1. What is managed code?

code that is run by the clr

2. What is unmanaged code? Examples

code not run by clr. com interop, pinvoke

3. how do you force gc to run?

System.GC.Collect

4. How many generations are there in Garbage collection?

3 gens - Generation 0,1,2.

5. When does GC run?

when the memory is low. - decided by clr

6. what do we write instead of destructor?

Dispose method of IDisposable interface

Advantage of IDisposable - Dispose() method is automatically called after a

```
using (Class1 obj= new Class1())
{
....
}
```

7. What do you get after compilation?

Assembly

8. Format of Assembly is in ____

MSIL/IL/CIL

9. Diff between EXE and DLL

EXE is generated for a console app - has an entry point(static void Main()).

Dll – class library - no entry point

In .net core exe only calls the .dll

10. When is IL converted to native code?

when function is first called.

11. What is this process called?

JIT compilation - done by Jit compiler

12. What does assembly contain?

Code and metadata (MSIL, Assembly Manifest, Type metadata and Resources)

13. What is assembly manifest?

data about the assembly, eg name, version, culture, public key, plus class metadata

15. What are app domains?

Area within a process where the application runs. allows more than one application to run in a single process safely

16. What is reflection?

finding contents of the assembly programmatically. also allows us to call code from the assembly without adding a reference to it. can be used to call private methods also

Classes used are in System.Reflection namespace. Assembly, Type, MethodInfo,PropertyInfo etc

17. How do you see the contents of the assembly with a command line utility?

ILDASM

18. Roslyn command line compiler

to build assembly

dotnet build ProjectName

to run assembly

dotnet AssemblyName

19.

Properties –

set – while assigning a value

get – while reading a value

auto property – used when no validations, compiler generates code for get and set and also generates private variable

public int Prop1{get;set;}

21. What are the .net csharp datatypes, and their corresponding CTS types and the size of each of them?

char, bool - Boolean, byte-Byte, sbyte, short-Int16, ushort-UInt16, int-Int32, uint-UInt32, long-Int64, ulong-UInt64, float-Single, double, decimal (all value types)

string, object (reference types)

?Named parameters

- pass parameter by name insted of position eg obj.Method(a:100, b:200)

? local functions

- function within another function

- allow you to access variables declared in outer code

- static local functions cannot access variables declared in outer code

?why static variable

- single copy for entire class, access with ClassName.VariableName

?static func

- can be called without creating an object of class

- can only access static members directly.

? static constructor?

- called when class is loaded - either when 1st object is created or 1st static member is accessed

- parameterless, implicitly private

? static class

- cannot be instantiated, no inheritance

- can contain only static members

? how to access base class members

- base.member

? access specifiers

private - same class

public - everywhere

protected - same class, derived class

internal - same class, same assembly (same project)

protected internal -same class, derived class, same assembly (same project)

private protected - same class, derived class that is present in same assembly

?diff between hiding and overriding

- any method can be hidden, only VIRTUAL methods can be overridden

- base class declare method - virtual, derived class keyword - override

- for hiding derived class use keyword - new (optional) also works without new keyword but compiler gives warning

- overriding allows late binding

? late binding

- declare reference of base class. allocate memory for either base class or any of its derived or sub derived classes. when method is called, automatically calls correct method

? sealed methods

- prevents overriding in further classes

- sealed must come with override keyword

? abstract class

- cannot be instantiated

- can only be used as base class

- may or may not contain abstract methods

?abstract method

- pure virtual function

- does not have code body, only signature

- must compulsorily be overridden in derived class otherwise derived class also needs to be abstract

?sealed class

-opposite of abstract class

- can be instantiated

- cannot be used as base class

? interfaces

- earlier could only contain method signatures. NOW can also contain method implementation - this is called default implementation of interface methods

- contract - all methods must be implemented by class - except default methods - optional

? what can they contain

- methods, with code or without code

- properties

- static members

- BUT NO INSTANCE MEMBERS

?class implementing interface

```
public void Method1(){...} -- public implementation
void Interface.Method2() {....} -- EXPLICIT implementation

? to call public implementation
Class1 o = new Class1();
o.Method1();

? to call explicit implementation(read methods 2,3,4 from code)
Interface oI = new Class1();
oI.Method2();
((Interface)o).Method2();
(o as Interface).Method2();



? read IComparable and IComparer from Array sorting example
- IComparable - has to written in same class
- IComparer - can be written in different class also. More flexible, allows sorting in multiple ways

? read up code examples on ref and value types

//ref - changes made in func reflect back in calling code

//out - changes made in func reflect back in calling code
    //the initial value is discarded
    //out variables MUST be initialized in the function
//in – readonly, value cant be changed
```

