Object Oriented Concept
Using C++

By: Mrs. Ashwini Pawar
Object oriented concepts

- Object oriented concepts
- Advantages
- Disadvantages
- Features
- Applications of OOPS
Object oriented methodology

• Object
  • Object are the basic run-time entities in an object oriented system.
  • Object is an Instance of a class.
  • Or we can say that object are state or behaviour.
  • For Eg. a female, a male, a place, a bank account, table.
  • They may also represent user defined data, such as vector, time and lists.
  • Program objects should be chosen such that they match closely with the real world objects.
  • Object take up space in memory and have an associated address like a structure in C
  • When a program is executed the object interacts by sending message to one another.
  • E.g: if customer and account are two objects in a program then the customer object may send a message to the account object requesting for the bank balance.
  • Each object contains data and code to manipulate the data.
  • Object can interact without having to know details of each others data or code.
Different system can have different object as discussed below:

1. Data structure like linked list, stacks, queue etc.
2. Employee payroll system
3. Item in inventory system
4. Customer in banking system
5. GUI elements like windows, menus, icons etc
6. Various elements in computer games like canons, guns, animals etc
7. Computer in a network model.
Diagram Of Object

Object

Attributes

Operation

Operation

Operation
Example

Example: StudentObject

- Enroll()
- Displayinfo()
- Performance()
- Result()
- st_name
- st_id
- branch
- semester
**Classes**

- Objects contain data and code to manipulate that data.
  - A class is a user-defined data type that consists of data and member function of an object.
  - Once the class has been defined, we can create any number of objects for that class.
  - Each class is associated with the data of type class with which they are created.
  - A class is thus a collection of objects of similar type.
  - E.g., Mango, Apple, Orange are members of the class fruit.
  - Classes are declared by using the keyword `class` followed by class name.
Diagram Of Class
Syntax and Example

**Syntax:** Class class_name
{
  Attributes;
  Operations;
};

**Example:** Class Student
{
  char st_name[10];
  int st_id;
  char branch[10];
  char semester[10];
  void Enroll();
  void Displayinfo();
  void Result();
  void Performance();
};
Data Encapsulation (also called as data hiding)

- Wrapping of data and function together is called as Encapsulation.
- Data is not accessible to outside world and only those functions which are wrapped in the class can access it.
- Data hiding is different from the security
- Security technique is used to protect computer database. To provide a security measure user have to provide or give a password before accessing the database so data is secured from unauthorized users.

**Example Of Encapsulation:**

- Our legs are binded to help us walk. and also hands are binded to hold different things.
- Another Example: In company there are multiple sections like Accounting, Finance, sales. Each Section handles work differently, means Accounting related with account section, Finance related with Finance and sales handles information about sales, so here we can say that different departments hide their information. Nobody allows to access directly data with each other, means data are hidden, here encapsulation is occur.
Diagram of Encapsulation

**Class:** student

**Attributes:** st_name, st_id, branch, semester

**Functions:** Enroll(), Displayinfo(), Result(), Performance()
Data Abstraction

- Abstraction refers to the providing important features and hiding their details.
- Classes use the concepts of abstraction and are defined as a list of abstract attributes such as size, weight and cost and functions to operate on these attributes.
- They encapsulate all the essential properties of the objects that are to be created.
- The attributes are sometimes called data members because they hold information.
- The functions that operate on these data are sometimes called member functions or methods.
- Since the classes use the concept of data abstraction they are known as abstract data type (ADT).
Example Of Abstraction

- We Know the car is Running but we don’t know how car is stop, we only know there is break but details are hidden.
- Another example is our body have a different parts but we can’t see the details.
Inheritance

- Inheritance is the ability to derive a new class from an existing one.
- It supports the concepts of hierarchical classification.
- It provides reusability and extendibility.
- **Example:** Parent Child Relationship.
- Child can acquire some properties of Parents, like talking, Smiling.

```
<table>
<thead>
<tr>
<th>Vehicle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Two Wheeler</td>
</tr>
<tr>
<td>Four Wheeler</td>
</tr>
<tr>
<td>Kinetic</td>
</tr>
<tr>
<td>Hero Honda</td>
</tr>
<tr>
<td>Santro</td>
</tr>
<tr>
<td>Indica</td>
</tr>
</tbody>
</table>
```
• This is possible by deriving a new class from the existing one.
• The new class will share the combined feature of both the classes i.e of base class and derived class.
• Note that each new sub-class will have its own properties or features in addition to the previous or base class features.
Polymorphism

• Poly –many  morphism  - form  
• It is important concept of OOP .
• Polymorphism is a Greek word which means the ability to take more than one form.
• E.g Consider the operations of addition for two numbers, the operation will generate a sum. If the operands were strings then the operation would produce a third string by concatenation.
• Another **Example** we can send normal text message and we can send video message also.
• The process of making an operator to exhibit different behaviours in different instances is known as operator overloading.
• Using single function name to perform different types of tasks is known as function overloading.
Types of Polymorphism

- Compile Time Polymorphism
  - Function overloading
  - Operator overloading

- Run Time Polymorphism
  - Virtual Function
Advantages Of C++

- **Portability:** Which allows the user to run the same program on different operating systems.
- **Multi-paradigm:** Paradigm means the style of programming, it includes the logic, structure, and procedure of the program.
- **Low-level Manipulation:** C++ language is similar to procedural and machine language like c, so it allows low-level manipulation of data at certain level.
- **Scalability:** Means ability of a program to scale. It means the program are running on a small scale and large scale of data.
- **Complexity:** Any time Software complexity can be managed.
- **Data Hiding:** Through the data hiding we can secure the programs.
- **Use of Inheritance:** We can remove the redundant code.
- **Message Passing:** Using the message passing we can communicate with the object.
Disadvantages Of C++

- **Not Secure:** C++ language is not secure because it has a pointer, Friend Function, Global variable.
- **Difficult to use in web application:** when we are trying to use C++ language in web application it is difficult to debug and complex.
- **Large Program Size:** The Programs developed by programmers is larger than the procedural language.
- **Slower Execution:** Since the Programs are larger in size, so that more instruction are to be executed, so that slower execution of programs.
- **Require Proper Planning:** Before designing the program the programmer should have a proper planning before designing a program.
Features Of C++

- **Simple:** This Language provides structured approach, set of Library functions and data types which is easier to use.
- **Rich Library:** It provides lot of inbuilt functions so that development of program is fast.
- **Memory Management:** It supports dynamic memory allocation. we can free the allocated memory by using free() function.
- **Extensible:** This language is extensible because we can easily adopt new features.
- **Object Oriented:** It is object Oriented language means development and maintenance of a program is easier than the procedural language.
- **Recursion:** Recursion Means we can call the function within the function. It provides the reusability for every function.
Applications Of CPP

- User interface design such as windows, menu.
- Real Time Systems
- Simulation and Modeling
- Object oriented databases
- AI and Expert System
- Neural Networks and parallel programming
- Decision support and office automation systems etc.
Thank You