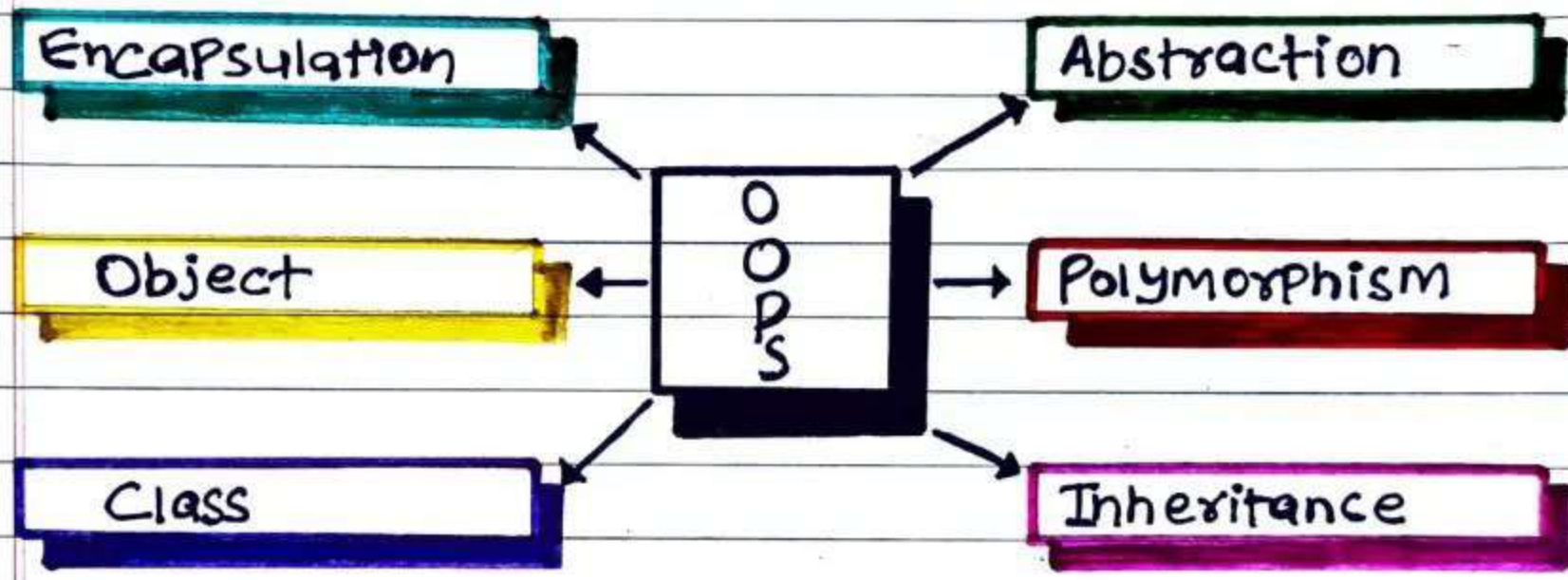
OOPS in C++

The main aim of OOP is to bind together the data and the functions the operate on them so that no other part of the code can access this data except this function.

TYPES OF OOPS.



⇨ **Class :** It is a user defined data types, which holds its own data members and member functions, which can be accessed and used by creating an instance of that class.

⇨ **Object :** When a class is defined no memory is allocated but when it is instantiated (i.e., Object is created) memory is allocated.

⇨ **Encapsulation :** In OOP, Encapsulation is defined as binding together the data and the functions that manipulates them.

⇨ **Abstraction :** Abstraction means displaying only essential information and hiding the details.

- Abstraction using classes
- Abstraction using Header files (math.h → Pow())

⇨ **Polymorphism :** In simple words, we can define polymorphism as an ability of a message to be displayed in more than one form.

- Operator overloading
- function overloading
  - ↳ int sum(10,20,30)
    int sum(10,20)

⇨ **Inheritance :** The capability of a class to derive properties and characteristics from another class is called inheritance.

- Subclass
- SuperClass
- Reusability

Dynamic Binding :

In dynamic binding, the code to be executed in response to function call is decided at run time.

Constructors :

A constructor is a member function of a class which initializes objects of a class. In C++ constructor is automatically called when the object creates.
It has same name as class itself.
Constructor don't have a return type.

1. Default constructor (NO parameter passed)
2. Parametrized Constructor
3. Copy constructor.

Destructor in C++ :

Derived class destructor will be invoked first, then the base class destructor will be invoked.

Access Modifier :

Public :- can be accessed by any class.
Private :- can be accessed only by a function in a class (inaccessible outside the class).
Protected :- It is also inaccessible outside the class but can be accessed by subclass at that class.

Note :- If we do not specify any access modifier inside the class then by default the access modifier for the member will be private.

