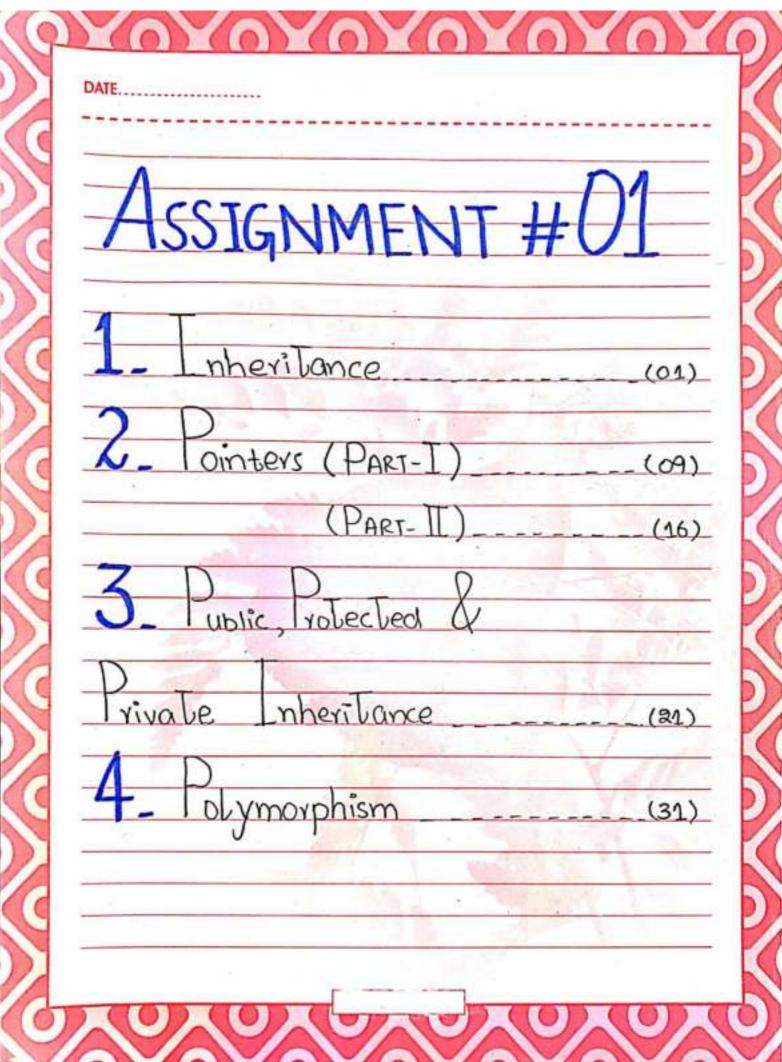
OYOYOYOYOYOY
DATE
SUBMITTED BY:
"SADAF SALEEM"
#2929
CO OND (A A)
CS-2" (M)
CCT DO
CS1-302
SUBMMITTED O:
- « C V C «
SIR KHURRAM SHAHZAD SL."
GC UNIVERSITY FSD
OIC OLATATION LON



	QUESTION	NO 01
"INHERITANCE"		
-itJu	level and Mu	Hiple inheritance:
ntroduct	Hon:	
		more than one classe
		nave properties but of
_	base classes.	1
, Base	CHOSSES KEDY ONE	hanged by this process
Example:		e' lette !
	Transmitter	Reciever
	IVCIVI MITTER	II P C PVPV
	Tra-omijior	REGIONEL
		pedice.
		REGION TO THE PARTY OF THE PART
	transmit()	vecieve.

Yanımi	ther" and "Reciever	classes both
Examp	ne syntax of C++:	
	Transmitter.	Reciever
		All bullets
	transmit ()	reclever)
class	Phone : Public Tro	The state of the s
	ple program	3.367
	ass transmitter !	female and the second s
Public		
	void transmit () {	THE REAL VIEW
	cout ec " recieving	" ccend1;}
	The state of the s	- 1

Pg: 02

my PI	none - transmit (); 11 Transmitting
-	none recieve (); Recieving.
Adva	ntages of Multiple Inheritance:
r	eatures of more than one classes can be
	into a single class_
(ode duplication can be avoided.
Proble	me in multiple Inheritance:
	HW OCTUITY .
	Digmond shape problem-
	Diamond shape problem.
Ambi	
	guity in Multiple Inheritance:
It	guity in Multiple Inheritance: a same member is coming from more
It	guity in Multiple Inheritance: a same member is coming from more one base class then it can create
It than umbi	guity in Multiple Inheritance: a same member is coming from more one base class then it can create guity when using it in child class.
It than umbi	guity in Multiple Inheritance: a same member is coming from more one base class then it can create
It than umbi	Diamond shape problem. "quity in Multiple Inheritance: a same member is coming from more one base class then it can create quity when using it in child class. "quity when using it in child class."
It than umbi	Diamond shape problem. Guity in Multiple Inheritance: a same member is coming from more one base class then it can create guity when using it is child class. Aprile: Transmitter I did id
It than umbi	Diamond shape problem. Guity in Multiple Inheritance: a same member is coming from more one base class then it can create guity when using it is child class. Aprile: Transmitter I did id
It than umbi	Diamond shape problem. "guity in Multiple Inheritance: a same member is coming from more one base class then it can create guity when using it in child class. Aprile: Transmitter Reciever

