What is OOPS?	Object Oriented programming and structures. Object-oriented programming (OOP) is a computer programming model that organizes software design around data, or objects, rather than functions and logic.
What are major pillar of OOPS?	1. Encapsulation 2. Abstraction 3. Inheritance 4. Polymorphism
What is Encapsulation?	Wrapping up data and functions in a single unit.
How do we encapsulate ?	By creating User defined types.
How many UDT's are there in .Net ?	There are 5 UDT's in .Net 1. Class 2. Struct 3. Interface 4. Delegate 5. Enum
What is Abstraction?	Hiding the complexity of your implementaiton(UDT) from end users is called abstraction.
How do we abstract?	By using access modifiers
what are the various access modifiers in .Net	Public - Public members of a class are accessible via methods and functions on class. Private - Private members of a class are only accessible via member functions and never via object. Protected - Protected members are just like private members i.e. available via member functions and never via object with exception that they are accesible via member functions of derived classes as well. Not via derived class objects Internal - Internal is just like public within assembly(project) but outside assembly they behave like private data members Protected Internal - Behaves likes public within assembly and outside assembly they behave as protected
What is Inheritance ?	Inheritance is extending the base class definition by adding data and functions. We can also hide or override methods of base classes
What is polymorphism?	One thing many forms. Its of 2 types 1. Compile time polymorphism - All polymorphic function calls are resolved at compile time example is method hiding and method overloading. Overloading is of 2 types i.e. operator overloading and method overloading 2. Runtime Polymorphism - All polymorphinc function calls are resolved at runtime example is overriding.
What is method overloading ?	Redefining a method with same name but different signature. We can overload a method in same class as well as child class. In case of overloading we follow the rule THY OBJECT THY METHOD means call the function defined in class of which we are creating instance. Overloading depends on 1. Number of parameters 2. Order of parameters 3. Type of parameters overloading does not depends on 1. Method return type. You cannot have a method with same name but different return types in a class
What is method overriding ?	Redefining a method with same name and signature but in child class. Overriding is an inheritance concept unlike overloading. In case of overriding we use rule 1. Casting + Overriding -> call child With overriding we can force base class to call child class methods. We can only override a method defined as virtual/abstract in base class
What is method hiding?	Hiding is an inheritance concept just like overriding. We redefine a method with same name and signature in child class but without using override keyword. We can use new keyword to tell compiler that its not by accident that we have redefined a method
What is virtual keyword ?	Virtual keyword is used to define methods with a default definitions in base class. Child can use the default virtual implementation if it satisfies their requirement or hide/overide them . If child overrides the implementation then child object and base objects created via casting will call child class implementation. Virtual is CAN override
What is abstract keyword ?	Abstract methods are example of template programming. Base class provide abstract method but without body leaving the implementation details to be provided by child class. Since base class don't have body of these methods they must be defined in child classes for them to termed as concrete. If child class choose not to override they need to be created as abstract class as well. Abstract is MUST override
What is and interface	Interface is a contract between 2 parties. All methods of interface must be implemented by child class. Counter question - is it mandatory to implement all methods? Answer - We can redeclare methods as abstract class if we don't want to implement in child so that responsibility of impementation gets transferred to next child class in hierarchy t
What is abstract class?	A class which cannot be instantiated. Sole purpose of creating abstract class is inheritance . Its not mandatory for an abstract class to have abstract methods but if there is an abstract method in a class then it must be abstract
Where will you use interface ?	Interface are used in 1. Dependency injection 2. Defining contract for child classes to implement 3. Sharing contracts with end user in case of APIs 4. With Test Driven Development its easy to mock interfaces
When will you use abstract classes ?	Abstract classes are used 1. When inheriting from framework classes - As a thumb rule of architecture we should never inherit from framework classes directly so we create a level of indirection by creating an abstract base class that inherits from framework class and then let other classes inherit from base abstract class. This helps in code injections later on
What are the various types based on storage ?	Value types and reference types. Value types are value at address. They are stored on stack. example int, decimal. There is only a single memory lookup in case of value types References types are address at address. They are stored on Heap example Class, Interface ,Delegate are reference types. There are 2 memory look ups in case of reference types
What is an object ?	Object is blue print for a class. Storage of object depends on data members defined in class. Its a runtime entity. Class is a compile time entity
What are static members how are they different from instance members?	Every class will have only one copy of static data members no matter how many objects you create . Objects will have their own copy of all other data members of class that are not static. Static members are always referred by class name e.g. Person.Id where person is name of class and Id is a public static data member
What is a constructor?	Constructor are functions having 1 Same name as that of class 2. No Return type . Not even void 3. Called implicitly by compiler at time of object creation.
Can we overload a constructor ?	Yes

How can we call an overloaded constructor of same class	Using this keyword (Me in VB.Net)
	class Person{
	public Person() : this(1){
	}
	public Person(int i){
	}
	}

How can we call constructor of base class?	Using base keyword (Super in VB.Net)
	class Human{
	public Human(int i){
	}
	}
	class Person : Human{
	public Person() : base(1){
	}
	
	}

What is a destructor ?	Function with same name as that of class prefixed with ~(tilde) . They are called when object goes out of scope to clean up memory
	public class Car
	{
	public Car()
	{
	Console.WriteLine("Car constructor");
	}
	
