

Q. Code Compilation in C#.

- 1) C# Compiler compile source code into managed code.
- 2) Putting the newly created code in into assembly.
- 3) The CLR is loaded, carrying out the assembly.

Q. Difference b/w Class and Struct.

- | <u>Class</u> | <u>Struct</u> |
|---|--|
| - Support inheritance | - Does not support inheritance. |
| - Class can pass by reference | - Struct is pass by copy (value type) |
| - Member are private by default | - Member are public by default |
| - Suitable for large complex object | - Suitable for small object |
| - Can use waste collector for memory management | - Cannot use garbage collector and hence no memory management. |

Q. What is Virtual method and Abstract method.

Virtual Method \Rightarrow This can be implement under the abstract or non-abstract method class. This can access by the base as well as inherit class.

if you inherit the class where this method is implement then you will have to use override keyword to use this method.

Abstract Method \Rightarrow Only can declare under abstract class. This method will get body once you inherit this abstract method under ~~non abstract~~ class. we use it when we have require same method with diff. logic.

Q1) Type of classes in C#?

⇒ ① Partial class ⇒ Partial class can be shared or divided in multiple files. The keyword partial denote it.

Abstract is restricted class that we can't use.
② Abstract class ⇒ The abstract class one whose object cannot be instantiated. The only way to get the class is inherited. It must have include at least one method.
Abstract keyword denotes it. Can't use to create object.

③ Sealed class ⇒ Sealed class cannot be inherited. To access the sealed class we must create the class's object. The keyword sealed used for it.

④ Static class ⇒ It cannot inheritance. member are static. Static variables. It can only access static items.

Q2) Equality operator(==) and Equals() method?

⇒ Equality operator(==) Compares reference identity.
use to compare values of two variable.
It check if both the variables are equal in terms of value not reference.

Equals() method ⇒ Object.Equals() are use to compare content of two object.
It check if the both object are equal in terms of content.

Q. What is CLR?

= Common language runtime. It compile the translated code to machine language with the help of JIT Compiler. It provide memory allocation, garbage collection, Exception handling and other services.

Q. What is Managed and Unmanaged code?

Managed Code \Rightarrow Managed Code are those are developed by the .Net framework is known as managed. directly Executed by CLR with help of managed code Execution.
Any language written in .Net framework is managed code.

Unmanaged Code \Rightarrow Developed outside of .Net are unmanaged. which are not go through CLR.

Q. What is Interface?

Interface is a class blueprint. It is like abstract class. all the method that will be declared in Interface are abstract. It cannot use to create object. They don't have body (method have body) it is provided by the Implement class.

Q. Thread & Multithread?

Thread is collection of multiple instructions that can be executed to allow our program to perform concurrent processing.

C# has only one thread by default. We can be created to run the code in parallel with the original one.

multithread - differ thread handle differe process called multithread.

Q. Property of thread - is Alive. When thread is active then is actual

Name - Return thread name
Priority -

Q. Diff Stage of Thread.

Unstarted - created

Running - began executed

Wait/Sleep/Join - A thread call ~~wait/sleep~~ or the wait and Join

Suspend - has to be halted.

Aborted - Died but not been changed to the state stop.

Stopped - has can to halt.

(47)

Q. Deadlock :- when process can't be completed its execution because two or more threads are waiting for completion of one another. Common in multithreading.

Q) What is CLR?

= Common language runtime. It compile the translated code to machine language with the help of JIT Compiler. It provide memory allocation, garbage collection, Exception handling and other services.

Q) What is Managed and Unmanaged Code?

Managed Code \Rightarrow Managed Code are those are developed by the .Net framework is known as managed. directly Executed by CLR with help of managed code Executing.
Any language written in .Net framework is managed Code.

Unmanaged Code \Rightarrow Developed outside of .Net are unmanaged which are not go through CLR.

Q) What is Interface?

Interface is a class blueprint. It is like abstract class. all the method that will be declared in Interface are abstract. It cannot use to create object. They don't have body (method have body) it is provided by the Implement class.

Q Multicast Delegate?

That hold the reference to multiple function

Q Synch Asynch operation

Synch: Tech Thread safe code. which only one thread can access any given time.

Asynch:- The method call return immediately so the other programme can perform other operation while the called method complete.

Q Reflection in C#?

It is ability to access the assembly's metadata during runtime. A program reflects itself and uses metadata to inform or change its behavior.

Q Generic Classes - that are create object that do not have datatype.
datatype can be assigned at runtime.

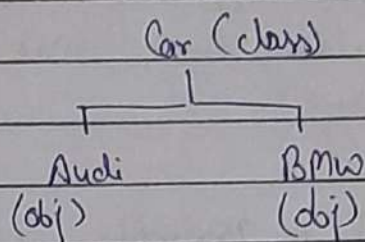
Q Get & Set

They refer to accessor. use to read or write the value of private field.
gain access to that private

⑧ What is object & classes?

Object: object is Instance of class. Basically a block of memory that has been allocated according to the blueprint.

class: A class is a data structure that combines the variable and function in the single unit.



⑧ Oop's fundamentals?

① Encapsulation: When the unnecessary information is hidden from viewing outside. Only the necessary info can be viewed is called Encapsulation.

② Abstraction: Abstraction is a process which identifies and eliminating irrelevant details from the object critical behaviours and data.

③ Inheritance: It is ability to new classes and gaining data from the parent classes. Extend the behaviour from parent class.

④ Polymorphism: "one name many forms". Same name of method but diff. implementation.