?read up on Nullable types

int? a; a.HasValue a.Value

? read up on Indices and Ranges example

?read up on Tuple and ValueTuple example –

Used to pass multiple values as a single value to a function

Tuple – reference type, max 7 items, no name, 8th item onwards is a nested tuple

The ValueTuple is a lightweight value type object

and has its memory stored on the stack.

ValueTuple is a mutable struct and exposes its elements as fields.

No limits on number of items and can name each item

```
ValueTuple<int, int> vt0 = new ValueTuple<int, int>(10, 20);
(int, int) vt1 = (1, 2);
  var vt2 = (1, 2);
```

22. How do you convert from one type to another?

Convert.To....()

23. Operator overloading function

public static ClassName operator+(ClassName o1, ClassName o2)

```
ClassName o1= new ClassName();
ClassName o2= new ClassName();
o1 = o1 + o2;
```

24. How to declare simple array?

```
int [] arr = new int[5]
```

25. How to declare double dim array?

```
int [,] arr = new int[5,3]
```

25. How to declare jagged array?

```
int [][] arr = new int[3][]
arr[0] = new int[4]
arr[1] = new int[3]
arr[2] = new int[5]
```

26. Methods of the Array Class

To do yourself

27. Size of array

arr.Length property

28. Length of 2nd dimension

```
arr.GetLength(1)
```

29. Number of dimensions

```
arr.Rank
```

How to pass multiple parameters to a func?

use params keyword - all passed parameters are converted to Array

```
int Add(params int[] arr)
```

30. What are indexers

Allow you to access an object like an array/collection

```
Class1 o = new Class1()
o[0]  = 10;

public int this[int subscript]
{
get{}
set{}
}
```

31. What are value types?

stored on stack

Enums and structs

32. Differences between struct and class

1. value type/ ref type
2. inheritance not allowed in structs

33. What is boxing/unboxing?

boxing is store a value type into ref type

unboxing is store a ref type into value type

36. What are delegates?

Delegate is a class that inherits from System.MulticastDelegate class. MultiCastDelegate inherits from Delegate class.

Delegate is used to call a function indirectly (func pointer)

37. How to call more than one func using delegate(multicast)?

```
MyDel obj = (MyDel)Delegate.Combine(new MyDel(f1),new MyDel(f2))
```

OR

```
MyDel obj = f1
obj+=f2
```

then call using obj()

38. How to declare and call an event?

in class...

```
public event DelegateName EventName;
```

to call it...

```
EventName(parameters if any);
```

39. How to add an event handler in class?

```
objCls1.EventName += FuncName
```

40. How to create our own attribute?

Inherit from System.Attribute

41. How to create a class that uses a Generic parameter?

class MyClass <T>

{....}

42. What are the generic constraints?

class MyClass <T> where T : class

{....}

either

class/struct/classname/interfacename/new()

43. What interface do all collection classes implement?

ICollection/ICollection<T>

44. Collections based on a list

IList/IList<T>

45. Collections based on a Dictionary

IDictionary/IDictionary<TKey,TValue>

46. Collection classes and methods

ArrayList, HashTable, SortedList, Queue, Stack, DictionaryEntry

List<T>, SortedList<Tkey, TValue>, Queue<T>, Stack<T>, KeyValuePair<TKey,TValue>

47. All exceptions in .net inherit from ____ class

System.Exception

48. Which exception must be caught first in catch?

Derived classes first, Base classes last

49. Access specifiers in .net

public, private, protected, internal, protected internal, private protected

50. Def access specifier within a class

private

51. Def access specifier within a namespace

internal

52. How to refer to the base class?

base keyword

53. How to refer to the current object?

this keyword

54. Explain following keywords with relation to inheritance

new

virtual

override

sealed (method)

abstract (class)

abstract (method)

sealed (class)

55. How to implement interface

- Implement interface public void Method(){...}

- Explicitly implement interface void Interface.Method(){...}

56.Static class limitations

can only have static members

cannot be instantiated

cannot be used as a base class

57. Static constructors

no parameters

implicitly private

called when the class is loaded into memory (either 1st obj created or static member accessed for 1st time)

