Chapter 2
Object Oriented Programming

- Procedural Programming
- Object Based Programming
- Object Oriented Programming
- Basic Concept of OOP
- Implementing OOP Concept in C++
- Advantages and Disadvantages of OOP
Implementing OOP concept in C++

- OOP is an approach for writing software in which data and behaviour are packaged together as a Class whose instances are Object
Procedural Programming

- Procedural Programming is susceptible to design changes. In this type of languages whenever the definition of type changes the function referring to this type must also be changed.

- struct student
  {
  int rollno;
  char name[20];
  int clas
  };
Function for this structure

- void readstudent (student s1)

{  
  cout<< "Enter roll no; 
  cin>>rollno;
  cout<< "Enter Name 
  gets(name);
  cout<<"Enter Class; 
  cin>>clas;
  }

If the type change then function must also change

- struct student
  
  ```
  { 
    int rollno;
    char name[20];
    int clas
    float marks;
    char grade;
  };
  ```

- void readstudent (student s1)
  
  ```
  { 
    cout<< "Enter roll no:"
    cin>>rollno;
    cout<< "Enter Name:
    gets(name);
    cout<< "Enter Class:"
    cin>>clas;
    cout<< "Enter Marks:"
    cin>>marks;
  Calculate grade
  }
  ```
Drawback of Procedural Programming

- Procedural Programming leads to increased time and cost overheads during design changes. As design changes lead to many modification in the code, this lead to increased time and cost overhead at times.
Object Based Programming

- Object Based programming is a newer paradigm that implements some features of object-oriented programming, but not all. In Object based programming, data and its associated meaningful functions are enclosed in one single entity as CLASS. Classes enforce information hiding and abstraction.

- In other words, users of a class are allowed to access its interface, but they cannot access its implementation details.
Object based programming localizes the implementation details

- In object based programming whenever there is any changes in the definition of type, user interface remains unaffected generally
class student
{
    Int rollno;
    char name[25];
    Int clas;

    Public :
    void readstudent();
    void displaystudent();
}

This remain hidden
This makes user interface
If the changes occurred in design then the interface remain unchanged

- class student
  - 
  - 
  - Int rollno;
  - char name[25];
  - Int clas;
  - int marks;
  - float grade;

Public :

void readstudent();
void displaystudent();
Object based programming is subset of Object Oriented Programming

- The object based programming can be thought of as subset of object oriented programming as it also implement some of the features of OOP, such that Information Hiding, Abstraction, Classes, Function overloading etc. But it is not implementing the Inheritance and Polymorphism
Advantages and Limitation

- It overcome the shortcomings of Procedural Programming
- It localize changes and hides implementation details from user.
- It Support user defined data type
- This can not represent the real world problems.
- It does not hold the features of Inheritance- due to this reusability is not present.
- It does not hold the features of Polymorphism
Object Oriented Programming

- The Object Oriented programming paradigm is superset of object based programming. It is offer all the features of Object based programming and overcomes its limitation by implementing INHERITANCE. So that real world problems can be represented programmatically.
OOP Concept

- **OBJECT** --
  - Object is an identifiable entity with some characteristic and behaviour.

- **CLASS**
  - A Class is a group of objects that share common properties and relationship.
Example

- Cars have been identified as OBJECT. They have characteristics like steering wheel, seats, a motor, brakes etc. and their behaviour is their mobility. Car, however is not an object that belongs to the class \textit{CAR}. Car is a subclass of another class \textit{Automobiles} which again is a subclass of \textit{Vehicle}. Object is the instance of class. For example, Physically located Maruti Zen is an object of car.
Difference between Procedural and OOP approach

<table>
<thead>
<tr>
<th>Function 1</th>
<th>Function 2</th>
<th>Function 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>data 1</td>
<td>data 2</td>
<td></td>
</tr>
</tbody>
</table>

Properties
------------
Behaviour

Data and function is enclosed within object

All data is openly available for all function
About the OOP

- The object oriented programming has been developed with a view to overcome the drawback of conventional programming approaches. The OOP approaches is based on certain concept that help it attain its goal of overcoming the drawback or shortcoming of conventional programming approaches.
BASIC CONCEPT OF OOP

- The object oriented programming has been developed with a view to overcome the drawback of conventional programming approaches. The OOP approaches is based on certain concept that help it attain its goal of overcoming the drawback or shortcoming of conventional programming approaches.
Features of OOPs

- Data Abstraction
- Data Encapsulation
- modularity
- Inheritance
- Polymorphism
Data Abstraction

- Definition :-

- Abstraction is concept of simplifying a real world concept OR

- Abstraction refer to the act of representing essential features without including the background details or explanation.
Example

- You drive a car. You only know the essential features of driving a car. e.g. Gear handling, steering handling, use of clutch, accelerator, brakes etc.

