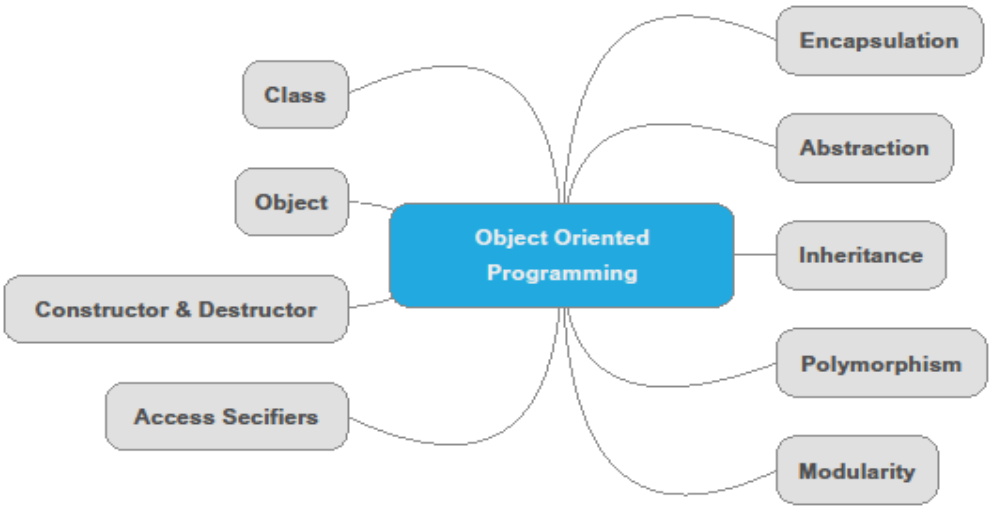
Unit-I
Object Oriented Programming in C++

OOP, Classes And Objects — Chapter: 02



Class :- A class is collection of data (data member) and functions (member functions or methods) working on the data. It can be seen as a blue print for the object. No memory is allocated when a class is created. Memory is allocated only when an object is created.
Object :- An Object is an instance of the class means have all the properties defined in the class.
Data member:- The data variables declared within the class.
Member functions :- Member functions are the methods which are declared/defined inside the class and operate upon the data member.
Data Abstraction: - Data abstraction represents essential features without including background details.
Data Encapsulation:- Binds the data and its functions into a single unit called class.
Data hiding:-Hides internal object details (data members). Data hiding ensures exclusive data access to class members and protects object integrity by preventing accidental or intended changes.
Inheritance: Inheritance is the process of forming a new class from an existing class or base class.
Base Class :- The class from which methods and data members are derived to new class is knows as base class. The base class is also known as parent class or super class.
Derived Class:- The class that is deriving data and methods from base class is called derive class. Derived class is also known as a child class or sub class.
Polymorphism:-Poly means many and **morphs** mean form (Multiple Forms). Refers to the ability of processing of data in more than one form.
Access specifier :-private, protected, public (default access specifier is private)
Accessibility of private, protected and public members

Accessibility	**Private**	**Protected**	**Public**
Through member functions	Yes	Yes	Yes
Through object of the class	No	No	Yes
Through derived class	No	Yes	Yes

Syntax of a class	Example

class <class_name>
{
private:
declaration of data member;
declaration/definition member function;
protected:
declaration of data member;
declaration/definition member function
public:
declaration of data member;
declaration/definition member function
};
	class student
	{
	private:
	char name[30];
	int age;
	int marks;
	protected:
	char grade;
	public:
	void getdata();
	void showdata();
	};

Referencing class members:- All the data members of the class are directly accessible to the member function of that class. They don't need any object name to be prefixed before it but from outside the class any reference to the data member is done with the dot (.) operator.

syntax for creating an object:

<class_name><Object_name>;

Example:

student s1;

Accessing members from object of the class:-A data member and member function declared under public access specifier can be accessed by the objects directly.

objectname.member;

e.g.

s1.getdata();

s1.showdata();

Defining class methods/Member functions. Member functions of the class can be defined in the following two ways

(a) Inside the class definition (inline function)

In this method, the function is defined within the class body and are treated as inline by default.

(b) Outside the class definition.

In this way function prototype is declared within class body and function is defined outside the class with the help of Scope Resolution operator (::).

Syntax for defining a member function outside the class definition.	Example for defining a member function outside the class definition.

<return type><class name> :: <function name>(parameter list)
{
body of the function
}
	void student::showdata()
	{
	cout<<"\n Name "<<name;
	cout<<"\n Age "<age;
	cout<,"\n Marks"<marks;
	}

Example: Defining member function within class body.	**Example: Defining member function outside class body.**

class STUDENT
{
int rollno;
char Name[20] ;
char Address[30] ;
char PhoneNo[15] ;
public :
void enter( )
{
cout<<"\n Enter Rollno name address and phone no. ";
cin>>rollno;
cin.getline(Name,20);
cin.getline(Address,30);
cin.getline(PhoneNo,15);
}
void display( )
{
cout<<"information of student is";
cout<<rollno<<Name<<Address<<PhoneNo;
}
} ;
	class STUDENT
	{
	int rollno;
	char Name[20] ;
	char Address[30] ;
	char PhoneNo[15] ;
	public :
	void enter();
	void display();
	}
	} ;
	void STUDENT :: enter( )
	{
	cin>>rollno;
	cin.getline(Name,20);
	cin.getline(Address,30);
	cin.getline(Phoneno,15);
	}
	void STUDENT :: display( )
	{
	cout<<"information of student is";
	cout<<rollno<<Name<<Address<<Phoneno; }

Short Answer Type Questions (2 Marks)

Q1.What do you understand by class and object in C++?
Q2.What all members of a class are accessible to outside the class body?
Q3.What do you understand by data abstraction?
Q4. What do you understand by data hiding?
Q5 What do you understand by Encapsulation?
Q6. What is polymorphism?
Q7. What do you understand by member function? How does a member function differ from an ordinary function?
Q8.How does a class implements Data Abstraction and Encapsulation?

Long Answer Type Questions (4 Marks)

Q1. Define a class train with following members.
Private members
Trainno - type int
Destination -type String
Distance - type float
Fuel - type float
A member function calfuel() to calculate and assign value of fuel as per the following criteria:

Distance	**Fuel**
<=1500	250
>1500 and <=3000	1000
>3000	2500

Public member:

1. feedinfo(): to input train no, destination, distance and invoke calfuel() function to assign value of fuel.
2. showinfo(): to display all the details for a train.

Q2. Define a class employee with following specifications:
Private members:
empno integer type.
ename 20 characters long String.
basic,hra,da float type.
netpay float type.
ctotal() A function to calculate the total basic.
Public member:
read_data() A function to read empno, ename, basic, hra, da and call ctotal ()to calculate total.
display_data() A function to display all the data members on the screen.