Manish Sultanpure ***C++ Language***

* **First pure object oriented programing language is Smalltalk.**
* **Class concept was come from SIMULA 67 language.**

**C++ language was developed by bjarne stroustrup in 1978 in AT & T Bell lab.**

**bjarne stroustrup AT & T bell lab me kam karte the or vo C and simula67 language pr kam kerte the.** **unko Simula67**  **language ke concept acche lagte the to unhone socha kyo na mai C programming me oops ka use.**

**karu to unhone 1980 me class and object ko C programing me add kiya and 1980 me uska name rakha C with class .**

**1983 me jab C language me Simula67 ke sare concept add ker diye to uska name C++ rakha gaya.**

**Q what is OOPs?**

**Ans:-OOPs is a programing paradigm which is based on object.**

**Top OOPs languages:-**

* **C++**
* **Java**
* **Php**
* **Python**
* **Smalltalk**

**C++ Syllabus :-**

* **Class**
* **Object**
* **Data abstraction**
* **Data encapsulation**
* **Friend function**
* **Friend class**
* **Constructor (default constructor, parameter cons., copy cons.)**
* **Destructor**
* **Constructor overloading**
* **Inheritance**
* **Type of inheritance**
* **Inheritance with constructor**
* **Static keyword**
* **Static method**
* **Polymorphism**
* **Function overloading**
* **Method overloading**
* **Method overriding**
* **Operator overloading**
* **Virtual function**
* **Pure virtual function**
* **Diamond problem**
* **Break keyword**
* **Continue keyword**
* **Constant keyword**
* **Type def keyword**
* **File handling(I/O)**
* **Dynamic memory allocation**
* **Recursion**
* **Bubble short**
* **Template class**
* **Inline function**
* **Internal function**
* **External function**
* **Public**
* **Protected**
* **Private**

**insertion**( << ) **operator**:- **it is use with cout .**

**extraction**( >> ) **operator**:-**it is use with cin.**

**cout:- it is use for print content on console .It comes from iostream.h heddar file.**

**cin:- it is use for take input from user .It comes from iostream.h heddar file.**

**Advanced version of structure is class.**

**Structure ke ander ka sara data by default public hota hai. But class by default private hoti hai. java me default hoti hai.**

**Private ka mtlba vo data kabhi bhi class ke bahara use nhi ho sakta hai.Hum class ke data ko manually public keyword ke help se public bana sakte hai.**

**class ke ander and method ke bahar jo variable declare hota hai usse instance variable kahte hai.**

**Class ka use karne pr bina object ke class ka data access nhi ho sakte hai.**

**Jo class ke bahar declare hota hai usse function kahte hai and jo class ke ander declare hota hai use method kahte hai.**

**Class ke ander ka data object se access hota hai esliye en method ko instance method kahte hai.**

**Class:-class is not real time antity. It has no of similar instructions and data are called class. Ex:-Fruit, Animal.**

**Object:-object is real time antity. It has particular type of data of class.**

**Ex:- apple , dog ,cat.**

**NOTE:-In C language \n is use for line change and in C++ endl is use for line change it is manipulator. But \n takes 2 bytes endl does’nt take any byte.**

***Class*:-class is collection of data member and member functions OR class is blue print for object.**

**EX:-car is one class and lots of different – different brands car is available in market but every car has some properties are same like 4 wheels , speed limit etc..**

* **A class is user define data type. Which has data member and member functions.**
* **Data member are the data variable and member function are the functions used to manipulate variables together. These data members and member functions are define the properties and behavior of object.**

***Object:-*An object is an instance of class. When object is create memory is allocated by compiler. Ex:-apple, dog ,cat.**

***Data abstraction:-* abstraction means display only essential information and hiding the implementation details. Data abstraction refers to providing only essential Information about the data and hiding the background details or implementation to the outside world.**

**Ex-: ATM machine everyone knows how to withdrawal money but we don’t know how it internally works means we don’t know implementation details.**