Friend class :

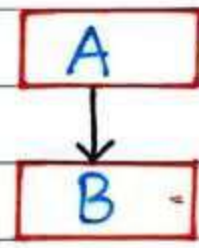
A friend class can access private and protected members of other class in which it is declared as friend.

Ex :- friend class B;

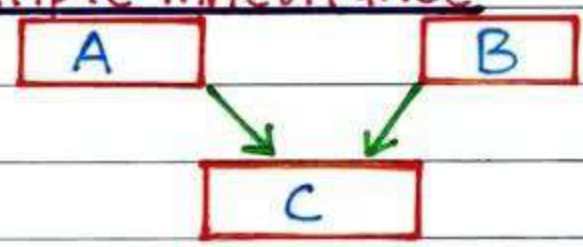
- **Inheritance**

```
Class Subclass : accessmode . baseclass
{ ___
}
```

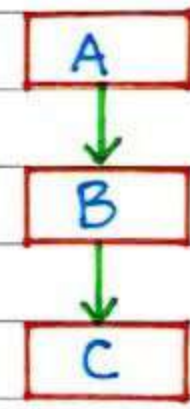
1. Single inheritance :



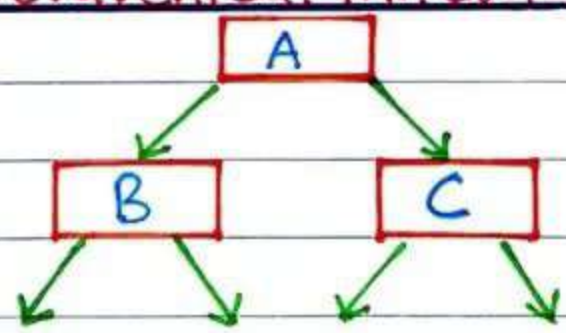
2. Multiple inheritance



3. Multilevel



4. Hierarchical inheritance



5. Hybrid

Combination of one or more type.

- **Polymorphism**

→ Compile time Poly → Operator overloading.
→ Function overloading.

→ Runtime Poly

↳ Function overriding occurs when a derive class has a definition of one or more members of base class.

Advantages of Data Abstraction

- Avoid code duplication and inc. reusability.
- Can change internal implementation of class independently.

Structure Vs Class

Most important difference is security.
A Structure is not secure and cannot hide its member function and variable while class is secure and can hide its programming & designing details.

Local classes in C++ :

A class declared inside a function becomes local to that function and is called local class.
All the methods of local class must be defined inside the class only.

Virtual function and Runtime Polymorphism :

⇒ A virtual function is a member function which is declared with a base class and redefined (overriden) by derived class. functions are declared with virtual keyword base class.

⇨ Runtime Polymorphism, also known as the Dynamic Method Dispatch, is a process that resolves a call to an overridden method at runtime. The process involves the use of the reference variable of a superclass to call for an overridden method.

Exception Handling in C++ :

try : represent a block of code that can throw an exception.

catch : represent a block of code that get executed when error is thrown.

throw : Used to throw an exception.

There is a special catch block → catch(...)
It catches all types of error.

Inline Function

Inline is a request not command.
It is function that is expanded in line when it is called. When the inline function is called, whole code get inserted or substituted at the point of inline function call.

```
inline return-type function( )
{ ==
}
```

Function Overloading

Function overloading is a feature in C++ where two or more functions can have same name but different parameters.

```
Void print (int i   )
{      cout<< "Here is int " <<i <<endl;
}
Void print (float i )
{     cout << "Here is float" <<i << endl;
}
 int main
{          print (10) ;
           print (10.12) ;
}
```

Differences b/w C and C++

C	C++
• C supports procedural program.	• C++ is known as hybrid language, because it support both procedural and object orlented Programming.
• As C does not support the oops concept so it has no support for polymorphism, encapsulation and inheritance.	• C++ has support for Polymorphism, encapsulation and inheritance as it is an OOPS language.
• C is a subset of C++	• C++ is a superset of C.

C	C++
• C contains 32 Keywords	• C++ Contain 52 keywords (public, private, protected, try, catch, throw)
• C is a function driven language.	• C++ is an object driven language.
• Function and operator overloading is not support in C.	• C++ supports function & operator Overloading.
• C does not support exception handling.	• C++ supports exception handling using try and catch.

- Structure is a collection of dissimilar elements.

- **Static Members in C++**
  - **Static variable in a function :**

When a variable is declared as static, space for it gets allocated for the lifetime of the program. (default intialized to 0)
Even if the function is called multiple times, the space for it is allocated once.

- **Static variable in a class :**

> Declared inside the class body.
> Also known as class member variable.
> They must be defined outside the class.
> Static variable doesn't belong to any object, but to the whole class.

