2/04/2	023
4 billa	rs of OOPs
	n say everything is abstraction. 4 pillar
00 PS	s are :
1) Encaps	
a) Inheri-	tance
3) Polymo	rphism
	ction - Implementation hiding
Encaps	sulation
Wrapp	ing of data members & member functions capsule (class) is known as encapsulation
in the	capsule (class) is known as encapsulation
By this	s we are trying to achieve data hiding.
	apsule Member functions
<u>Class</u>	
and the state of t	Data members Data members
2	nember functions (Class > Wrapper
Parl	data members of
rectect	t encapsulation is that in which everything as private. We can access these private
Is set a	s via setter & getter.
R. thi	is we are increasing the security, like
NAP OF S	type of data to show & what type
el data	a not to show.
Note + Abstr	act = not going into much details
	I am made a description of the said
Ex→ class A	nimal & management
	public :
	int age;
	void eat () 2  cout << "Eating";
3;	3 Cout 22 "Earing"
	Scanned With Co

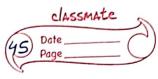
	Inheritance
	There will be a super/parent/base class and
	there will be subclass /child/ derived class.
	Derived class will inherit the properties of
	base class. There can be multiple derived
	class. There can be manipre
	classes as well.
	Syntax
	class child: mode parent
	Public, privace of the
$\epsilon_{\Upsilon}$	Class Drie 15
<u>C</u>	class Animal {
· - H	public:
	int age ; void eat () {
U	Voia eat () [
	cout << "Eating";
	3 ;
	class Dog: public Animal {
	Class Dog & pablic Tilling
1.00	3; and a girl to all some state and a state of
7	main () {
	Dog dl
	dl·eat();
5.4	3 1
1	
	0/b - Eating
	The above code did not give an error as
	the data members & member functions gets
	the data members & member functions gets inherited to the Dog class i.e. child or
	derived class.
	$(1, 1, 2, 3)^{11} > 2 1$

			W.	
	Chart of inheritar	nce		
	Base class	Til	-1 in boxita	D.C.4
	access modifier	Public	Protected	Private
	Public	Public	Protected	Private
	Protected	Protected	Protected	Private
	Private	NA	NA	NA
	NA - Not accessi	9		
Ext	class Animal {	Base class as	cess modifier	_
	int a	ge 5		
	35 7 To	the of inheritar	nce	
	class Dog : public	Animal {		
	0		,	]
	int age i	→ public	(Inheui	ted
	void eat	() { 3 ->	bublic I from	base
	3 ;		class	
	when	71 -001	1.6.	
1	Hence we were de not getting an ern	oing Direct	Lunction We	l in
	not getting an evi	on as ear	D TOTAL TOTAL TOTAL	<u> </u>
	public mode.			
	Protected mode	(access mo	difier)	
	Accordible in devi	ved class. 1	his is combin	nation
	of private + (only	accessible	in the deriv	ed
	class).			
90	Private - not inhe	rited		
	Protected - inher	ited		

Scarined With Carill

	Types of inheritance
	Single Level inheritance -> There is one base class and only one child class.
	Parent
T I	Child
	Single level inheritance
$\in x_{1}$	class car {
	Cac
	class rapid: public car { Rapid
	3;
	Dominal Silver
Note =	Inheritance is basically is-a relation. We
	can say rapid is a car.
	derived base class
	Class
<u> 2)</u>	<u>Multilevel</u> inheritance
	Dan - 1
	Parent
	in a trace Child
	( manifestation of Estate to the same of the
	GrandChild
1	, was a series of the last of
	There can be many levels.
	There can be many levels. Here also is-a relationship will be followed.

Scarnieu with Cam



	W.
EXT	class Fruit { 3;
	class mango: public Fruit { 3 ; class alphanso: public mango { 3 ;
	class alphanso: public mango { 3;
Note	Hence we can say alphanso is a mango &
	mango is a fruit.
3)	<u>Multiple inheritance</u>
	Δ
100	B, B
	$\begin{array}{c} \longrightarrow & \subset & \longleftarrow \\ \end{array}$
	There are two base classes and single child or
	derived class.
$6x \rightarrow$	class A { 3 ;
	Class B & 3 j
	class C : public A, public B { 3 ;
	There is a problem in the above inheutance.
18-11	that what if there are same Items in A&
	B class & hence compiler can get confused
	here. This is known as diamond problem 4
-0.	we can solve it by scope resolution operator.
1996	Cobj : susing var-name of class B cout << obj . B :: var-name i
	cout << obj. B: Var-name
	7
	Diamond
	problem
	FOBICITE

ocanneu wiin caM

4)	Hierarchical Inheritance
	Base Class  Childl Childl  Single base class, multiple derived classes.
	class caré 3 ; class scorpio: public caré 3 ; class maruti: public caré 3 ;
5-11	Car Scorpio maruti Scorpio is a car and maruti is a car.
5)	Hybrid inheritance - It is the mixture of above studied inheritances.
	Single  level  inheritance  Polymorphism - Important for interviews  Poly - many
-	morph-forms This means existing in many forms.
- 11	

Scarnieu with cam

· ·	~
70	There are 2 types of polymorphism namely
	compile time & run time polymorphism.
	Compile time polymorphism It is based on operator overloading &
	It is based on operator overloading &
	Function overloading.
*	Function overloading
A STATE OF	class Maths {
Thype	public:
	int sum (int a, int b) {
-	return a+bi
	11. <u>9</u>
	Int sum (inta, int b, int c) 2
<u></u>	retwin a + b + c i
	· · · · · · · · · · · · · · · · · · ·
	int sum (int a, float b) {
	return a+b;
101 Jr	3)
100	main O 2
7.2	Maths obji
	cout << obj. sym (2,3); // Ist func cout << obj. sym (2,3,5); // 2nd func
	cout << obj. sum (2,5.6f)) // 3rd func
	2 COURT - SISTER
	In function overloading, return type of function should be same.
	Julian should be same.
	TUNCTION STORES
*	Operator overloading
	Operator overloading  For example using + operator, we can print  the difference also & this can be done via
	the difference also & this can be done via
	The Supplies
and the same	Scanned With Car

Scarined With Caril

	Oberator avenue adias
	Operator overloading.  Syntox perator symbol
	return type operator () {
	recuire egge operator ()
	3 awwent obj
	a)+B input parameter member function
$\epsilon_{x}$	member function
	Class Param {
	public:
	int vali
	Void operator + (Param 4 obj2) {
	int value 1 = this + val;
F	
	int value 2 = Obj2·val; cout << (value 1 - value 2);
	3
	3;
	main () {
	Payam objl, obj2;
	Obj1.val = 7)
	Obji: Val = 2;
1 61	Obj1 + Obj2 j → 5
5.71	31,15 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
Vote →	Objl =) current object
	+ 7 member function
	Obj2 7 input parameter
	Le sant montas construires
	This is basically oberator overloading as
	+ operator is existing in many form
	This is basically operator overloading as + operator is existing in many forms such as addition of numbers, objects, concatenation
A 6	etc.
	and the second s

<del>ocanne</del>u wiin oaM