Pq 1 03

	_	
Phone		
Phone my phone;	id	97
	id	
	my phone;	
Example Program:	As In the	= 1/1
elass Transmitter {	46 300	reals in
Protected:		
- int id;		W TORK
Public:		Vi i
Void transmit () {	- 300	A TENT
couter "Recieving" ce	end 1: 3	Victor 1
3.	,,,,,,	100
3)	7 7 7 19 20 5	1 11/16/
Class phone: Public Trans	witter public Re	cieves &
Public:	, ,	t
yoid print IDC)		1
cout « printing to ID	in alone class :	"cid "
Cour Himming to the	11) phone costs	end 1;
3	diam'r.	2,001,
9	I will be a second	7.
1		

Solution of Ambiguity in Multiple Inheritance class Transmitter { Protected; int id; Public: void transmitting "ccendl; } class Reciever { Protected: int id; Public: void recieve L) { void recieve L) { void recieve L) } class Phone: Public Transmitter, public Reciever Public:		Error: - id is ambiguous.	
class Transmitter { Protected: int id; Public: void transmitting "ccendl; } class Reciever { Protected: int id; Public: void recieve L) { cout cc Reciever L) { cout cc Reciever L) } cout cc Reciever L) } class Rhone: Public Transmitter, Public Reciever	Disambiguation:		
Protected; intid; Public: void transmitting "ccendl; }. class Reciever { Protected: intid; Public: void recieve () { void recieve () { void recieve () { vout cc recieving "ccendl; } }; class Phone: Public Transmitter, public Reciever	Solution of	Ambiguity in Multiple Inheritance	
Public: Void transmit () & coutec "Transmitting" ecendl; } class Reciever { Protected: int id; Public: Void recieve () { cout < c recieving " ecendl; } }; class Phone: Public Transmitter, public Reciever	class Transmit	Hev {	
Public: Void transmit () & coutic "Transmitting" exendl; } class Reciever { Protected: int id; Public: Void recieve () { cout < c recieving " exendl; } j; class Phone: Public Transmitter, public Reciever	Protected:		
Public: void transmit () { cout cc "Transmitting " ccendl; } }; class Reciever { Protected: int id; Public: void recieve () { cout cc recieving " cc endl; } }; class Phone: Public Transmitter, public Reciever			
class Reciever { Protected: int id; Public: void recieve L) { cout cc recieving " cc endl; } }; class Phone: Public Transmitter, public Reciever		Marie Alleria	
class Reciever { Protected: int id; Public: void recieve L) { cout cc recieving " cc endl; } j; class Phone: Public Transmitter, public Reciever	The Particular Control of the Particular Con	smit () &	
class Reciever { Protected: int id; Public: void recieve L) { cout cc recieving " cc endl; } class Phone: Public Transmitter, public Reciever			
Protected: int id; Public: void recieve () { cout cc recieving " cc endl; } }; class Phone: Public Transmitter, public Reciever	1;		
Protected: int id; Public: void recieve () { cout cc recieving " cc endl; } }; class Phone: Public Transmitter, public Reciever	class Recieves	{	
Public: void recieve L) { cout < c recieving " < c endl; } class Phone: Public Transmitter, public Reciever	Special Control of the Control of th	The true of the true of with	
Public: void recieve L) { cout cc recieving " cc endl; } }; class Phone: Public Transmitter, public Reciever	int id:		
void recieve L) { cout cc recieving " cc endl; } }; class Phone: Public Transmitter, public Reciever	AND THE RESERVE TO SERVE THE PARTY OF THE PA		
class Phone: Public Transmitter, public Reciever		iere L) }	
class Phone: Public Transmitter, public Reciever			
	3;		
Public:		: Public Transmitter, public Reciever	
yoid print IOC) {			

20:05

DATE	
cout"	Printing ID in phone class: "ecendia co
"TD L	om Transmitter: " 44 Transmitter: id 4
end	ce "ID from Reciever" as Reciever :
id Le	end 1;
3	
2.	
35	Cons Parolution populator
	Scope Resolution operator
2.33	also called Disambigution
	operator-
Ambiguity	in Multiple Inheritance
Example Pr	Dyram # 02:-
class	Transmitter
Protected:	
int ic	d:
Public:	
void to	ransmit () L
10utes	"Transmitting" << endl; }
3:	Address Type State of Manager
37	
class	Decienes {
Class Protected:	Decienes {
AND DESCRIPTION OF THE PARTY OF	

06

DATE	AP
void recieve() {	
couter "Recieving" ecendlis	
void printID ()	
cout cc "Reciever ID: " ceid co	endl: ?
1;	, 1
class Phone: Public Transmitter; Pub	lic Recieved &
3;	9
void main () {	16 20 B
Phone my phone;	71.20
my Phone . print (ID (); Error = Dr	int ID is
amb	lauous.
Disambiguition:	1 2 2 2
Example Program #02:	- 8 -880
class Transmitter }	UNEF
Protected:	3 -4
int id;	
Public:	A North
void transmit () {	HA X5"
cout cc "Transmitting" ccende:	1
yold print ID () {	
cout ce "Transmitter ID?" ex ich La	endl; 1
3:	3.4
37	

Pq:07

class Recieve	ex {
Protected:	
; bi tai	
Public:	
void recieve	3 ()
coutce	" Recieving " exendl; }
void print I	203
	'Reciever ID: "id wendl; }.
3;	
3;	olic Transmitter, Public Reciever
void main ()	6 Cisambiguanos
phone m	y Phone;
	Fransmitter: Print IDO;
3. myPhone.	Recieves:: print ID ();
40	
11 Brint IDCO of -	Transmitter class
Yard Village Control	Recieves class
11 bring DC) of	