***Data encapsulation:-* wrapping up the data member and member function in single unit is called class.**

**For example**: **capsule, it is wrapped with different medicines.**

**Programing ex:- there is cube function whose predefine functionality like**

**class A{**

**public:**

**int cube(int x){**

**return x\*x\*x;**

**}**

**} ;**

**User can use directly like**

**A a;**

**Cout<<a.cube(10);**

**Advantage of OOPs over Procedure-oriented programming language**

1. **OOPs makes development and maintenance easier where as in Procedure-oriented programming language it is not easy to manage if code grows as project size grows.**
2. **OOPs provide data hiding whereas in Procedure-oriented programming language a global data can be accessed from anywhere.**
3. **OOPs provide ability to simulate real-world event much more effectively. We can provide the solution of real word problem if we are using the Object-Oriented Programming language.**

* **C++ instance variable ko direct initialize nhi ker sakte hai .Java me ker sakte hai.**

**Like int x=20;//instance variable**

* **Har ek instance ke corresponding instance variable ki shaprate memory create hoti hai.**

**Sabse pahle mithai ko room means {method} ke ander dhundega ager nhi milege to hall means {class} me dhundega ager hall me bhi nhi mile to gussa aa jayega means error de dega.**

* **Yadi class private hai to Function ke ander se method ko call nhi ker sakte hai.**

**this pointer**

**this pointer hamesa instance variable ko hi point karta hai.**

**this pointer current object ke refrence ko hold karta hai.**

**Syntax:- this ->x=x;**

**friend function**

**public class ke n number of function ko access ker sakte hai.but private class ke koi bhi function ko access nhi ker sakte hai.**

**But ham 1000 funtion me se 700 ko hi access karna chate hai 300 ko nhi to hame es case me friend fuction bnayege.**

* **Yadi ham private class ke function ko class ke bahar access karna cahate hai to friend function ka use karege. Jis public function se access karna chate hai na use class ke ander friend keyword se declear karege. Like friend void show(); .**
* **We can make friend of main method also.**
* **We can make friend class also for access all the data of private class.**

**Ex:-friend class A;**

**Syntext of friend function:-**

**class class\_name**

**{**

**friend data\_type function\_name(arguments);//  friend function.**

**};**

**Characteristics of a Friend function:**

* **The function is not in the scope of the class to which it has been declared as a friend.**
* **It cannot be called using the object as it is not in the scope of that class.**
* **It can be invoked like a normal function without using the object.**
* **It cannot access the member names directly and has to use an object name and don’t membership operator with the member name.**
* **It can be declared either in the private or the public part.**

**Constructor**

**Q what is constructor?**

**Ans:- A constructor is special member function of class . which initialize the object of class.**

**Constructor is different from normal function:-**

* **Constructor must has same name as the class name.**
* **Constructor don’t have return type.**
* **It is automatically call when object is created .**
* **If we don’t specify a constructor compiler add by default default constructor.**
* **Constructor should be declared in public section.**

**Constructor are mainly three types:-**

* **Default Constructor.**
* **parameterized Constructor.**
* **Copy Constructor.**

**Q what is default constructor?**

**Ans:- without parameter constructor called default constructor. It is add by compiler by default.**

**Q what is parameterized constructor?**

**Ans:-A constructor with parameter is called parameterized constructor.**

# **Q what is copy Copy Constructor?**

**Ans:- jab hum ek constructor ke data ko dusre constructer me use kerna chate hai to to hum ek constructor ke object ko dusre constructor ke object me initialize ker ke kar sakte hai.yaha karna hi copy constructor kahlata hai.**

## Copy Constructor is of two types:

* **Default Copy constructor:** **The compiler defines the default copy constructor. If the user defines no copy constructor, compiler supplies its constructor.**
* **User Defined constructor:** **The programmer defines the user-defined constructor.**

**Syntax Of User-defined Copy Constructor:**

1. Class\_name(**const** class\_name &old\_object);

**When Copy Constructor is called**

**Copy Constructor is called in the following scenarios:**

* **When we initialize the object with another existing object of the same class type. For example, Student s1 = s2, where Student is the class.**
* **When the object of the same class type is passed by value as an argument.**
* **When the function returns the object of the same class type by value.**

