⑨ Object-Oriented Programming:—

C++ language supports object-oriented programming, including the four pillars of object-oriented development:—

→ Encapsulation
→ Data Hiding
→ Inheritance
→ Polymorphism.

One-by-one we will give explanation with examples.

Encapsulation → It is one of the key features of Object-Oriented programming. This feature allows programmers to bundle all data members and functions that are similar type and behaviour in a single bundle named class.

In other words, encapsulation is a process of wrapping similar codes (data members and functions working on them) inside a single class.

For example:-

```
class Rectangle {
public:
    int length;
    int breadth;
int getArea() {
    return length * breadth;
        }
    };
```

Data Hiding → In C++ we can make data hiding by specifying access specifiers — private and protected. In this way, public specifier can be used for the functions that access these private and protected datas.

Inheritance → This is the capability of a class to derive properties and characteristics from another class. It is also very important feature of Object Oriented Programming. For example:- We have a class related to admission of students. From there we can now extend another class for examination form fill-up, so that all necessary basic data of students are available there. Hence this is the feature where we can use pre-defined codes that is re-usability of programs. It is also very important feature of Object Oriented Programming.

Syntax of Inheritance is :-

```
class Students {
        private:  int Roll;
                  char name[30];
                  char course[10];
        protected: - - - -
        public : - - - -
          };

class ExamForm : public Students {
        private: ___
        protected: ___
        public: - -
          };
```

In above class - Students, all necessary data filled while taking admission are re-usable through class ExamForm and are kept in protected: type. The data that are no needed for extended class are kept in private: section.
The functions are usable in class itself as well as in derived class are kept in public: section.

Polymorphism → The meaning of Polymorphism is - the ability to take many forms. Here, we can display a message in many forms. A real life example of polymorphism is - a person who at the same time can have different characteristics. A man at the same time is a father, a husband, and an employee. So, the same person exhibits different behavior in different situations. This feature is called Polymorphism.

In C++, There are two types of polymorphism —

- Compile-time Polymorphism
- Runtime Polymorphism.

Compile-time Polymorphism can be applied by function overloading or operator overloading.
Run-time Polymorphism can be achieved by Function Overriding. It is also called late binding and dynamic polymorphism.

We will discuss both later with examples.