ODECT	ION NO.	02.
"POINTERS"		
	The state of the s	ple that hows
This addr	ess is the J	location of anoth
Example: Memory	vouiable in	ocation of anoth
Object. (Typically Example:	another vo	ocation of anoth
Example: Memory Address	vouiable in	ocation of anoth
Example: Memory	vouriable in memory	ocation of anoth
Example: Memory Address	vouriable in memory	ocation of anoth
Example: Memory Address 1000	vouriable in memory	ocation of anoth
Example: Memory Address 1000 1002	vouriable in memory	ocation of anoth
Example: Memory Address 1000 1002 1003	vouriable in memory	ocation of anoth

VOIV	nter Variables:
°+ '	If a variable is going to hold a pointer must be declared as such.
	A pointer declaration consists of:
	an * type
	and the variable name
	i.e; type * name;
-v.	
FXAV	
Cidi	nple;
e reci	mple;
	int * ptr;
bas	int * ptv;
	int * ptv;
	int * ptr;
bas	e type an* variable name
bas	int * ptr;
Poin	of type an variable name name name
Poi	nters variables cont. The base type of the pointer defines what
Poin	nters variables cont. The base type of the pointer defines what
Poin	nters variables cont. The base type of the pointer defines what
Poin	nters variables cont. The base type of the pointer defines what
Poin	nters variables cont. The base type of the pointer defines what

EX	ample:
	int *ptr >
	7 1
pas	e type an * variable name
The	pointer operators:
INTE	POTITIET OPERATORS.
1-	There are two special pointer operator.
e de la constante de la consta	
	(i) - & * (ii) - *
Service .	
2-	The & is a wary operator that return
	the memory address to its operand.
C * .	
EVI	imple:
plac	es into 'm' the memory address of count variab
Y	It was nothing to do with the value of count.
The	Pointer operators cont.
=>	there are two special pointer operators.
	Cir. *
	<u>Gi). 7</u>

	is unary operator that returns the
vodue	located at the address that follows.
Examp	ne:
T.	m contains the memory address of count
-	ov = *m; places the value of count into a
=> 70	nere are two special pointer operators:
	(i). ¢
	(ii), *
• Bo	the 'f' and 'x' have a higher precedence
than	all other gritmmetic operators except
the	Uwani i facile tick in the same
	mond winds, with conten they are
equal	unary minus, with which they are
equal	
equal	
Point	er Expressions:
Point As	with any variable, you may use a
Point As pointer	with any variable, you may use a on the right-hand side of an
Point As pointer assign	with any variable, you may use a on the right-hand side of an ment statement to assign its value
Point As pointer assign	with any variable, you may use a on the right-hand side of an
Point As pointer assign	with any variable, you may use a on the right-hand side of an ment statement to assign its value other pointer.

	main (-	
- 6	ixt xi			1-1-	-	
		1, x p2	;		1/2	
	B 7 =		-			
		P1;		36 1-		
-		cp2;		-		7-
2	return	0;	Mary and			
3.	- 10	1 1 1	I. V.	1. 1	The control	,
/* ^	-1. A -		P and I		1	50
16	TINT J	he add			XS	
		40	values	7		o Table
				-		10.00
-	Th PT	and P	g now	point	to x	2.0
		- N		point	to x	
		thme		point	to x	
ointer	Ave	thme	ic :			a
ointer	AVE with	thmet any va	ic:	Jou ma		a
ointer	AVE with	thmei	reable, y	Jou ma	A rice	eV.
Y As ointer	AVE with on the	any va	reable, y	Jou ma x	y use	eV.
Y As ointer	ave with on the e are a may	any va right only use	reables y thereof	Jou ma x	y use	eV.
Y As ointer	Ave with on the e are a may	any va right only use	reable, y hand hoo	Jou ma x	y use	eV.
Y As ointer	Ave with on the e are a may	any va right only use	reable, y hand hoo	Jou ma x	y use	eV.
Y As ointer	Ave with on the e are a may	any va right only use	reable, y hand hoo	Jou ma x	y use	eV.
ointex ointex They	Ave with on the e are unay	any va right only use iddition	reables y thereof	Jou ma x arithma nters:	y use	eration
cointex They hat you	Ave with on the e are unay	any va right only use ddition subtract	reables y thereof	Jou ma x arithma nters:	y use	eration

D7	++; causes pa to have the value 2002.
the	reason for this is that each time p2 is
incre	mented, it will point to the next integer.
*	Assume integers are a bytes long.
Poiv	nter Arithmetic (ont.
2	There are only two grithmetic operation
that	you may use on pointers.
	(i) Addition
	(ii) Subtraction.
EXO	mple:
	let po be an integer pointer with a
	ent value of 2000.
	11; cause p1 to have the value 1998.
	same reason is for subtraction.
	Assume integers are a bytes long.
=>	
=> that	There are only two grithmetic operation
=> that	There are only two grithmetic operation
=> that	There are only two grithmetic operation you may use on pointers:
=> that	There are only two grithmetic operation

Example:		
chair + cha (chay	1)3000;	
in+*1= (in+ *	3000;	
		13
- ch	3000	} :
ch+1	3001	J
enta	3009	}. i+1
ch+3	3003	1 24.2
chty	3004	3.6+2
chts	3005.	J
0 , 1	on pointers	- Comment of the Comm
that you may use in Addition (is Subtrace You are not limite	on pointers n. tion_	- Comment of the Comm
that you may use in Addition (i). Addition (ii). Subtract You are not limited of observement operators.	on pointers n. tion_	
that you may use in Addition (i). Addition (ii). Subtract You are not limited observement operators. Example: You may add	on pointers n. tion. d to the	increment and
that you may use in Addition (i). Addition (ii). Subtract You are not limited of operators. Example:	on pointers n. tion. d to the	increment and

=>	There are only two grithmetic operations that
you	may use on pointers:
u ,	is Addition.
	(ii) subtraction.
	You may subtract one pointer from another
12	order to find the number of objects of
thei	is base type that separate the two.
	All other arithmetic operations are prohibited
	6) - you may not multiply or divide pointers -
- (ii) - you may not add two pointers
	iii) you may not apply the bitwise operator
	them-
	white the state of
Pa	
	14-2
	14-2
	inters and Arrays:
Po	inters and Arrays: There is a close relationship b/w
Po	inters and Arrays:
Pow	there is a close relationship b/w hers and arrays.
Pow	inters and Arrays: There is a close relationship b/m Hers and arrays.
Pow	inters and Arrays: There is a close relationship b/m Hers and arrays. Char str[80], *p1;
pow Exa	inters and Arrays: There is a close relationship b/m Hers and arrays.

