Explain the 1.			
Explain the concept of pure virtual functions, obstruct base classes and interfaces. Discuss the role of abstract			
classes and interfaces. Discuss the role of abstract			
classes and interfaces. Discuss the role of abstract  Answer:  Answer:			
The state of the s			
Pure Vistual Function:-			
pine vixtual function is a contine that has no body			
and is assigned to the o value in the base class. It			
Hostract Bara Class			
Close is a class that has an ar more but			
- virtual functions. It cannot be instantiated, but it			
virtual functions. It cannot be instantiated, but it can be used for polymorphism.			
Interface class:-  It is a considered of abstract have class that			
has only one vistual function which is pure and			
other data members. It defines a common beha			
which can be implemented by different classes.			
Role of Abstract class:-			
Role of Abstract class:-  The role of Abstract class in clesioning class inheritance is to provide a common interface as			
Inheritance is to provide a common interface or			
behaviour for the derived classes. It also enforce			
the derived classes to overide the pure virtual for			
Hence abstract classes behaves as a contract be			
the base class and the derived class.			
CYIOTH ADAC			
CUST 2023			

Explain the concept of association aggregation and composition is object-axiented programming. Provide an world context type of relationship in a real Answer:  Association:  Association:  Association:  Association is a solution between two ox more classes that are atherwise unrelated According to this context one class can use another class without depending on it. It can be one-to-one, one-to-many, many to-one it is a weak relationship.  Example:  A doctor and patient are related and both can exist independently. A doctor can have many partients and a patient can visit multiple electors.  Acoreofation:  It is a special type of relationship. It is represented by a "hous-a" or "a-part-of" relationship. It means one class can has another class as member variable, but they have their own implementations.  Example:  An employee and a department of a university are aggregated with eachother. A department can have many employees but employee does not belong to a single clepartment. Both can exist separately.	Part #
Association:  Association is a selation between two or more classes  that are atherwise unrelated. According to this context  one class can use another class without depending an  it. It can be one-to-one, one-to-many, many-to- many, or many-to-one. It is a weak relationship-  Example:  A doctor and patient are related and both can  exist independently. A doctor can have many patients and a patient can visit multiple clostors.  Association:  It is a special type of relationship. It is represented  by a "has-a" or a-part-of" relationship. It means  one class can has another class as member variable,  but they have their own implementations.  Example:	Explain The
Association:  Association is a selation between two or more classes  that are atherwise unrelated. According to this context  one class can use another class without depending an  it. It can be one-to-one, one-to-many, many-to- many, or many-to-one. It is a weak relationship-  Example:  A doctor and patient are related and both can  exist independently. A doctor can have many patients and a patient can visit multiple clostors.  Association:  It is a special type of relationship. It is represented  by a "has-a" or a-part-of" relationship. It means  one class can has another class as member variable,  but they have their own implementations.  Example:	composition concept of amount
Association:  Association is a selation between two or more classes  that are atherwise unrelated. According to this context  one class can use another class without depending an  it. It can be one-to-one, one-to-many, many-to- many, or many-to-one. It is a weak relationship  Example:  A doctor and patient are related and both can  exist independently. A doctor can have many patients and a patient can visit multiple clottors.  Association:  It is a special type of relationship. It is represented  by a "has-a" or a-part-of" relationship. It means  one class can has another class as member variable,  but they have their own implementations.  Example:	example for object-organism aggregation are
Association:  Association is a selation between two or more classes  that are atherwise unrelated. According to this context  one class can use another class without depending an  it. It can be one-to-one, one-to-many, many-to- many, or many-to-one. It is a weak relationship-  Example:  A doctor and patient are related and both can  exist independently. A doctor can have many patients and a patient can visit multiple clostors.  Association:  It is a special type of relationship. It is represented  by a "has-a" or a-part-of" relationship. It means  one class can has another class as member variable,  but they have their own implementations.  Example:	world each type of real programming. From
Association:  Association is a selation between two or more classes  that are atherwise unrelated. According to this context  one class can use another class without depending an  it. It can be one-to-one, one-to-many, many-to- many, or many-to-one. It is a weak relationship-  Example:  A doctor and patient are related and both can  exist independently. A doctor can have many patients and a patient can visit multiple clostors.  Association:  It is a special type of relationship. It is represented  by a "has-a" or a-part-of" relationship. It means  one class can has another class as member variable,  but they have their own implementations.  Example:	Answer of relationship in a reco
that are otherwise unrelated. According to this context one class can use another class without depending an it. It can be one-to-one, one-to-many, many-to-many, or many-to-one. It is a weak relationship.  Example:  A cloctor and patient are related and both can exist independently. A doctor can have many patients and a patient can visit multiple cloctore.  Accordant type of relationship. It is represented by a "bas-a" or "a-part-of" relationship. It means one class can has another class as member variable, but they have their own implementations.  Example:	: 1 3000
that are otherwise unrelated. According to this context one class can use another class without depending an it. It can be one-to-one, one-to-many, many-to-many, or many-to-one. It is a weak relationship.  Example:  A cloctor and patient are related and both can exist independently. A doctor can have many patients and a patient can visit multiple cloctore.  Accordant type of relationship. It is represented by a "bas-a" or "a-part-of" relationship. It means one class can has another class as member variable, but they have their own implementations.  Example:	$\Delta_{-}$