**Constructor overloading**

* **Ek hi class**
* **Same name**
* **Different parameter**

**Ek hi class ke ander same name ke different parameter ka constructor crate karna hi constructor overloading kahlata hai.**

**Difference between constructor and method**

**Constructor method**

* **Constructor or class ka name same method or class ka name different ho**

**Hota hai. Sakta hai.**

* **Constructor ka koi return type method ka return type hota hai.**

**Nhi Hota.**

* **Constructor automatic call hota hai method ko implicitly call karna padta**

**when object is created. hai.**

* **Ek objet ke corresponding constructor method ko 1 object se n number of**

**Ek hi bar call hota hai. time call ker sakte hai.**

* **Constructor ko hamesa public area method ko kahi bhi create ker sakte**

**me create karte hai. hai.**

**Note:- C++ me constructor ke agge return type likhte hai to error ayege. But java me constructor ke agge return type likhne se vo apne ap ko method maan leta hai.**

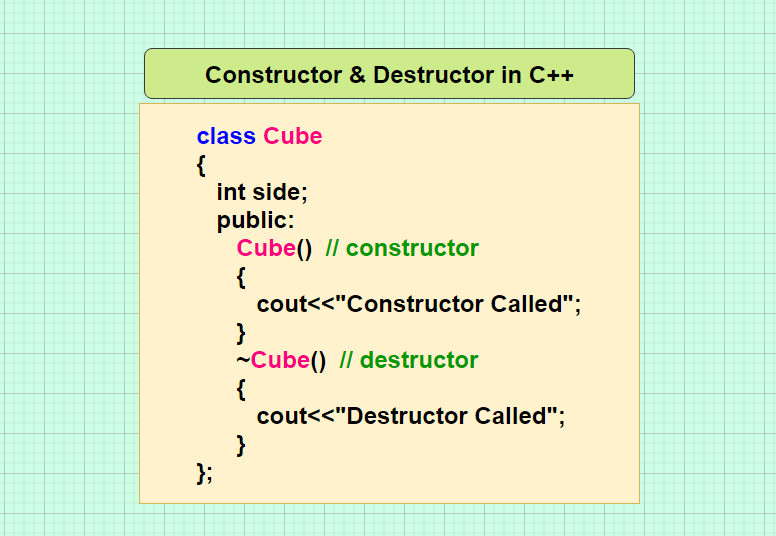
# **Destructor**

**Q what is** **destructor?**

**Ans:-A destructor works opposite to constructor. It destructs the objects of classes. It can be defined only once in a class. Like constructors, it is invoked automatically.**

**A destructor is defined like constructor. It must have same name as class. But it is prefixed with a tilde sign (~).**

#### **Note: C++ destructor cannot have parameters. Moreover, modifiers can't be applied on destructors**



***Inheritance***

# **Q what is inheritance**

**Ans:-In C++, inheritance is a process in which child class acquires all the properties and behaviors of its parent class automatically. In such way, you can reuse, extend or modify the attributes and behaviors which are defined in other class.**

**In C++, the class which inherits the members of another class is called derived class and the class whose members are inherited is called base class. The derived class is the specialized class for the base class.**

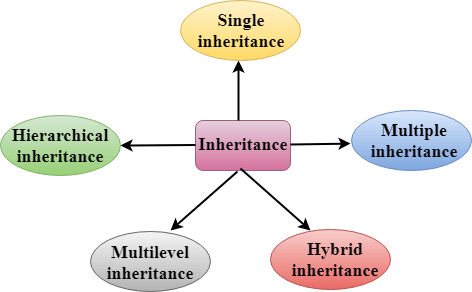
## Advantage of C++ Inheritance

**Code reusability:** Now you can reuse the members of your parent class. So, there is no need to define the member again. So less code is required in the class.