DATE				
TO Access	the fifth	element	in 'sty',	you
could wi		08		
	str [4]	*(P1+4)	
780	- 1			
Both state	ments will	return	fifth elem	en4.
Arrays o	f pointer	s :		
	rs may be	arrayed	like any	other
data type				
The d	eclaration	for an int	pointer av	ray of
size to is	· was	1115		
	int * x	1207;		will be
To assign			intoger 1	variable
To dailan	In the the	ind plans	aut of the	.0
	to the th	VIAO CIENN	en of h	
pointer gre	ay, write		777 16	THE TAX
			- 1 - 14 - 17 -	AND IN
60.75	x [2]	= & var;		
to find the	e value of	vory', wri	[e]x* »	1
Passing to	Function	12	LANA V	11100
If yo	u want to	pass an	array of p	on yers
into a f	unction, 40	u can us	e the sar	ne
into a f	at you	use to (pass other	grrays-

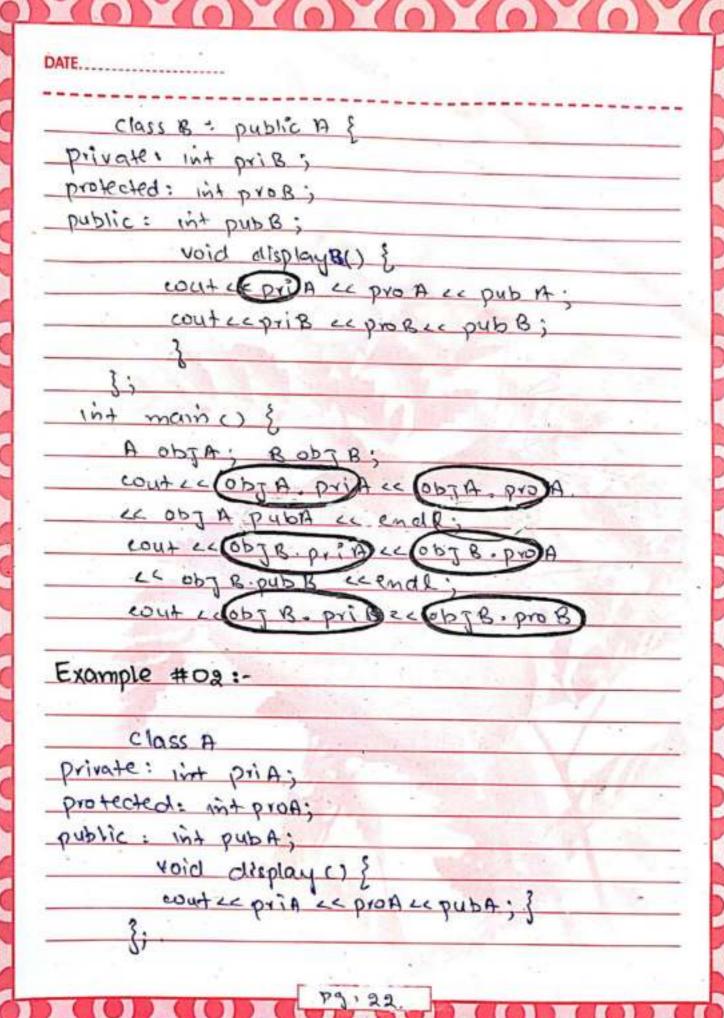
A 1/25	chan that can recieve array 'x
looks like	,
	d display_array (int *q[])
٤,	
l'n	<i>+ +</i> ;
// ININ	(t=0; t<10; t+1)
	outec * ov [t];
3	16 -30 -3 - 16 - 19 - 19 - 19 - 19 - 19 - 19 - 19
, and the second	
Multiple 1	indirection (Pointer to Pointer)
· You can	have a pointer point to another
· You can	have a pointer point to another points to the target value. Th
· You can	have a pointer point to another points to the target value. The called multiple indirection, or
· You can pointer that situation is	points to the target value. The called multiple indirection, or
· You can pointer that situation is	points to the target value. The called multiple indivection, or pointers.
pointer that situation is pointers to	points to the target value. The called multiple indivection, or pointers.
pointer that situation is pointers to	points to the target value. The called multiple indivection, or pointers.
pointer that situation is pointers to	points to the target value. The called multiple indivection, or pointers. pointers. variable dross
pointer that situation is pointers to	points to the target value. The called multiple indivection, or pointers.
pointer that situation is pointers to	points to the target value. The called multiple indirection, or pointers. pointers. variable value single Indirection
pointer that situation is pointers to	points to the target value. The called multiple indivection, or pointers. pointer variable value single Indirection pointer variable variable
pointer that situation is pointers to	points to the target value. The called multiple indirection, or pointers. pointers. variable dross Value Single Indirection
Pointer that situation is polinters to ad	points to the target value. The called multiple indivection, or pointers. pointer variable value single Indirection pointer variable variable