➢ There will be only one copy of static member variable for the whole class.

Ex :

```
Class Account
{       private :
            int balance;
            static float roi ;
        public :
            void setBalance(int b)
        { balance = b ;   }
};
```

// intialised outside class

```
float Account :: roi = 3.5 f;
void main
{
    Account a1;
}
```

- Object can also be declared as static.

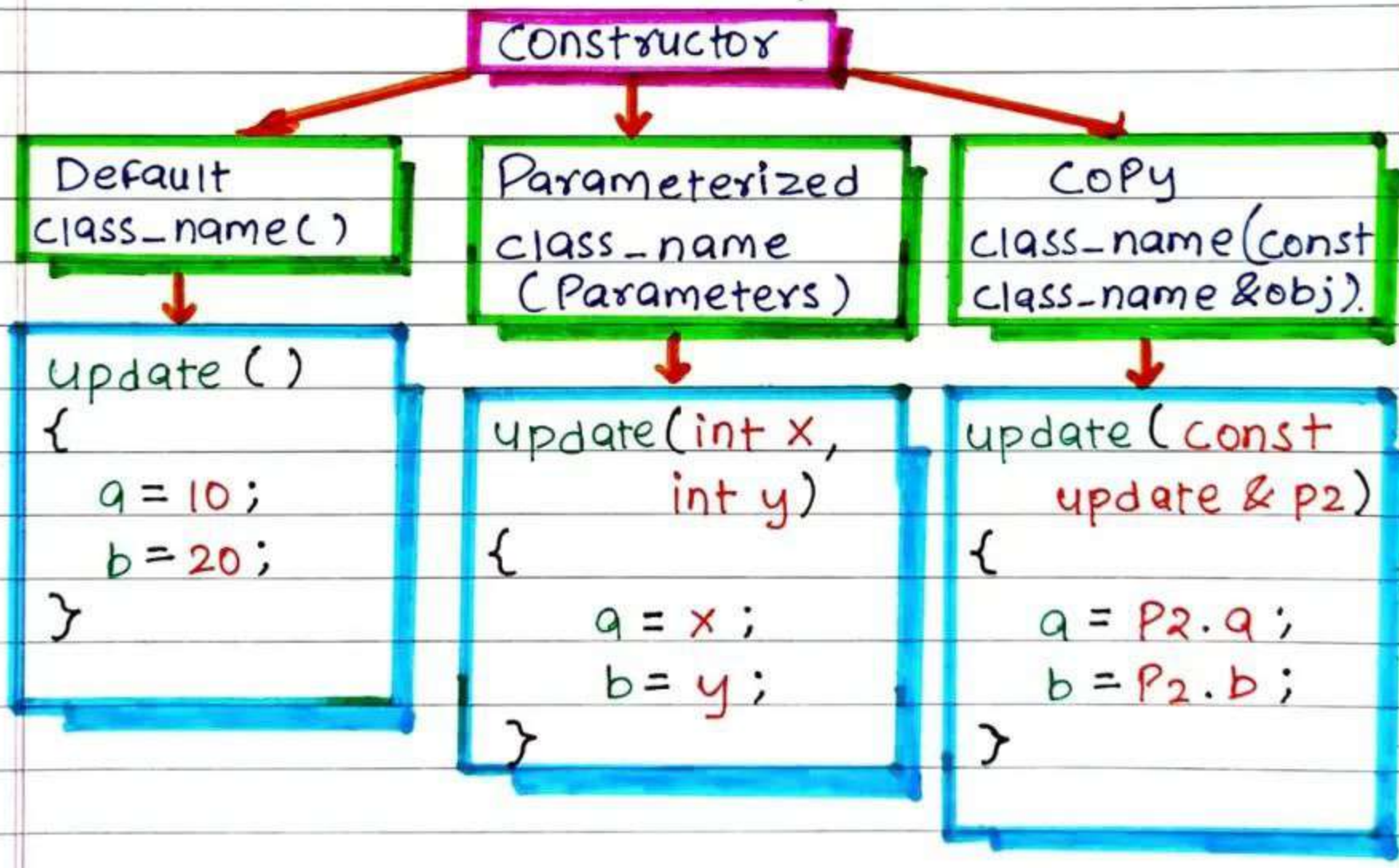
  Static Account a1;

Static Function in a Class

Static member functions are allowed to access only the static data members or other static member functions.

Constructors :

- Constructors is an special member function of the class. It is automatically invoked when an object is created.
- It has no return type.
- Constructor has same name as class itself.
- If we do not specify, then C++ compiler generates a default constructor for us.



Compiler generates two constructor by itself.

1. Default Constructor
2. Copy Constructor.

But if any of the constructor is created by user, then default constructor will not be created by compiler.