## Types of Inheritance

**C++ supports five types of inheritance:**

* Single inheritance
* Multiple inheritance
* Hierarchical inheritance
* Multilevel inheritance
* Hybrid inheritance



* **Hamesa super class ka constructor pahle call hota hai inheritance ke case me.**

## Single Inheritance

**Single inheritance** **is defined as the inheritance in which a derived class is inherited from the only one base class.**

C++ Inheritance

**Where 'A' is the base class, and 'B' is the derived class.**

## Multilevel Inheritance

**Multilevel inheritance** **is a process of deriving a class from another derived class**.

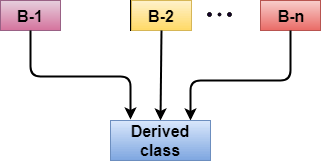
C++ Inheritance

**When one class inherits another class which is further inherited by another class, it is known as multilevel inheritance in C++. Inheritance is transitive so the last derived class acquires all the members of all its base classes.**

**Multiple Inheritance**

**Multiple inheritance** **is the process of**

**deriving a new class that inherits the attributes from two or more classes.**

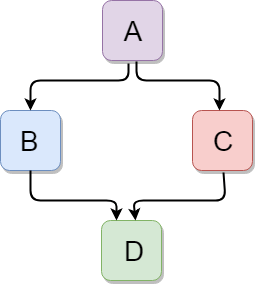


**Syntax of the Derived class:**

1. **class** D : visibility B-1, visibility B-2, ?
2. {
3. // Body of the class;
4. }

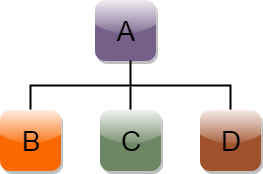
## Hybrid Inheritance

**Hybrid inheritance is a combination of more than one type of inheritance.**



**Hierarchical Inheritance**

**Hierarchical inheritance is defined as the process of deriving more than one class from a base class.**



**Syntax of Hierarchical inheritance:**

1. **class** A
2. {
3. // body of the class A.
4. }
5. **class** B : **public** A
6. {
7. // body of class B.
8. }
9. **class** C : **public** A
10. {
11. // body of class C.
12. }
13. **class** D : **public** A
14. {
15. // body of class D.
16. }

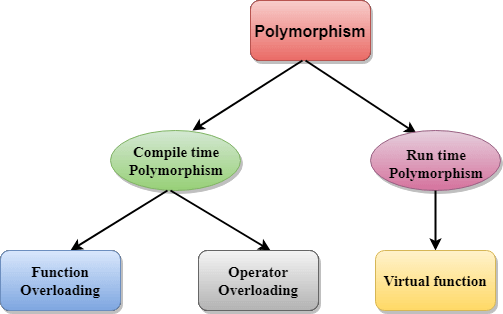
**Polymorphism**

**Q what is polymorphism?**

**And:-** The **word polymorphism** comes from the Greek **words** for "many shapes" **. Typically, polymorphism occurs when there is a hierarchy of class and they are related by inheritance.**

**In C++ polymorphism means that a call to a member function will cause a different function to be executed depending on the type of object that invokes the function.**

**Type of polymorphism**



* **Polymorphism means more than function with same name ,with different working.**
* **Polymorphism can be static or dynamic.**
* **In static polymorphism memory will be allocated at compile time.**
* **In dynamic polymorphism memory will be allocated at run time.**

**Ex:-A person at a same time can have different characteristics like a man, at the same time is a father, a husband, an employee, so the same person posses different behaviors in different situation . This is called polymorphism.**

**Function overloading :- \* same name function.**

* **Different parameters.**

**There is no class in function case.**

**Method overloading:- \* same function name.**

* **Same class.**
* **Different parameter.**

**Method overriding:- \* same function name.**

* **Different class and relation (inheritance).**
* **Same parameter list.**

**Operator overloading :-Variable ke jese object ko bhi ek object se durse object me assign ker sakte hai. ese hi operator overloading kahte hai.**

* **2 object ko add ker ke third me put karna hi operator overloading hota hai.**
* **Operator ki definition ko change karna object overloading kahlata hai.**