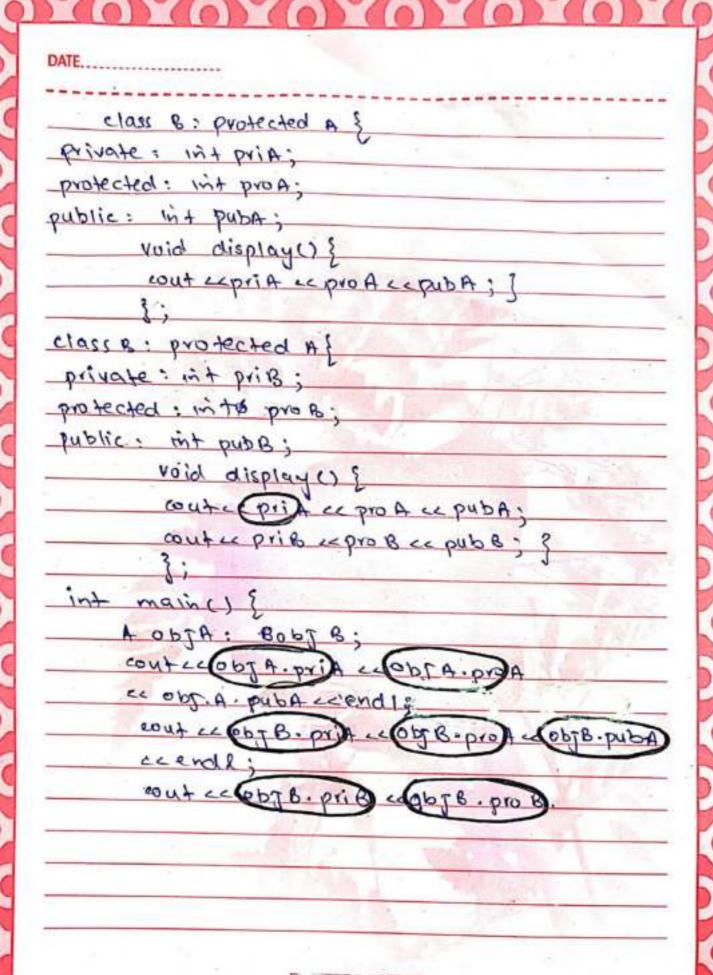
DATE	
	In the case of a pointer to a pointers
tuo 1	circl pointer contains the address of
400	second pointer which points to
obje	ct that contains the value desired,
Multi	ple indirection (Pointer To Pointer) cont.
. 1	to access the target value indirectly
point	ed to by a pointer to a pointer you apply the asterisk operator twice
must	apply the asterisk operator twice
Exav	nple:
-7.151	1 int x, *p, **9;
	X=10;
	P = 4 X;
	9=4p;
	couter ** q; /* print value of x *1
	1 / / / / / / / / / / / / / / / / / / /
11.	is delegad on a contract to an interes
Hen	e, p is declared as a pointer to an integer
anax	of or a bomper to a bounter up an
integ	er-
[a]al	

		e45 :			
				ven A	140
e:	I SUFF	A.T. b	1		
· int	+ ptv	= NULL;		4	-
nhters	cannot)	oe de-vi	eference	ol	
imple ?	ptv	is pointed	ol 40 v	all th	er
d igi	ice an	error.			
a the latest and the	-				i e
A WIND NO				100	
210		. 11 10		0 .55	17
P THE	NA.			TAR	
d	A Price	14.	179.5	1/ "	
	THE STATE OF	The same	-601	1	
	755		18/3	X 48 81	11
4	AND HOVE			A.B.	
	1000				
	ANT.	M. a		1	40
	e:	e: int + ptr	e: int + ptr = NULL; sinters cannot be de-re	e: . int + ptr = NULL; sinters cannot be de-reference	e:

: 20

		ion No		
"PUBI	TC PR	OTECTE	O, AN	ID
PRI	VHIE	INHE	RITAN	CF.
•		22 0 1000		
In herit	ance Acc	cess char	+:	
		Tv	heritance	
		-	A	-
	C # C - 1	Public	protected	Private
55	Public	public	protected	private
Acces	Protected	Protected	Protected	privale
4 [Private	Private	Private.	Private
Examble :	# O1 :-			
clas	S A E			/
	int priA	`		
	int pro			
	in+ pub A	5.547		
publics			(
publics	void d	() A wolds	-	

