CPP

1) class and objects
- objects are entities in the real world
- class is like a blueprint of these entities.

2) Access specifier
1) Private : data & methods accessible inside class
2) Public : " accessible to everyone
3) Protected : " accessible inside class & to it's derived class

3) 4 Pillars

i) Encapsulation

1 => Encapsulation is wrapping up of data & member function in a single unit called class.

2 => Encapsulation is the concept of hiding data and allowing access only through controlled interface using class members.

- data hiding
- Ex...

a) Constructor
- Special method (function for syntax specially) invoked automatically at time of object creation.

Ex:- ...

types
1) Parameterized
2) Non-parameterized (default)
3) Copy

Ex:- ...

imp this (->)

this is a special pointer in c++ that points to the current object

Ex:- Person p1;

this -> properties = *p1.properties

- give a ex of cons and destr.. sequence in inheritance.

b) Destructor

=> A destructor is a special member function of a class that is automatically called when an object is destroyed.

Ex:- used in Dynamic memory allocation

ii) Inheritance

- Inheritance means properties & member functions of base class are passed on th to the derived class.

optional

* mode of Inheritance

Base class	Derived class private mode	Derived class protected mode	Derived class public "
private	not Inherit.	not "	not "
protected	private	protected	protected
public	private	protected	public

* types of inheritance

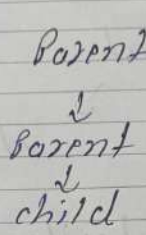
1) Single Inheritance

Ex:- parent

child

2) multi-level Inheritance

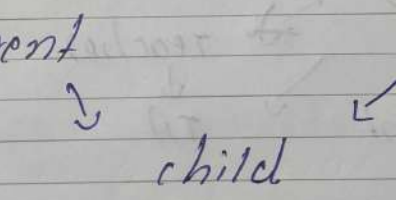
Ex.



def :- this inheritance is a type of inheri... where a class is derived from another derived class.

3) multiple Inheritance

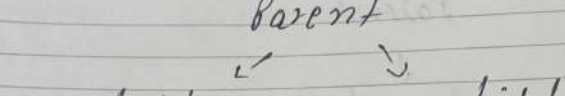
EX



def :- multiple inheritance is when a class inherits from two or more base classes at the same time.

4) Hierarchial Inheritance

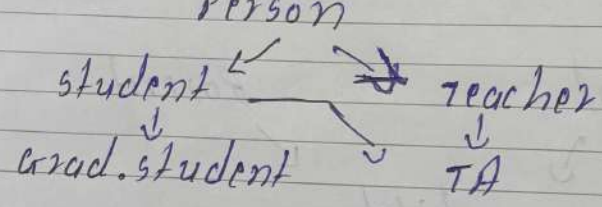
Ex:-



def :- Hierarchical inheritance is when multiple classes inherit from the same base class.

5) Hybrid Inheritance

Ex :-



iii) Polymorphism

=> Polymorphism is the ability of objects to take on different forms or behave in different ways depending on the context in which they are used.

i) Compile time

a) Constructor overloading
b) function overloading
c) operator overloading

Ex:- Provide ex in vs code.

ii) Run time

a) function overriding

- Parent & child both contain the same function with different implementation.

i) with pointers
ii) with references

* virtual functions

=> A virtual function is a member function that you expect to be redefined in derived classes.

Ex code.

1) Compile time

def :- executed is determined at compile time.

2) Run time

def :- executed is determined at Run time.

iv) Abstraction (Data Abs I vaar joi revu)

=> Hiding all unnecessary details & showing only the important parts.

a) Abstract class

- blueprint for other class

Ex code.

* static keyword and function (jota ma bhulvu
* friend function