

OOP -

Encapsulation - wrapping of data Ex Class

Abstraction - representing essential things and hiding complexity object and access

Inheritance - we do reusability and extensibility by using predefined components (classes, interfaces) which are tested. Which will save efforts, time etc.

Data binding(polymorphism) - ability of behaving differently in diff. Situation.

Compile time - Fun. overloading and opr. overloading

Run Time - virtual function (with pointer of base we access function of child by assigning object of child to pointer of base)

Class - user defined DT / blue print for creating object / structure / template of DT

Object - real world entity - things which exist in reality - tangible / intangible things

Will have properties and behaviors and those properties will be having states - values

When we will create any class / declare it is just structure no memory gets assigned to any var. Or function until and unless object gets created.

While creation of object constructor functions get called

C++ provides default, copy constructor by default if not written any constructor

But if we have given definition to any constructor manually then for other constructor we need to give definition.

Classname identifier ;

Memory gets allocated to object depends on number of data members in class

Member function occupies common memory

Defining functions in C++-

1. In class
2. Outside class with class name and :: operator.

Functions

Call by val.

Call by ref.

Call by add.

Inline function -

In class when we define functions inside class all functions treated inline by default. But if it is big function (more lines of code ) will not treat as inline

Friend fun

When any out sider function wants to access private/ protected data member of class we need to declare that fun as friend inside class in any scope.

1. Out sider fun. Is friend
2. Member fun of 1 class friend of other class // need forward declaration
3. Whole class as friend of other class - all members of that friend declared class will be friend to other class // need forward declaration

Constructor - special member fun - name is same as class name and no return type and body  
Used to construct object - initialize data members for that object.

Types of cons.

Called automatically

Destructor - to destroy object get called automatically when obj becomes no longer in use./ will go out of scope

Get called in reverse order in constructor get called

DMA in c++-

New - to allocate memory - ptr=new DT( val) - for initialization/[ size] - for array

Delete - to free memory delete p;

Static var & funt-

Class level funt. Or var never come along with object it's kind of global var.

No need of obj. To call or access these mem.

called/ accessed with class name :: opr.

Const - value cannt be changed / fun cons will not be modifiable

This - pointer in c++

Used to access current object in it's called function ( normal or constructor)

Inheritance -

There will be multiple classes used when one class acquires properties of other class.

Class acquire properties - derived / child / sub class

Class whose properties get inherited in other class- parent/super/Base class

Type-

Single - one base one child class

Multiple - many parent class of a child

Multy level - level get created one class derived from other derived class

Hybrid - combination of above types

Hierarchical - one parent can have many child

Hybrid - virtual inheritance - diamond problem - multipath inheritance - problem of duplicate copies of super class by multiple path comes down in child class ==> ambiguity error

To resolve - while inheriting super parent class we declare it virtual.

Polymorphism

Compile time -

Fun. overloading - multiple functions with diff signature ( no of para./ seq. Of arg. / type of arg)

Operator overloading - giving some special meaning to operator by defining function.

It get called when we use this operator with user defined var. On in combination

$C1 + c2$   $c1 + 4$ ;  $4 + c2$

OO is done - 1. By member fun. - RHS operand passed and LHS operand used to call function.

2. By friend function - all operands will be passed as an arg.

Runtime poly-

Virtual fun. - virtual function (with pointer of base we access function of child by assigning object of child to pointer of base)

It will be done in inheritance where fun. Overriding will be done first.

Fun overriding - fame function defined in parent & child class ( snake name same parameter same RT) only body will be diff.

Child func. Copy will get override on parents copy in child class after inheritance.

So when we achieve RT polymorphism with the help of ptr to parent assigns to child object without virtual function not possible

Pure virtual fun - when there is no specific implementation of parents virtual function or we don't want to write definition for parents virtual function so we assigns = 0 to declaration of virtual function. So it becomes pure VF.

Virtual fun(){} - virtual func.

Virtual fun()=0; - pure virtual func.

We declare any single pure function inside class - > Abstract class

Abstract class -

The class whose object will not be created or not allowed to create object.

Template - concept is used for generic programming - blue print or structure for making / creating \ different functions or different classed according to different data types

1. Function template - is nothing but function structure get execute for different types of values

No need of creating different func. Definition

add(2,3) , add(3.4 ,4.5) add('A', 'C') ....

2. Class template - to create class structure which will be used with different data type as it is class with 2 int will work same as it is for class with 2 float / 2 double ....

Syntax -

template<class T > followed by function definition / class declaration

T - generic type which will be used for any type value is passed.