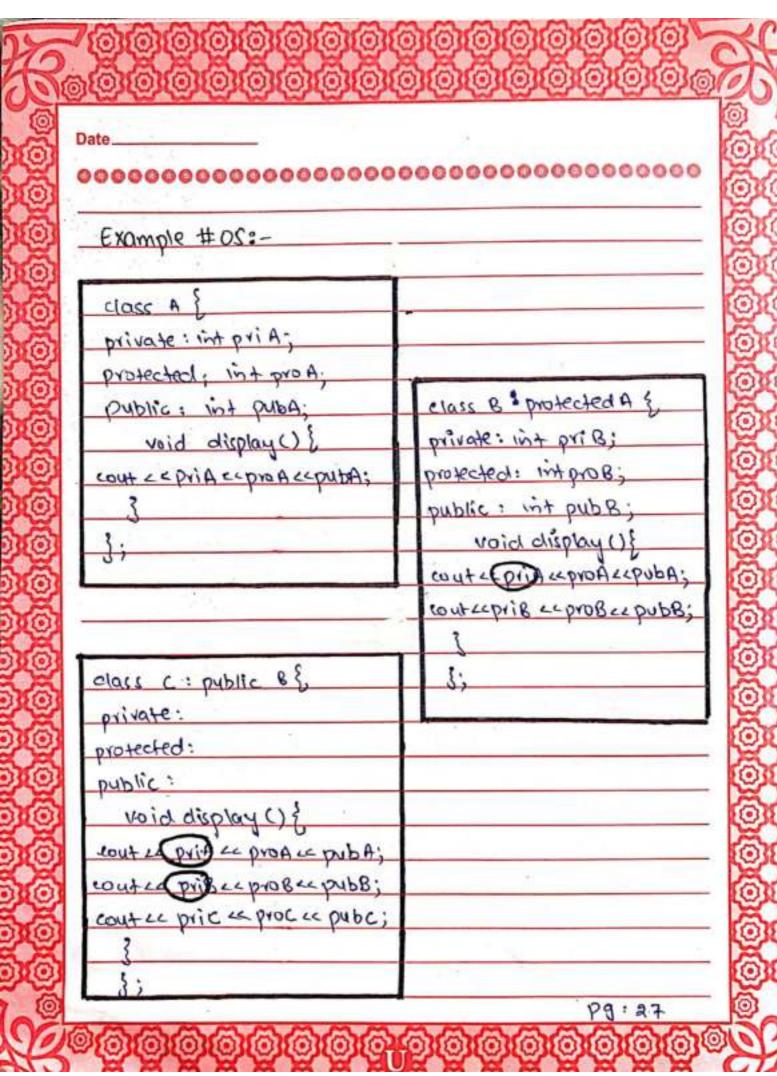




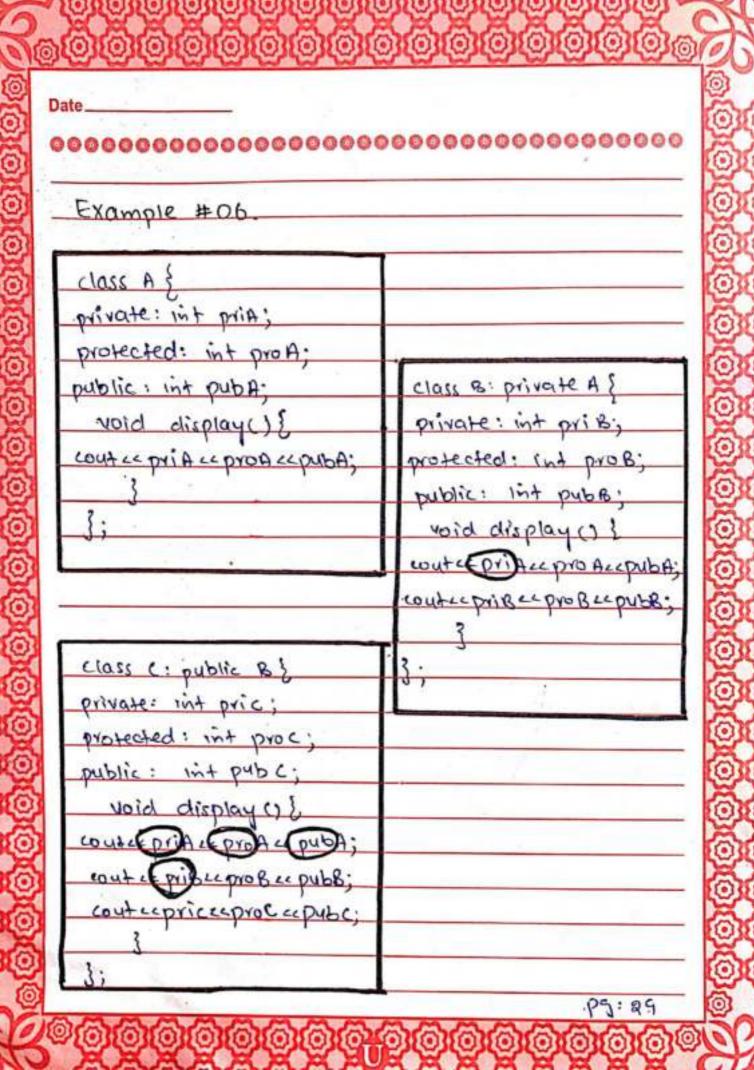
Example	#03:-		
cla	s A {		
private	: wit priA;		
protected	: Int pro A;		Maria
public :	int pub A;	1.00	
	oid display () {	ALL VIEW	
	coutecprises pr	OA CE DUB 4: ?	
34	3,		
class	B: private A !	4 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	16
	: int pris;		,
The second secon	: int pros;		FEVE
	int pubb;	A Phulls	The same
	rold display () {	A STATE OF B	V Black
C	ut - (priA) = proA	ce pubA:	San Pr
	out LE priBLE pro		SAK
		1. 3	
	3 - 10 - 10 - 10 - 10 - 10 - 10 - 10 - 1	1000	
int mo	in () {	(1
	BODTB;		- 8
	Les 667 A PriA	ACYO A TAO	All
	objA: pubA ccen		
	utce (obj B. pri) c		PT 8- PUD
	cendl;		
	utec 6678 priBec	01 78 0000	

