Note by: Roshan Bist
=Siddhanath Science Campus,
Mahendranagar

Subject: Object Oriented programming (C++)

Email: roshanbist900@gmail.com

Contact: 9806470952

Unit-1

Introduction to Object Oriented Programming

⊛ Procedure Oriented Programming (Structured Programming) →

It is a programming language that contain well-structured steps and procedures that uses different functions for different tasks in a program. In procedure oriented programming program it follows top to bottom approach. There is no access specifier in procedural programming. It follows systematic approach to solve the problem. For procedure oriented programming functions are more important than data in the program.

The advantage of using procedure-oriented programming language is it increases effectiveness and less time consumption in the program in any language. The drawback of this type of programming language is, in case we need to revise an external data structure, we need to revise all functions that access the data. Another serious drawback with the ~~prod~~ procedural approach is it does not model real world problems very well. This is because functions are action-oriented and do not really correspond to the elements of the problem.

(Not more imp)

Features/ Characteristics of procedure-oriented programming are:

i) Emphasis is on doing thing (algorithms).

ii) Large programs are divided into smaller programs known as functions.

iii) Most of the functions share global data.

iv) Employs top-down approach in program design.

v) Data move openly around the system from function to function.

vi) Functions transform data from one form to another.

Imp ⊛ **Object-Oriented Programming →**

It is a programming language based on the concept of objects and classes ~~in the form~~ which contain data in the form of fields often known as attributes and code in the form of procedures often known as methods. It is associated with the concept of class and objects and various other concepts revolving around these two like Inheritance, Polymorphism, Abstraction, Encapsulation etc.

The major factor in the invention of object-oriented approach is to remove some of the flaws encounted in procedural approach. It treats data as a critical element in the program development and does not allow it to flow freely around the system. The data of an object can be accessed only by the functions associated with that object. However, functions of one object can access the functions of other objects. The disadvantage of this is difficult to understand for beginners.

features of OOP are:- (Not more imp)

i) Emphasis is on data rather than procedure.
ii) Programs are divided into parts known as objects.
iii) Access specifiers are used.
iv) Data is hidden and cannot be acessed by external functions.
v) Objects may communicate with eachother through functions.
vi) New data and functions can be easily added whenever necessary.
vii) Follows bottom-up approach in program design.

⊛ Defferences between Procedure-Oriented and Object-Oriented Programming.

Procedure-Oriented Programming	Object-Oriented Programming
i> In procedure oriented programming, program is divided into small parts called functions.	i> In object-oriented programming, program is divided into small parts called objects.
ii> It follows top-to down approach.	ii> It follows bottom to up approach.
iii> There is no access specifier in procedural programming.	iii> Programming have access specifiers like private, public and protected.
iv> Adding new data and function is not easy.	iv> Adding new data and function is easy.
v> It does not have any proper way for hiding data so it is less secure.	v> It provedes access specifier private for hiding data so it is more secure.
vi> Overloading is not possible in procedure-oriented.	vi> Overloading is possible in object-oriented.
vii> It is based on unreal world	vii> It is based on real world.
viii> Examples - C, FORTON, Pascal etc.	viii> C++, Java, Python, C# etc are its examples.

Imp ⊛ Characteristics/ Terms of Object-Oriented Languages:-

It is necessary to understand some of the concepts/terms used extensively in object-oriented programming which are as follows:.

1) **Objects** → Objects are the basic run-time entities in an object-oriented system. Objects are member variable (variable of member function) of a class which are user-defined data types.

Example:-

```
Class person
        {
            Char name [20];   } data members
            int id;
            public:
            void getdetails(){}  // member function
         };
    int main()
        {
          person p1; //p1 is object
        }
```

Programming problem is analysed in-terms of objects and the nature of communication between them. Program objects should be choosen such that they match closely with the real-world objects. Object take up space in memory and have an associated address like structure or union in C. When a program is executed the objects interacts by sending message to one another with class associated with it. In fact, objects are variables of the type class.

Class does not take memory space.

2) Class → Objects contain data and code to manipulate data. The entire set of data and code of an object can be made a user-defined data type with the help of class. In fact, objects are variables of type class so, once a class has been defined, we can create any number of objects belonging to that class. A class is thus a collection of objects of similar type. Classes are also user-defined data types and behaves like a built-in types of a programming language.

The syntax to create an object is ~~simal~~ similar to the syntax used to create an object (variable) 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.

& The syntax for class is as follows:

```
class class_name
            {
               private:
             // data members & member functions
               public:
             //data members & member functions
               protected:
             // data members & member functions
            };
```

We will use private, public and protected access specifiers according to our need in program. Class is a user-defined data type like structures and unions in C. If we do not provide these access specifier to data members then, by default these data members will be private.

Note:- In theoretical understanding object comes first then class comes but, while writing program class comes first then the object comes.

OR hiding

3) Data abstraction and Encapsulation →

The wrapping up (or combining) of data and functions into a single unit is known as encapsulation. The data is not accessible to the outside world and only those functions which are wrapping in the class can access it. The insulation of the data from direct access by the program is called data hiding or information hiding.

Data abstraction refers to, providing only needed information to the outside world and hiding implementation details. Classes use the concept of abstraction and are defined as a list of abstract attributes such as size, weight, cost etc. and functions to operate on these attbibutes called member functions. The attributes are sometimes called data members because they hold information. The advantage of abstraction is we can change implementation at any point without affecting users of complex class.

4) Inheritance → Inheritance is the process by which the objects of one class acquire the properties of objects of another class. It helps to share common characteristics with the class from which it is derived. For example: the bird 'robin' is a part of class 'flying bird' which is again a part of class 'bird'.

The concept of inheritance provides the idea of reusability. This means that we can add additional features to an existing class without modifying it.

OR
Overloading

5) Polymorphism → Polymorphism is the ability to make more than one form. Any operation may show (exhibit) different behaviours in different instances (needs). For example consider the operation of addition. For two numbers, the operation will generate sum but if the operands are strings, then the operation will produce third string by concatenation. The process of making an operator to show different behaviours in different instances (conditions/urgency/need) is known as operator overloading.

6) 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 called 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.

7) Message Passing → Message passing is the communicating objects with one another by sending and receiving information. A message for an object is a request for execution of a procedure, and therefore will call a function in the receiving object that generates the desired result. It involves the name of object, name of function (message) and the information to be sent.

invaded → moved into

Date.

Page No.

(less imp)

* Benefits/Advantages of OOP:

i) Through inheritance, we can eliminate redundant code and extend the use of existing classes.

ii) The principle of data hiding helps the programmer to build secure programs that can not be invaded by code in other parts of the program.

iii) It is easy to partition the work in a project based on objects.

iv) The data-centered design approach enables us to capture more details of a model in implementable form.

v) Software complexity can be eaisly managed.

vi) It is possible to map objects in the problem domain to those in the program.

vii) Object-Oriented systems can be eaisly upgraded from small to large systems.

viii) It saves program development time with higher productivity.

(less imp)

* Application areas of OOP:

i) Real-time systems

ii) Simulation and Modeling.

iii) Object-oriented databases.

iv) Hypertext, hypermedia and expertext.

v) AI and expert systems.

vi) Neural networks and parallel programming.

vii) Decision support and office automation systems.

viii) CAD systems.