two categories.
- ✓ **1. Object-based programming languages**

  It supports Data Encapsulation, Data hiding and access mechanism, Automatic initialization and clear-up of objects,operator overloading

  Ex: Ada

- ✓ **2. Object-Oriented Programming Languages.**

  It incorporates all of object-based programming features anlong with two additional features namely inheritance and dynamic binding.

  Object based Feature + Inheritance + Dynamic Binding

  Ex: C++, Simula, Smalltalk, Pascal, Java

THANK YOU!