Q What is Delegate?

Delegate is a type that represents reference to method with a particular parameter list & return type.
Include System.delegate namespace.

Q How to use Delegate with Event?

Delegate authority is used to initiate and manage Event. A delegate must always be declared first follow by Events.

Q What are Events?

Events are event action that cause the application to receive notification to which it must respond.

Q Type of delegate

Single - only call one method

Multiple Delegate :- Can call multiple method known as multicast delegate. (+) (-) operators used to subscribe or unsubscribe.

Generic Delegate :- Does not required instance has 3 types.

Action :- we can use action keyword to replace the definition of delegate and Events.

Func :- That can define method that takes argument result

Dot Net

Characteristics of .NET Core:

1. Cross platform:- Can be support Linux, Windows, MacOS any of OS.
2. Friendly:- It can be used in different web framework like Angular, react and JavaScript.

Q. How does the .net framework work?

→ .net framework based applications that are supportive in C#, F# or Visual basic are compiled to CIL (Common Intermediate language).

⇒ Code will be stored in assembly file with extension like .exe.

⇒ Common language runtime (CLR) take and convert (CIL) Common Intermediate language into machine code.

⇒ Now the machine code can be even execute on specific arch. of comp.

(Q.) Difference b/w C & C#.

C

C#

- | | |
|--|---|
| 1. The C programming allows procedural | 1. C# allows object oriented program |
| 2. Garbage Collector are not supported | 2. Garbage Collector handle by C# |
| 3. The C programming language is cross platform. | 3. C# needs .NET framework. |
| 4. It support pointers. | 4. in C+ pointers are unsafe mode. |
| 5. low level abstraction. | 5. high level abstraction |
| 6. use in Industrial Commercial & Engineering program. | 6. use for software develop & web networking. |

Q. Property of Array.

- Length - It return the total no of element in array.
- Size - Indicates whether the array's length is fixed or not.
- Read - Indicates whether the array is read only.

Q. Array class - Array can be created, manipulated, stored for, and stored using the array class. array class does not belong to ~~an~~ System.

Q. String - String is grouping of char objects. String variable can also be declared in C#.

Q. Escape Sequence - A backslash indicates the escape sequence (\).
It is treated if it has single character.

Q. What are regular Expression?

regular Expression is a template that can match a set of input operators, constructs and character literals can also use in the pattern.

Q. Parsing - Parsing is the procedure of converting a String into another data type.

Q. Difference b/w Continue & Break Statement.

Break Statement - Break is used to terminate the loop. It causes the program's control to exit the loop.

Continue Statement - Continue Statement instructs the program's control to exit the current iteration. However, it does not affect the loop.

Q. final, finally, finalize?

final - Indicates the variable, method, or class is unchangeable.

finally - finally is like try-catch... it is used for exception handling. Most of the time, this block is used to close resources such as database connections, I/O resources, etc.

finalize - Call before removing object from memory.

Q. Array -> An array is a data structure that stores multiple variables of the same type. A set of variables stored in a single memory location.

Q. Jagged Array - It is an array with array elements. It is also known as an array of arrays. Can be single or multiple dimensions.

Q/ What is Destructor in C#

They are methods inside the class used to destroy instance of the class when they are no longer needed.

garbage collector has control on it in .NET framework.

Q. What is Abstract class?

- This can be achieved by Abstract keywords. It will always be a base class.
- Should always be inherited. Can't create instance of this class.
- If we ^{don't} want any program to create object of class, we make such class abstract.
- There will be no implementation of abstract class, only child class can implement it.

Q/ Boxing and Unboxing Code?

Boxing Code
Process

Boxing is a ~~method~~ where converting value type to reference type.

```
int Val1 = 50; // Boxing
```

```
object boxingVal = Val1
```

Unboxing Code

Unboxing is the explicit converting of the same reference type back to the value type.

```
// Unboxing
```

```
int unboxing = (int) boxingVal;
```


Q// Exceptional handling in C#.

It can be handle by 4 key words: Try, Catch, finally, Throw.

1. Try: Contain block of code to check for Exception
2. Catch: Catches the Exception by Exception handler
3. finally: It's block of code written to run regardless of Exception is caught
4. Throw: When problem occurs, it throws an Exception.

Q// What are Input/Output classes?

The System.IO namespace in C# Contains use to perform various operation. create, delete, opening, close etc.

Ex:

File: Aids in manipulation of a text file.

Stream writer: This class is used to write character to a stream.

Stream reader: read character.

etc.

Q// Stream Reader/ Stream Writer class.

The namespace System.IO contain it use to read and write character.

Member of Stream reader - close(), Read(), Readline().

Member of Stream writer - close(), write(), Writeline().

Q. Namespaces in C#

It is use to organize the classes in .Net. It help to control the scope of method and classes in larger .Net programming projects.

In easy way it is use to keep one set of name file away to other sets of names.

Q. What is using Statement?

- using keyword we do or indicates that the program uses in specific namespace.

- Ex:- System. The System is a namespace in the context.

Q. Abstraction?

⇒ It use to display only essential features and hide unnecessary information.

Internal Info can be hidden by using parameter as private and using private keyword.

Q. Polymorphism = Define as Poly means many, morphism means forms.
Same method but diff Implementation.

Compile-time ⇒ which method will be called decided in Compile time.
overloading. Same name of method but different parameters.
also known as early binding.

Runtime ⇒

overriding ⇒ Parameters or no of parameter will be same but diff implementation. late binding.