- But when driving a car you don’t need to know the wiring, motor working etc.
Encaptulation

- Concept:
  Encaptulation is the most fundamental concept of OOP. It is the way of combining both data and the function that operate under a single unit.

- Definition:
  The wrapping up of data and operation / function into a single unit is known as Encaptulation.
The only way to access the data is provided by the functions. These functions are called "Member Function" in C++.

The data can not be access directly. If you want to read a data item in an object, you just call a member function in the object. It will read the item and return the value to you. You can’t access the data directly. The data is hidden. So it is safe from accidental moment.

Data and function are said to be encapsulated into a single entity.
Inheritance

- Definition :-
- Inheritance is the capability of one class of things to inherit capabilities or properties from another class.
- In other words, All the properties (Members) of base (Parent) class is present automatic in derive (Child) class.
For Examples

- Teacher and Student are the sub (Child) class of super (Parent) class Person. So all the properties of Person class is present in the Student and Teacher.
- Automobiles and Pulled Vehicles are sub class of Vehicles.
- Car and Bus are sub class of Automobiles.
- Rickshaw and Tonga are sub class of Pulled Vehicle.
There are several reasons why inheritance was introduced into OO language.

- Capability to express the inheritance relationship which makes it ensure the closeness with real world models.
- Idea of reusability
- Additions of Additional features to an existing class without modifying it.
Polymorphism

- Definition :-
- Polymorphism is the ability for a message or data to be processed in more than one form.
- Why Polymorphism :-
- Polymorphism is key to the power of object oriented programming. It is so important that languages that don't support this features can't advertise themselves as OO languages.
Polymorphism is a property by which the same message can be sent to objects of several different classes and each object can respond in a different way depending on its class.
Shape
area()

Circle
area(circle)

Triangle
area(triangle)

Rectangle
area(rectangle)
Implementing OOP concept in C++

- OOP is an approach for writing software in which data and behaviour are packaged together as Class whose instances are Object.
- A Class is a named software representation for an abstraction.
- An object is a distinct instance of a given class that is structurally identical to all other instances of that class.
- Software code in OOP is written to define classes, instantiate objects, and manipulate these objects.
Implementing Objects

- Our aim is to implement real-world objects in the form of software objects.
- Real-world objects have physical characteristics (state) and behavior, e.g., a motor car has:
  - Characteristics: Current Gear, Two wheel, Number of gears
  - Behaviors: Braking, Accelerating, Changing gears
- The software object also has state and behaviors:
  - These state is maintained through variables and data items.
  - The behaviors are implemented through functions.
Access Mechanism for Data

A Class group its members into three categories

- Private : This members remains hide from outside the world even in child classes.
- Protected : This members hide from outside the world but visible in child classes.
- Public : Access from any where in the program
Class Definition

- Class
- {
  - private : [optional]
  - Hidden data members and methods declaration
  - protected :
  - Unimportant implementation details
  - public :
  - Exposed important details
- };

Advantage of OOP

- The advantages offered by OOP are:
- Re-Use of Code: Linking of code to objects and explicit specification of relations between objects allows related objects to share code.
- Ease of Comprehension: OOP codes are more near to the real world models.
- Ease of Fabrication and Maintenance: Abstraction and Encapsulation allow for very clean design, when an object is going into disallowed state.
- Easy redesign and extension: We can easily enhanced the design without and extra Burdon.
Disadvantages of OOP

- There are many demerits here
- With OOP, Classes tend to overly generalized
- The relation among classes becomes artificial at times.
- The OOP program design is tricky
- Also need to proper planning and proper design for OOP programming.