**Static binding:-**

**Q what is compile time binding?**

**Ans:- Static binding** happens when all information needed to call a function is available at the compile-time.

* **Compiler sabse pahle cheak karta hai ki kis class ka object hai.**
* **Compiler cheak karega ki class A me show name ki method hai ya nhi.**
* **Jab compiler ko show name ki method mil jayege .**
* **Compiler ab object ko class A ki show name ki method kai sath compile time pr hi bind ker dega.**
* **Yadi compile time binding ho jate hai to fer kabhi bhi run time pr binding nhi hogi.**
* **But yadi compile time pr binding nhi hoti hai to fer defiantly run time pr binding hogi.**
* **Run time pr binding hamesa address ke corresponding hoti hai.**

**Dynamic binding:-**

**Q what is dynamic binding?**

**Ans:-** Dynamic **binding** happens when the compiler cannot determine all information needed for a function call at compile-time.

**Compile time binding ko rokne ke liye virtual keyword ka use karte hai.**

**Pure virtual function**

**Q what is pure virtual function?**

**Ans:-bina body vale function ko pure virtual function kehte hai.**

* **Pure virtual function ko 0 se initialize karna compulsory hai.**
* **Pure virtual function ko virtual keyword se declare karte hai.**
* **Yadi kise bhi class me ek bhi pure virtual function hai to use abstract class kahte hai.**
* **Ham kabhi bhi abstract class ka object nhi bana sakte hai.**
* **Abstract class ke data ko access karne ke liye inheritance ka use karna compulsory hai.**
* **Ham ek class me kitne bhi virtual function bana sakte hai only Sub class me super class ke sare vitual function ki body banana compulsory hai.**
* **Yadi ham asa nhi karte hai to sub class bhi abstract ban jate hai.**

**Syntax:-**

**Virtual void show()=0;**

**Typedef**

**Typedef keyword ka use hum kise existing data ka new name rakhne ke liye kerte hai.**

**Ex:- typedef int manish;**

**Manish x=10;**

**Ab manish bhi ek int data type ka mana jayega int ki jagha manish ka usebhi ker sakte hai or int ka bhi.**

* **Mainly ye predefine ki definition change ker deta hai.**

**Inline function:-**

**C++ provides an inline function to reduces the function call overload. Inline function is function that is expanded inline when it is called . when the inline function is called whole code of inline function get inserted or substituted at the point of inline call .This substituted is performed by the C++ compiler at compile time. In function may increase efficiency if it small.**

**Internal function:-**

**The internal() method of stream manipulators in C++ is used to set the adjustfield format flag for the specified str stream. This flag sets the adjustfield to internal. It means that the number in the output will be padded to the field width by inserting fill characters at a specified internal point.**

**Syntax:**

**ios\_base& internal (ios\_base& str)**

**External function:-**

**External function is function which is declare in class without bodyand it’s body is declare out side the class with the help of [::] scope resolution operator .**

**Like return\_type class\_name :: method\_name(parameters list){}**

**Void A :: sum(int a,int b){}**

**Diamond shape problem:-**

**Q what is diamond shape problem?**

**Ans:-jab bhi hum hybrid inheritance ka use karte hai to ambiguse error ate hai jisse compiler confuse ho jata hai ki vo kise pahle access kare . ese hi diamond shape problem kahte hai.**

**Jab hum multipal inheritance and hierarchical inheritance ka ek sath use karte hai to ye diamond ki tarha ban jata hai.**

**Ex:-**

**class** A

1. {
2. **public** **void** display()
3. {
4. System.out.println("class A display() method called");
5. }
6. }
7. **class** B **extends** A
8. {
9. @Override
10. **public** **void** display()
11. {
12. System.out.println("class B display() method called");
13. }
14. }
15. **class** C **extends** A
16. {
17. @Override
18. **public** **void** display()
19. {
20. System.out.println("class C display() method called");
21. }
22. }
23. //not supported in Java
24. **public** **class** D **extends** B,C
25. {
26. **public** **static** **void** main(String args[])
27. {
28. D d = **new** D();
29. //creates ambiguity which display() method to call
30. d.display();
31. }
32. }