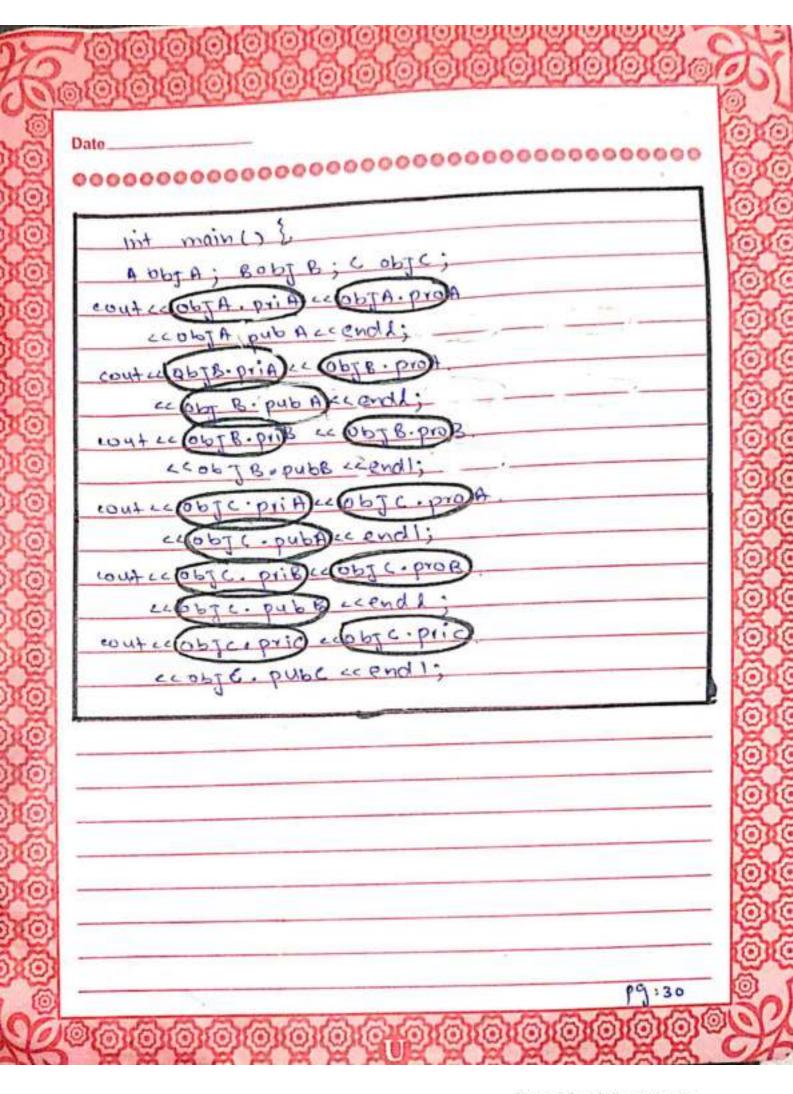
Example # 04:class A private: wit pri A; protected: int proA; chass B: public A public: int pubA; private: int pri 8; void display () { protected: int prob; cout cepriA LE proALE pubA; public: int pubB; void display () } coute (pri A c pro A ce pubA; contecpribe problepable; class c: public & { private: int pric; protected: int proc; public; int pub c; .void display () { cout eq priA cepro A cepubA; couted prib caprob ex pubs; cout ce pricecprocecpube;

A OBJA; BODJB; C OBJ COWLE ODJA. PriA) CC OBJA	<i>C</i> :
cource (obja. priA) ec (obja	C 3
cource (obj A. priA) ec (obj A	
	1. 1540)
LL ODJA: PUBALGENO!	
controple by d << 60	18. byold
ccobTB. pubA Knendi:	
couter 6 pd 8. Dis Bre	PLRDAOR.
.ccoby 8. pubb ecen	dl.
cout ecobje. prisece	J.C. DAO DI
coutecobje. pubA cceno	TC. PTOR
cont c. baple section	ndl:
contecopTe. pridecop	TC-PYO C
ccobj C - pubc ccen	dl;
	THE RESERVE
APPLY TO THE PARTY OF THE PARTY	the second second
THE SELECTION OF THE PERSON OF	TOTAL STATE
A CHARLES	NONE /
	TAND THE VIEW
	THE RESERVE
The state of the s	
The state of the s	7



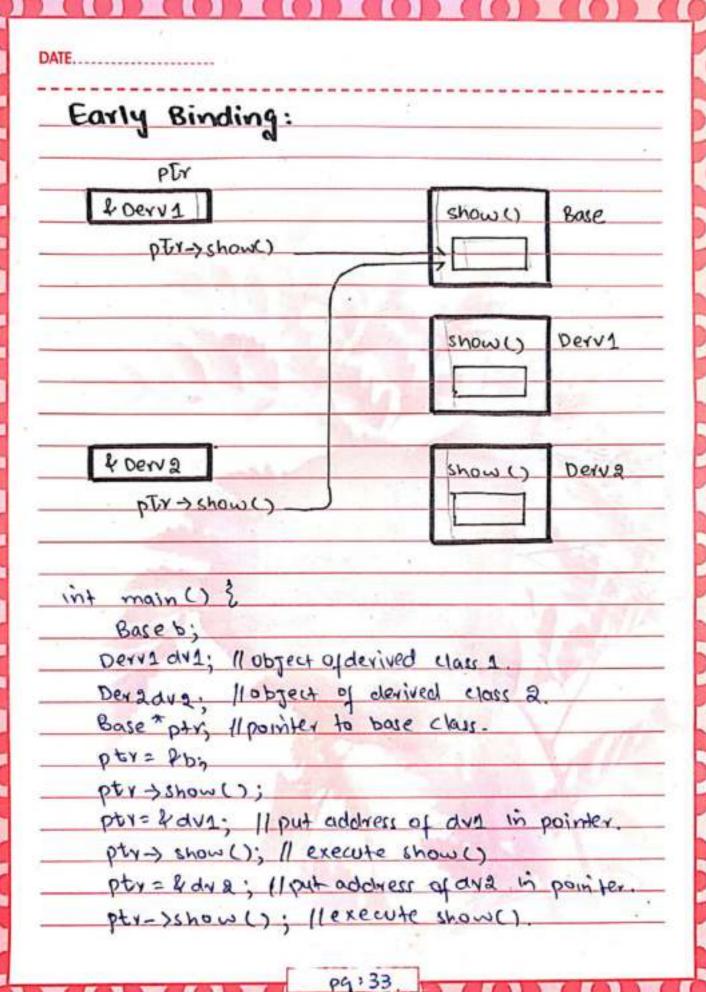
Date	
000000000000000000000000000000000000000	000000000000
int main() {	
A objA; BobjB; C objC;	
COUTE CODT A.prid CC ODT A.proA	
ecoby A. pub Accends;	
· contex 6678. pri Acc 6678. proA.	The state of the s
ccopy B. pubA ccandl.	The same
cout cc 667 B. pris cc 667 8.pro8	Mary Co.
ecopy BibapBesonds;	NE _{II}
early except c. bright except c. brogh	
ecopy c. public endly	-
contropic buigaropic bros	
2005 C. pub B ccendl;	
conter obje. prid ecobje. pro	
ecobje. pube ecendl;	
	3 4
A CONTRACTOR OF THE CONTRACTOR	P9: 48