58. How to call a func asynchronously using delegates?

objDel.BeginInvoke(callback, object)

59. How to get the return value for a func called asynchronously using delegates?

retval = objDel.EndInvoke(ar)

60. where/when to get the return value for a func called asynchronously using delegates?

in the callback method

61. How to call a method using Thread class

Thread t = new Thread(new ThreadStart(funcname))

t.Start()

62. How to call a method with object parameter using Thread class

Thread t = new Thread(new ParameterizedThreadStart(funcname))

t.Start(data)

63. How to call function using ThreadPool

ThreadPool.QueueUseWorkItem(func)

64. What is the ThreadState property?

- Unstarted, Running, StopRequested, Stopped, WaitSleepJoin, Background, AbortRequested,Aborted, SuspendRequested,Suspended

65. What is the Priority property? (ThreadPriority enum)

- Normal, AboveNormal, Highest, BelowNormal, Lowest

66. What does objthread.Join() do?

performs a waiting call

67. How to call a function using Task library?

Task t = new Task(func)

t.Start()

Task.Run(func) -- does not allow parameters to function

Task.Factory.StartNew(func) -- allows one parameter - object

68 Waiting call with a task?

t.Wait()

69. What are implicit variables?

var x = 10;

datatype based on RHS. Can only be used for local variables.

70 What are anonymous types?

var x = new {a=1, b="ss", c=true};

class with no name, created by the compiler

71 What are extension methods? How can they be called?

static method in static class. 1st parameter is the class for which you write extension method. Write keyword 'this' before 1st parameter.

int i= 100;

i.Display();

ClassName.Display(i);

72. _____ is inbuilt delegate for calling methods with void return value

Action

73. _____ is inbuilt delegate for calling methods with non void return value

Func

74. _____ is inbuilt delegate for calling methods with single parameter and bool return value

Predicate

75. Lambda example

Func<int,int> = x => x/2;

76. What is deferred execution in LINQ

Query executes only when you iterate thru the collection. This is by default.

?Immediate execution

- query runs immediately

- ToList(), ToArray(), ToDictionary() force immediate execution

77. LINQ

To see examples

78 LINQ query can be writen for any collection that implements _____

IEnumerable , IQueryable(which implements IEnumerable)

79 LINQ query returns _____

IEnumerable<T>

How to convert linq to plinq?

.AsParallel()

How to limit max threads on a plinq query?

.WithDegreesOfParallelism(4)

80. What are partial classes?

code in 1 class can be split across multiple classes using partial classes

81. Partial method?

Method is written in 1st class and implemented in other partial class. Compiler removes method if not written in 2nd class

82. How do you get Thread synchronization in csharp?

lock(obj) { .....},

Lock class.

using (lockObject.EnterScope()){},

lock (lockObject){},

lockObject.Enter(); .Exit();

if (lockObject.TryEnter()){}

Monitor.Enter(obj) and Monitor.Exit()

Interlocked.MethodsOfInterlocked

using Wait() or Sleep() or Join() methods (waiting calls)

Using derived classes of WaitHandle (AutoResetEvent, ManualResetEvent, Mutex, Semaphore)

Mutex - across processes - only one can run

Semaphore - across processes - max count can be set

83. when to use async and await?

When using Task. Specially if there is a waiting call. Useful for UI's

Non blocking call

Creates a Task, starts it and gets return value in same line

? rules for await

- can only be in an async func - method must return a task if used in MVC or if method is Main()

- can only call awaitable methods (methods that return a Task)

84. How to access drive, directory or file?

Drive obj

Directory.Static methods or create an object of DirectoryInfo class and use instance methods

File.Static methods or create an object of FileInfo class and use instance methods

85 FileMode enum

Create - always overwrites file

CreateNew - overwrites file, gives exception if file is present

Append - adds to file, creates if not present

Truncate - gives exception if file is not present, deletes content from file after opening, allows to write to file

Open - opens existing file in read mode

OpenOrCreate - opens existing file in read mode, creates if file not present

?read up code examples on

- How to write/read file in text mode - using formatted and unformatted data

- How to write/read file in binary mode

?what is serialization? -Serialization is the process of converting the state of an object into a form that can be persisted or transported. The complement of serialization is deserialization, which converts a stream into an object