* **Hum ese virual keyword ka use ker ke slove ker sakte hai class B and class C ko virtual declare ker ke. Esse compiler class B and class C ko ignore kr ke dyrect class A se communicate karega.**
* **Dynamic memory allocation crate hota hai new keyword ki help se or delete hota hai delete keyword ke help se. C me free keyword ki help se hota hai.**
* **C++ me ham kabhi bhi array me constant variable nhi rakh sakte hai.**
* **C++ me big data\_type small me store ho sakta hai.like int x=10.9;**
* **But java me nhi ho sakta hai incapatibale error dega**

# **Templates**

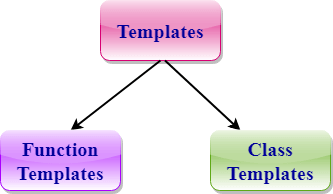
**Independent of data type is called template. Kyo ki template kise bhi type ke data type ko at a one time me support karta hai.**

**Template is a one type of frame which defines it’s actual meaning in C++. It create appropriate code at the time of exeqution.**

**A C++ template is a powerful feature added to C++. It allows you to define the generic classes and generic functions and thus provides support for generic programming. Generic programming is a technique where generic types are used as parameters in algorithms so that they can work for a variety of data types.**

**Templates can be represented in two ways:**

* **Function templates**
* **Class templates**



**Function Templates:**

**We can define a template for a function.it is known also as generic function. For example, if we have an add() function, we can create versions of the add function for adding the int, float or double type values.**

**Class Template:**

**We can define a template for a class. .it is known also as generic class. For example, a class template can be created for the array class that can accept the array of various types such as int array, float array or double array.**

### **Syntax of Function Template**

1. **template** < **class** Ttype> ret\_type func\_name(parameter\_list)
2. {
3. // body of function.
4. }

**Syntax of class template**

1. **template**<**class** Ttype>
2. **class** class\_name
3. {
4. .
5. .
6. }

**Points to Remember**

* **C++ supports a powerful feature known as a template to implement the concept of generic programming.**
* **A template allows us to create a family of classes or family of functions to handle different data types.**
* **Template classes and functions eliminate the code duplication of different data types and thus makes the development easier and faster.**
* **Multiple parameters can be used in both class and function template.**
* **Template functions can also be overloaded.**
* **We can also use nontype arguments such as built-in or derived data types as template arguments**.

**Q what is object type array?**

**Ans:-** **eska yahi benefit hai ki hum more than one employ ka record print karva sakte hai. esme class ka object array type ka bana ker use loop ke through access karva sakte hai.**

**yadi hume single employee ki data print karvana hai to simple object bana ker bhi ker sakte hai but yadi 1000 ka karvana hai to etne object na bana ker ek array type ka object bana ker loop se access ker sakte hai.**

**Q what is Bubble short?**

**Ans:** **In Bubble sort, Each element of the array is compared with its adjacent element. The algorithm processes the list in passes. A list with n elements requires n-1 passes for sorting. Consider an array A of n elements whose elements are to be sorted by using Bubble sort.**

**Algorithm** :

* **Step 1**: Repeat Step 2 For i = 0 to N-1
* **Step 2**: Repeat For J = i + 1 to N - I
* **Step 3**: IF A[J] > A[i]  
  SWAP A[J] and A[i]  
  [END OF INNER LOOP]  
  [END OF OUTER LOOP
* **Step 4**: EXIT
* **If number is grater then sweep.**

# **Files and Streams**

**In**[**C++ programming**](https://www.javatpoint.com/cpp-tutorial)**we are using the iostream standard library, it provides cin and cout methods for reading from input and writing to output respectively.**

**To read and write from a file we are using the standard C++ library called fstream. Let us see the data types define in fstream library is:**

|  |  |
| --- | --- |
| **Data Type** | **Description** |
| **fstream** | **It is used to create files, write information to files, and read information**  **from files.** |
| **ifstream** | **It is used to read information from files.** |
| **ofstream** | **It is used to create files and write information to the files.** |

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*END\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***