dot net framework  
application types

1. Windows application -> Stand alone- > Notepad.exe -> Independent  
2. Web Applicatoin -> google.com, Pantene.com- > websites are Hosted internet  
3. console Application-> Stand Alone  
4. XML Web Service ->   
Pantene.com -> Data to be explosed -> List of Users who have registered->   
Web service/WCF/Rest Full API service/Web API service  
5. Windows Service->   
6. Class Library project-> Error Handling or Email Sending-> Independent Entity

MS IDE-> Environment-> Visual Studio

PreRequistes->  
VS Installed  
SQL Server  
IIS-> Internet Information Services  
Admin rights

CLR, CTS and CLS  
Common Language RunTime  
Common Type System  
Common Language Specifications

VB.net and C#.net

VB-> integer-> VBC -> int 32  
C#-> int -> CSC -> int32

Common type-> Int32

Compiler -> Compile my code and check for compilation -> Artefacts

Artefacts-> dll, exe, .css,.js, .config, .cshtml, .aspx, .asmx, ascx

Managed Code -> Specifictions-> CLS, CTS -> Compile time-> CTS Complaint->  
CLR Manages the Memory-> GC-> Allocation and De allocation

int a =16;  
reserve some memory -> a-> Stack

Allocation and Deallocation

UnManaged Code-> CLS Complaint

ADO.NET->

Web Forms/Windows forms (presentation)  
ADO.NET components (command, Connection etc)  
Database(Data)

FCL/BCL-> Base Class Library/Framework Class Library

MSIL/IL-> Microsoft Intermidiate Language/ Intermediate Language

JIT-> Just In time compiler->  
IL Code-> Native Code

HostFileEditor Software  
https://hostsfileeditor.com/

DNS-> Domain Name Space Servers

DLL-> Dynamic Link Library -> Class Library Projects or Web sites/ Web Applications

Stand Alone application-> .exe

Jet brains Dot Peek

https://www.jetbrains.com/decompiler/

Assebmly ->Logical Grouping of code/program logic  
.exe/dll

1. Private Assebmly  
2. Shared/Public Assembly  
3. Satelite Assembly

Project 1 and Project 2  
Project 1 (private.exe)

Public/shared Assembly -> .exe/dll

GAC -> Global Assembly Cache -> DL  
some Steps to be followed  
Create/Generate Strong Name for the assembly  
4 Parts , Name, 4 Part version number

SateLite Assembly-> Culture settings  
Load the Culture Spefic artefacts for the particular project

US-> mm/dd/yyyy  
IN-> dd/mm/yyyy

Language syntax

Set of rules to be followed when some code in specific language  
C#-> Case Sensitive

1. Using Keyword  
2. Namespace -> Logical Grouping of Classes-> N number of Classses defined in one namespace  
3. Class -> Logical grouping of Methods/Properties/Variables/ etc  
4. Methods -> Functions/ sub  
5. Properties and Variables  
6. Access Modifier  
1. Private-> Scope is limited  
2. Public -> Anybody and Everybody (No Restrictions)  
3. Protected-> Inheritance-> the same base class and also to the Derived class  
4. Internal-> Limited to Assembly or the project  
5. Protected Internal->in the same project/assembly and only to the Derived Classes

7. Data Abstration Keywords -> Hiding

8. Constructors ->   
9. Data Types  
Storage (memory)  
1. Stack  
2. Heap

Two types of Data  
1. Value types  
For int, Float, Double etc-> Directly represent the value  
they are stored on Stack  
2. Reference types  
The Adress of the data  
Class, Interface, etc  
Does not contain the data directly  
They contain the address where the data is stored  
They are Stored on HEAP Memory

Data type conversions  
1. Implicit conversion , Long (int 64), Int (int 32)-> int32 -> Int64 a(int32)=b(int64) XXXX  
2. Explicit conversion

Variables  
1. Local Variables  
2. Instance Variables/ non Static Variables  
3. Static Variables  
4. Constant Variables  
5. ReadOnly Variables

C# Operators

1. Arithmatic Operator  
2. Relational Operator  
3. Logical Operator  
4. Assignment Operator  
5. Misc Operator

1. +,-,\*,/, ++,--

2. == (a==b),!= (a!=b),>,<,>=,<=- Out put of these operators can be Boolean Value only( True or False)

