Advantages of Object Oriented Programming :-

1. Exploits the expressive power of all object oriented programming languages.
2. Encourages the reuse of software components.
3. Leads to the systems that are more resilient to change.
4. Reduces the development risk.

Features of Object Oriented Programming :-

1. More focus on data.
2. Bottom-up approach.
3. Abstraction
4. Encapsulation
5. Inheritance
6. Polymorphism

Advantages Of Object Oriented Programming :

Advantages Of Object Oriented Programming

- OOP provides a clear modular structure for programs.
- It is good for defining abstract data types.
- Implementation details are hidden from other modules and other modules has a clearly defined interface.
- It is easy to maintain and modify existing code as new objects can be created with small differences to existing ones.
- Objects, methods, instance, message passing, inheritance are some important properties provided by these particular languages
- Encapsulation, polymorphism, abstraction are also counts in these fundamentals of programming language.
- It implements real life scenario.
- In OOP, programmer not only defines data types but also deals with operations applied for data structures.

Features Of Object Oriented Programming :

- More reliable software development is possible.
- Enhanced form of c programming language.
- The most important Feature is that it’s procedural and object oriented nature.
- Much suitable for large projects.
- Fairly efficient languages.
- It has the feature of memory management.
Structured programming only deals with procedures (functions) and variables. The basic idea behind the Object Oriented Programming is that it deals with the objects. In real life everything is an object and every object has properties and functionality.

As everything in the computer world is derived from the real life, like online books resemble with original hard bound books, the computer screen has more width than height as when we open a book its width is more than its height. So there was a need to make the computer programs more near to the reality. Object Oriented Programming is much nearer to the real world. In the reality everything is an object, it has some properties and do some functionality, inherits some functionality from others etc…

What is an Object?

Objects are the basic run-time entities in an Object Oriented System. They may represent a person, a place, a bank account, a table of data or any item that the program must handle. They may also represent user defined data such as vectors, time and lists.

Programming problem is analysed in terms of objects and the nature of communication between them. Program objects should be chosen such that they match closely with the real-world objects.

When a program is executed, the objects interact by sending message to one another. For example, 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 other’s data or code. It is sufficient to know the ” of massage accepted and the type of response returned by the objects.
Classes

If we say that a particular programming language is Object Oriented then it means, the programming language must have the facility to define classes in it. A class is an expanded concept of a data structure i.e instead of containing data only, it can contain both data and functions (operations on data).

What is a Class?

A class is an organisation of data and functions which operate on them. Data structures are called data members and the functions are called member functions. The combination of data members and member functions constitute a data object or simply an object.

In non-technical language, we can say that a class is a collection of similar object containing a set of shared characteristics. For example, mango, apple and orange are members of the class fruit. In a way, a class and its objects have the relationship of a data type and variables. A class is simply a template for holding objects. A class is abstract but objects are real. Simply by defining a class we don’t create an object just like the mere declaration of a data ” does not create variables. One or more classes grouped together constitute a program. Once a class has been defined, we can create any number of objects belonging to that class. For example, the syntax used to create an object is no different than the syntax used to create an integer object in C. If fruit has been defined as a class, then the statement fruit mango; will create an object mango belonging to the class fruit. A class is declared in the beginning of a program. A class can contain any number of data members and any number of member functions. A class can also have only data members.

How to declare a class in C++ Language

Here is an example of how to declare a class in C++ and create the objects of that class.

```cpp
class Line {
    public:
        Draw();
    private:
        int x1;
};
```
int x2;
int y1;
int y2;
}

10 Line line1,line2;

Both line1 and line2 are the objects of Line class.

Classes

A class is an organisation of data and functions which operate on them. Data structures are called data members and the functions are called member functions. The combination of data members and member functions constitute a data object or simply an object. In non-technical language, we can say that a class is a collection of similar object containing a set of shared characteristics. For example, mango, apple and orange are members of the class fruit. In a way, a class and its objects have the relationship of a data type and variables. A class is simply a template for holding objects. A class is abstract but objects are real. Simply by defining a class we don't create an object just like the mere declaration of a data ” does not create variables. One or more classes grouped together constitute a program. Once a class has been defined, we can create any number of objects belonging to that class. For example, the syntax used to create an object is no different than the syntax used to create an integer object in C. If fruit has been defined as a class, then the statement fruit mango; will create an object mango belonging to the class fruit. A class is declared in the beginning of a program. A class can contain any number of data members and any number of member functions. A class can also have only data members.

The first thing to do in object oriented programming is decide on a class hierarchy – that is, a hierarchy of kinds of objects, from very general ones to very specific kinds. In this simple example we can have a general class of ‘shapes’ and more specific classes for ‘circles’ and ‘rectangles’ (a realistic program would of course have many more such classes). A particular circle with specific size, colour etc will be an instance of the circle class. We can now decide on which operations and datafields can be defined for the most general class (shape). The following gives the type definition, and an example method (assuming that the type ColourType” is defined somewhere).

```cpp
class Shape {
public:
    Move(int x, int y);
    SetColour(ColourType c);
};
```
public:
    Move(int x, int y);
    SetColour(ColourType c);
private:
    int xpos, ypos;
    ColourType colour;
};

Shape::SetColour(ColourType c)
{
    colou