QUESTION NO. 04	
POLYMORPHISM" Early & Late Binding Virtual Functions.	
public:	
void show () [[normal	
wid show () { wut ex "perva n"; }	
2.	
class Derve : public Base 1	1 overtived
class Derva; public Base /	1 exertived
class Derva ; public Base 1	1 over 1 rec)

	Base
	The state of the s
A.de Call	OUTPUT
return 0;	No. of the last of
execute show()	
ptr->show();	5.00 - DAY
put address of dva in pointer	4 4
ptr = 2dv2;	
execute show ()	1/57
ptr->show();	
put address of dua in pointer	
ptv= fdv1;	
Prt-> show ();	
.ptr= 2b;	
I pointed to base class	- The state of
Base * pty;	100
object of derived class 2	
Dervadua;	
lobject of derived class 1:	
Dervidus;	
Base b;	
that main() }	



return 0;	TUATUO
0.	Base
	Base
	Base
N 65.77	
ate Binding	
rtual Member	Functions
The state of the s	Va y Care
n	
ass Base // base cla	3.22
Apric: >{keyword}	
virtual void show()	11 virtual function.
coutec "Base In	·· · · · · · · · · · · · · · · · · · ·
	, 1.
lass Derva : public	Base Halerived class a
public:	1100.11
void show () }	
	Sent.
)
	3.
	3.
cout ce " Derv 1 /n";	Base Maerived class a
cout ce " Derv 1 /n";	BUNDLEY BELY /
cout ce " Derv 1 /n";	BUSINESS BELL /
cout ce " Derv 1 /n ";	Base Merived class 2

int main() {	
Base b;	
·Dervaava;	
llobject of derived class 1	
Dervadva;	
llobject of derived class a	Spirite and a
Base *ptr;	A STATE OF THE STA
11 pointer of base class.	To 10 00 315
ptv = & b;	ETAL ALL WEE
pr+>show();	
pty= &dv1:	
1/ put address of dvs in pointer	
ptr-> show();	
Mexecute show ()	SHE THE
ptr= fava;	1 1 1 1 1 1 1
11 put address of dva in pointer	A MAN
ptv-)show();	THINA
flexecute show ()	
return 0;	
}	
And The Land	TUPTUO
A 7 1	Base
	Derva
	Derva

ate Binding:		
Ng	virtual	Base
4 Derv 2	show()	
ptr->show()		
	show()	Devv 1
PTV	ALL NOTES	
& Devva	show()	Derv 2
pty->show()		
nt main () {		
Base b;		1
Dervadua; 11 object of	derived exass 1	
Dervadua; Hobject		2
Base *ptr; (1 pointer		1
Ptv= & b;	Court of the Court	-/
prt->show();	SHERED STORY	All
ptv = & av1; 11 put e	uddress of dva i	pointe
ptr-> show(); llexec		
	ldress of dva ,	's point

2 '	709700
	Base
	Derv 1
	Derv 2
	The state of the s
stract classes and	
Pure Virtual Fund	rions
Pore Willows Volls	BIRTHARD TORK
class Base // base class	
THAT WHITE III DOING THE TANKS	
· · · · · · · · · · · · · · · · · · ·	
Public:	0; 11 pure virtual
public: virtual void show ()=	0; 11 pure virtual
Public: Virtual void show () =	Jan 1991 741
Public: Virtual void show () = 3; elass Derv1: public Base	Jan 1991 741
Virtual void show () = 3; elass Dervi: public Base 11.	Jan 1991 741
Virtual void show () = 3; elass Derv1: public Base public: Void show ()	derived class a
Public: Virtual void show () = 3; elass Derv1: public Base public:	derived class a
Public: Virtual void show () = 3; elass Dervi: public Base public: Void show () coute "Dervi In"; ?:	derived class a
Public: Virtual void show () = 3; elass Derv1: public Base public: Void show ()	derived class a
Virtual void show () = 3; 2lass Dervi. public Base 2 public: Void show () 2 cout Le "Dervi In"; 3.	derived class a
Virtual void show () = 3; elass Dervi: public Bose // g public: Void show () & cout ex "Dervi In"; g; class Dervi : public Bose	derived class a

int main () {	
Base bad;	
I cant' make object from	
abstract class	
Base * arr[a];	
array of pointers to base class	Select Land
Dervadva;	Marie Total
lobject of derived class 2.	- Take
Dervadua;	PDV //8//4/
object of derived class a.	
- arr[0]=&dv1	
put address of dud in array	
arr[1]= &dva	23
put address of duz in array	Direction of
arr[0]-)show();	
execute show() in both obj	
arr[1]-> show();	The state of
return 0;	3.
}	
	TUGTUO
	Derv 1
dinyl as	Derv 3
the state of the s	Making Name