	~Car()
	{
	Console.WriteLine("Car DESTRUCTOR");
	}
	}

What are managed resources ?	.Net has concept of automatic memory management. Managed means managed by CLR (common language runtime) .Apart from pointers, direct memory access and I/O operations rest all operations are managed
What is CLR?	Common language runtime is runtime execution environment of .net framework.
What are the various components of CLR?	1.Class Loader: It is used to load all the classes at runtime. 2.MSIL to native Compiler: It is a JIT (Just In Time) compiler it will convert MSIL code to native code. 3.Code manager: It manages the cade during runtime. Garbage Collector: ... Security Engine: ... Type checker: ... Thread support: ... Exception manager: COM Marshaler
How do we clean up memory of unmanaged resources?	All unmanged resources provided by .net framrwork implement IDisposable interface which provides a method called Dispose . When we want to clean up memory of these resources we explicitly call dispose methods on these objects
What is 2 step compilation in .Net?	In .Net there is 2 step compilation Step 1 : We write code in C#/VB.net and build it. After build source code gets converted to MSIL (Source code gets converted to assmeblies and MSIL is part of assembly) Step 2: At runtime the assembly code gets executed and the MSIL code in it gets conveted to Native code(platform specific code) by JIT (Just in time) compiler
What is a namespace ?	Namespace is a logical grouping of classes which is used to avoid name collisions if a type with same name is defined in multiple assemblies
What is an assembly?	An exe or dll is called an assembly in .Net
What are the various parts of assembly?	Assembly has 1. Assembly metadata or manifest - Also called header of assembly and contain information about - List of referenced assemblies 2. Type Metadata - Information about types contained in assembly 3. MSIL 4. Resources - - List of referenced resources like images/text files etc
What is a module ?	MSIL and type metadata is often referred as a module
Explain garbage collection.	.Net has automatic memory management. When an object goes out of scope its available for garbage collection. CLR has a component called garbage collector (which is a thread) . When garbage collector runs it frees up memory of these unused resources There are steps in garbage collection process 1. Mark - Marks the objects that are out of scope for deletion 2. Collect - Frees up memory of unused objects. This created holes in memory since object allocation is not contiguous. 3. Compact - Compaction changes memory location of leftover object in a way that all free memory becomes contiguos. 4. Reference update - updates the stack addresses with newly assigned heap location
Can we restrict object creation in .Net?	Yes, one can create an Object Creation by using the following methods- Abstract Class, Static Class, Private or Protected Constructor.
What is sealed class?	Sealed classes are classes that cannot be inherited example Enums

How do we define constants in .Net	By using const and readonly keywords.
	Const are compile time constants .Declaration and definition of const are in same line
	readonly are runtime constants. We first define variable as readonly and then can assign value at runtime inside the constructor.
	public class Test{
	const string hello = "HELLO";
	readonly string test;
	
	public Test(){
	test = "TEST";
	}
	}

What are various exception handling constructs in .Net	Try , Catch and Finally
Can we handle multiple type of exceptions ?	Yes with multiple catch blocks
How many catch blocks can be executed in case of exception?	Only one catch block can be executed
Is it mandatory to have catch block?	No we can have try-finally block as well
What happens if we don't have a matching catch block?	Exception will be unhandled
How to gurantee that a code block always executes whethere the	Write code in finally block
What should be order of catch blocks?	More specific (child class) to more generic (base classes)
Can a constructor be private ? If yes give some use cases	Yes a constructor can be private It is used to stop object creation of a class. It is used in Singleton class. It is used to stop a class to be inherited.

What is Singleton design pattern?	Designing a class in a way that only one instance of class can be created.we require
	1. Private constructor
	2. Private static instance member of class
	3. static method that instantiates static instance member and return
	public class Singleton{
	private Singleton() { }
	
	static Singleton _singletonInstance;
	public Singleton GetInstance() {
	if (_singletonInstance == null)
	_singletonInstance = new Singleton();
	return _singletonInstance;
	}
	}

What is factory design pattern?	Factory is creational design pattern which creates a single instance from a family of classes. Factory uses inheritance
What is DRY?	DRY is a code refactoring principle which states DONT REPEAT YOURSELF

What Is SOLID	SOLID is a refactoring principle S - Single responsibility Principle O - Open/Closed principle L- Liskov substitution principle I - Interface segregation principle D - Dependency inversion principle
What is SRP	Every class should only have one responsibility and thus only one reason to change. If a class has multiple responsibilities then refactor into multiple classes
What is SOC	Seperation of concerns is a refactoring principle which states that every application concern should be in a separate layer(project)
What are the various concerns for an application?	Presentation , Data , Logging and Auditing, Security, Business logic etc
What is Open and Closed printciple	Base classes should be open to extension and closed to modification. In case same functionality is required by mutliple child classes then we should create a new class inheriting from base class and introduce the common functionality in that class. Then all child clases can inherit from newly created base class
What is liskov substitution principle?	Child classes should not alter the behavior of base classes. If base class has a method say Add which has addition behavior than child should have different logic to add but cannot change intent to subtract
What is interface segregation principle?	Avoid fat interface. Clients should not be forced to depend upon interfaces that they do not use.
What is dependency inversion principle?	Whenever there is a depdency relationship between 2 classes it should be on abstract class and interfaces
What is code for abstraction?	Creating base abstract classes and interfaces
What is the use of static constructor ?	Called only once during application lifetime and use to assign static data members
Can we return multiple values from a method?	Yes, it is it possible that a Method can return multiple values at a time in C# by using the following: KeyValue pair Ref or Out parameters Struct or Class Tuple
What are design patterns?	Solution to common problems in software industry. They are of 3 types Creational Patterns: It mainly deals with creation of Objects and Classes. Structural Patterns: It deals with Class and Object Composition. Behavioral Patterns: It deals with Class and Object communication. That means they are concerned with the communication between class and objects.