- [Serializable] attribute for Binary or SoapSerialization

- ISerializable interface for custom serialization

  -- GetObjectData(SerializationInfo, StreamingContext) – serializing

  -- constructor with same parameters – deserializing

86. How to connect to Sql server db?

Microsoft.Data.SqlClient - nuget

Create SqlConnection object

call cn.Open

87. How to connect to Sql server db asynchronously?

Create SqlConnection object

call cn.OpenAsync()

88. How to Insert/Update/Delete records

Create SqlCommand object

use cmd.ExecuteNonQuery() (returns int - number of records affected)

89. How to read single value

Create SqlCommand object

use cmd.ExecuteScalar() (returns object - first row, first col value)

90. How to use select statement to return records

Option 1 :

Create SqlCommand object.

use cmd.ExecuteReader() to return SqlDataReader object

loop thru records using dr.Read()

Option 2 :

Create SqlCommand object.

Create SqlDataAdapter

Create DataSet/DataTable

use da.Fill() to populate DataSet/DataTable

loop thru records using a foreach for the Rows Collection of the DataTable

91. What is MARS?

in connectionstring use MultipleActiveResultSets=true

SqlDataReader does not place exclusive lock on connection

92. What is DataView used for?

to see records sorted or filtered

```
ds.Tables[0].DefaultView.Sort = "DeptNo";
ds.Tables[0].DefaultView.RowFilter = "DeptNo=10";
```

93 How to populate DataSet/DataTable?

```
da.SelectCommand = cmd
da.Fill(...)
```

94 How to update DataSet/DataTable?

```
da.InsertCommand = cmdInsert
da.UpdateCommand = cmdUpdate
da.DeleteCommand = cmdDelete
da.Update(...)
```

95. RowState property of each DataRow?

Unchanged

Modified

Added

Deleted

Detached

96. DataSet - What does GetChanges AcceptChanges RejectChanges do?

GetChanges - gets all rows with changes (not Unchanged)

AcceptChanges - called after da.Update. Sets rowstate to Unchanged. removes deleted rows, makes current values as original

RejectChanges - undoes all changes made in DataSet/DataTable

98. All controllers with View support inherit from ____ class

Controller

99. All controllers WITHOUT View support inherit from ____ class

ControllerBase

100. return type of Methods in controller

ActionResult : IActionResult

?Read on Derived classes of ActionResult

101. HttpGet/HttpPost/NonAction attribute

called for a Get or a Post. NonAction used to indicate that method is not to be recognized as an action

102. Html Helper Functions in MVC

?@Html.

@Html.TextBox - does not use model binding

@Html.TextBoxFor - uses model binding

103. State management options MVC (Important)- check syntax

ViewBag - controller to view for current request, lost on redirect, wrapper for ViewData

ViewData - controller to view for current request, lost on redirect

TempData - controller to view for current request, retained on redirect

Cookie - across sessions, stored on client machine in text file

Session - VERY VERY IMPORTANT - read up

Cache - single copy for all users with expiry policies

Program.cs

```
//first add session
//builder.Services.AddSession();//default with no options

//with options
builder.Services.AddSession(options =>
 {
 options.IdleTimeout = TimeSpan.FromMinutes(10); //session timeout
options.Cookie.HttpOnly = true; //whether client side script can access cookie. false means yes
```

```
options.Cookie.IsEssential = true; //cookie will be always
created if true
            });
```

Store into session using byte array- not convenient

```
 HttpContext.Session.Set("key1",bytearr);
//byte[] arr = HttpContext.Session.Get("key1");
```

Store into session using string or int- better

```
HttpContext.Session.SetString("key1", "value");
HttpContext.Session.SetInt32("key2", 100);
string s = HttpContext.Session.GetString("key1");
int i = HttpContext.Session.GetInt32("key2").Value;
```

convert object to json string and store

```
Employee emp = new Employee { EmpNo=1,Name="Vikram"};
string jsonEmp = JsonSerializer.Serialize<Employee>(emp);
HttpContext.Session.SetString("emp", jsonEmp);
```

```
//while reading it, reverse - deserialize
string e = HttpContext.Session.GetString("emp");
Employee emp = JsonSerializer.Deserialize<Employee>(e);
```

To read sessions in view - Program.cs

```
            builder.Services.AddHttpContextAccessor();
```