that are otherwise unrelated. According to this context one class can use another class without depending an it. It can be one-to-one, one-to-many, many-to-many, or many-to-one. It is a weak relationship.  Example:  A cloctor and patient are related and both can exist independently. A doctor can have many patients and a patient can visit multiple cloctore.  Accordant type of relationship. It is represented by a "bas-a" or "a-part-of" relationship. It means one class can has another class as member variable, but they have their own implementations.  Example:	A sociation:
it. It can be one-to-one, one-to-many, many-to- many, or many-to-one. It is a weak relationship-  Example:  A clostor and patient are related and both can exist independently. A doctor can have many portients and a patient can visit multiple clostore.  A cooresation:  It is a special type of relationship. It is represented by a "bas-a" or "a-part-of" relationship. It means one class can bas another class as member variable, but they have their own implementations.  Example:	-dissociation is
it. It can be one-to-one, one-to-many, many-to- many, or many-to-one. It is a weak relationship-  Example:  A clostor and patient are related and both can exist independently. A doctor can have many portients and a patient can visit multiple clostors.  A cooresation:  It is a special type of relationship. It is represented by a "bas-a" or "a-part-of" relationship. It means one class can bas another class as member variable, but they have their own implementations.  Example:	that are otherwise the or more classes
it. It can be one-to-one, one-to-many, many-to- many, or many-to-one. It is a weak relationship-  Example:  A cloctor and patient are related and both can exist independently. A cloctor can have many postients and a patient can visit multiple cloctors.  A coordor and patient can visit multiple cloctors.  A coordor and patient can visit multiple cloctors.  A coordor and patient can visit multiple cloctors.  It is a special type of relationship. It is represented by a "how-a" or "a-part-of" relationship. It means one class can has another class as member variable, but they have their own implementations.  Example:	one class can use
many, or many-to-one. It is a weak relationship.  Example:  A cloctor and patient are related and both can exist independently. A doctor can have many portients and a patient can visit multiple cloctors.  A coreo ation:  It is a special type of relationship. It is represented by a "bas-a" or "a-part-of" relationship. It means one class can has another class as member variable, but they have their own implementations  Example:	it. It can be another class without depending on
A cloctor and patient are related and both can exist independently. A cloctor can have many postients and a patient can visit multiple cloctors.  Asoresation:  It is a special type of relationship It is represented by a "how-a" or a part-of" relationship. It means one class can has another class as member variable, but they have their own implementations.  Example:	
A cloctor and patient are related and both can exist independently. A cloctor can have many portients and a patient can visit multiple cloctors  Agoregation.  It is a special type of relationship It is represented by a "bous-a" or a - part-of" relationship. It means one class can has another class as member variable, but they have their own implementations  Example:	Example. To - one. It is a weak relationship
Agoreoation.  It is a special type of relationship. It is represented  by a "has-a" or "a-part-of" relationship. It means  one class can has another class as member variable,  but they have their own implementations  Example:	
Agoreoation.  It is a special type of relationship. It is represented  by a "has-a" or "a-part-of" relationship. It means  one class can has another class as member variable,  but they have their own implementations  Example:	clocity and patient are related and both can
Agoreoation.  It is a special type of relationship It is represented  by a "how-a" or "a-part-of" relationship. It means  one class can has another class as member variable,  but they howe their own implementations  Example:	exist independently. A doctor can have many
Agoreoation.  It is a special type of relationship It is represented  by a "how-a" or "a-part-of" relationship. It means  one class can has another class as member variable,  but they howe their own implementations  Example:	portients and a patient can visit multiple dectors
by a "how-a" or "a-part-of" relationship. It is represented  one class can has another class as member variable,  but they have their own implementations.  Example:	
by a "how-a" or "a-part-of" relationship. It is represented  one class can has another class as member variable,  but they have their own implementations.  Example:	Agoreoation:
one class can has another class as member variable, but they have their own implementations.  Example:	It is a special type of relatively The
one class can has another class as member variable, but they have their own implementations.  Example:	by a represented
but they have their own implementations.  Example:	C = Part - ot = relationship = 11 means
Examples	The class can bas another class or
An employee and a department of a university are aggregated with eachother. A department can have many employees but employee does not belong to a	Out they have their own implementations
An employee and a department of a university are aggregated with eachother. A department can have many employees but employee does not belong to a	Example
aggregated with eachother. A department can have	An employee and a department of a main
many employees but employee does not belong to a	dependent of a University are
many employees but employee does not belong to a	aggregated with eachother. A department can have
liook danada da Pott	many employees but employee does not below to a
	and a second as
Single department. Both can exist separately.	single department. Both can exist separately.
Composition:	Composition:-
The a type of annualist the Parallel Harris	The a type of annualist the Parallel
CUST 2023	Composition:-  It is a type of amociation Pri which there is a CUST 2023

	Maller II II the Market Con-	
whole part "		
class can house " ourse a" one		
Llass can have other class as member variable and  Example:  A car and all		
Example Share some Die wriable about directional		
A con I is unique		
eachother. Both cannot exist without are composed with		
eachother. Both cannot exist without one another A cax  an engine and the engine had a that		
has an engine and the engine helongs to only that		
	Terigs 18 onig	
Question # 02-		
Provide		
Answers d'agram for given scenario en papex		
	3 scenario on pape.	
Class D. K		
Class Diagram:		
Product	11	
int Product ID; String product Name;	User	
olouble price;	int userID; Shopping cart (art;	
Product pinte;	// (/ ·····	
Product (intide string name, double p)	1770 12-1-12	
	void displan ()	
string get Name()	void display ()  Cart = display Proclucts()  Void create ()	
double get Price()		
() I will )	void add to cart (Product P)	
0*		
Shoppino Cart 12		
Product Products [100]		
ent size;		
Shopping Cart() }	?	
Shopping Cart (') }  Void add Product (Product * P)		
Void display Products ( ) double yer Total cost ( )		
CYVOM 2022		
CUST 2023		