IN VIEW

```
@inject IHttpContextAccessor HttpContextAccessor
```

```
@{
    //string b = HttpContext.Session.GetString("b");
    string s =
HttpContextAccessor.HttpContext.Session.GetString("b");
```

104. ViewStart

- run at the beginning of every view
- Layout = ".."; - common layout

105 How to use your Layout

Layout = "..." in your view

how not to use any layout

Layout = null in your view

108 DataAnnotations in MVC (Important)

To Do – Read on DataAnnotation examples – see what type of Annotations are used for validation

All DataAnnotations inherit from ValidationAttribute class

How to give annotations in another class

```
    [ModelMetadataType(typeof(EmployeeMetadata))]
    public class Employee
    {
        public int EmpNo { get; set; }
        public string Name { get; set; }
```

```
    }
```

109 What are strongly typed views?

Linked to Model - @model in View

110 What is scaffolding? different templates used?

readymade code generated based on model. samples created for Index, Create, Delete, Edit, Details

111 What is a partial view?

can be shown within another view.

112 How to show partial view?

```
<partial name="pview">
```

```
@Html.PartialAsync("pview")
```

115 What are types of filters in MVC? (IMPORTANT)

Filters in ASP.NET Core allow code to run before or after specific stages in the request processing pipeline.

Authorization

Resource

Action

Result

EndPoint

Exception

Implement IFilterNameFilter

116 What does [Authorize] and [AllowAnonymous] do?

- allows only Authorized users to access controller/method
- allows ALL users to access controller/method

117 How to prevent Forgery attack?

in view

@Html.AntiForgeryToken()

asp-antiforgery="false"

in controller action method

 [ValidateAntiForgeryToken]

119 What are the diff approaches in EF Core?

Code first

Database first

? which is main class in EF core

DbContext

? which is collection class

DbSet<..>

?add, update, delete

_context.collection.Add()

```
_context.collection.Update()
_context.collection.Remove()
or
_context.Update()

or
_context.Entry(employee).State = EntityState.Deleted;



? how to save changes
_context.SaveChanges()
_context.SaveChangesAsync()

?codefirst - see steps
fluent api
or dataannotations

AddMigration --- creates a script to create db
UpdateDatabase – runs the migration script

?dbfirst - see steps
ScaffoldDbContext – to create classes and class derived from
dbcontext class



?conventional routing mvc
app.UseRouting();
app.MapControllerRoute(
```

```
    name: "default",
    pattern: "{controller=Home}/{action=Index}/{id?}");
```

?attribute routing mvc

app.MapControllers();

[Route("api/[controller])]

?types of authentication type in asp.net mvc project

No Authentication no authentication. You will have to implement your own authentication logic.

Individual User Accounts: This option will create an application with user authentication using ASP.NET Identity. It provides registration, login, and password reset features out of the box. Please notice: This will store all the user information inside the sql server by using the connection string.

Microsoft Identity Platform: This option will create an application that uses Azure Active Directory (Azure AD) for authentication. It is suitable for applications that require corporate authentication.

Windows Authentication: This option will create an application that uses Windows Authentication. It is suitable for intranet applications.

?web api

RESTful service

using controller approach

using minimal api - app.MapGet(...) app.MapPost(...) etc

```
?configuring cors
            //cors default for all sites
            builder.services.AddCors(c =>
            {
                c.AddDefaultPolicy(defpolicy =>
defpolicy.AllowAnyOrigin().AllowAnyHeader().AllowAnyMethod())
;
            });


            //cors for a particular site
            //builder.services.AddCors(c =>
            //{
            //    c.AddPolicy("PolicyNameHere", options =>
options.WithOrigins("http://localhost:12345").AllowAnyHeader(
).AllowAnyMethod());
            //});




            //allow cors
            app.UseCors(); //uses def policy
            //app.UseCors("PolicyNameHere"); //uses specific
policy


------


        //[EnableCors]  //enable the default policy for this
method
```

```
        //[EnableCors("PolicyNameHere")]  //enable the
specific policy for this method
        //[DisableCors] //disable cors
        //can also be done for the controller instead of
method
```

How to deploy the asp.net MVC project

Publish option

Project type in React for calling web api?

-- React.js with ASP.NET Core

How to create React SPA type application?

-- using Asp.Net Core Web App(Razor Pages)

- How to call web api from react app?

XMLHttpRequest

Fetch API

Axios