3. && (AND), ||, ! (NOT) (A&&B)-> !(T&&T)-> False  
boolean a  
boolean b

4. = (assign this value) a=b;-> Assign the value of variable b to a  
+= C+=A THIS WILL DO C=C+A  
-= C-=A THIS WIL DO C=C-A  
\*= C\*=A -> C=C\*A  
/= C/=A -> C=C/A

5.  
sizeof -> size of Data type -> int a =0; sizeof(a)->  
typeof-> int a =0; typeof(a)-> Int  
?: a>b?:a:b  
is -> if(Ford is car)-> Boolean  
as-> implicit conversion of the a datatype

dot net framework  
application types

1. Windows application -> Stand alone- > Notepad.exe -> Independent  
2. Web Applicatoin -> google.com, Pantene.com- > websites are Hosted internet  
3. console Application-> Stand Alone  
4. XML Web Service ->   
Pantene.com -> Data to be explosed -> List of Users who have registered->   
Web service/WCF/Rest Full API service/Web API service  
5. Windows Service->   
6. Class Library project-> Error Handling or Email Sending-> Independent Entity

MS IDE-> Environment-> Visual Studio

PreRequistes->  
VS Installed  
SQL Server  
IIS-> Internet Information Services  
Admin rights

CLR, CTS and CLS  
Common Language RunTime  
Common Type System  
Common Language Specifications

VB.net and C#.net

VB-> integer-> VBC -> int 32  
C#-> int -> CSC -> int32

Common type-> Int32

Compiler -> Compile my code and check for compilation -> Artefacts

Artefacts-> dll, exe, .css,.js, .config, .cshtml, .aspx, .asmx, ascx

Managed Code -> Specifictions-> CLS, CTS -> Compile time-> CTS Complaint->  
CLR Manages the Memory-> GC-> Allocation and De allocation

int a =16;  
reserve some memory -> a-> Stack

Allocation and Deallocation

UnManaged Code-> CLS Complaint

ADO.NET->

Web Forms/Windows forms (presentation)  
ADO.NET components (command, Connection etc)  
Database(Data)

FCL/BCL-> Base Class Library/Framework Class Library

MSIL/IL-> Microsoft Intermidiate Language/ Intermediate Language

JIT-> Just In time compiler->  
IL Code-> Native Code

HostFileEditor Software  
https://hostsfileeditor.com/

DNS-> Domain Name Space Servers

DLL-> Dynamic Link Library -> Class Library Projects or Web sites/ Web Applications

Stand Alone application-> .exe

Jet brains Dot Peek

https://www.jetbrains.com/decompiler/

Assebmly ->Logical Grouping of code/program logic  
.exe/dll

1. Private Assebmly  
2. Shared/Public Assembly  
3. Satelite Assembly

Project 1 and Project 2  
Project 1 (private.exe)

Public/shared Assembly -> .exe/dll

GAC -> Global Assembly Cache -> DL  
some Steps to be followed  
Create/Generate Strong Name for the assembly  
4 Parts , Name, 4 Part version number

SateLite Assembly-> Culture settings  
Load the Culture Spefic artefacts for the particular project

US-> mm/dd/yyyy  
IN-> dd/mm/yyyy

Language syntax

Set of rules to be followed when some code in specific language  
C#-> Case Sensitive

1. Using Keyword  
2. Namespace -> Logical Grouping of Classes-> N number of Classses defined in one namespace  
3. Class -> Logical grouping of Methods/Properties/Variables/ etc  
4. Methods -> Functions/ sub  
5. Properties and Variables  
6. Access Modifier  
1. Private-> Scope is limited  
2. Public -> Anybody and Everybody (No Restrictions)  
3. Protected-> Inheritance-> the same base class and also to the Derived class  
4. Internal-> Limited to Assembly or the project  
5. Protected Internal->in the same project/assembly and only to the Derived Classes

7. Data Abstration Keywords -> Hiding

8. Constructors ->   
9. Data Types  
Storage (memory)  
1. Stack  
2. Heap

Two types of Data  
1. Value types  
For int, Float, Double etc-> Directly represent the value  
they are stored on Stack  
2. Reference types  
The Adress of the data  
Class, Interface, etc  
Does not contain the data directly  
They contain the address where the data is stored  
They are Stored on HEAP Memory

Data type conversions  
1. Implicit conversion , Long (int 64), Int (int 32)-> int32 -> Int64 a(int32)=b(int64) XXXX  
2. Explicit conversion

Variables  
1. Local Variables  
2. Instance Variables/ non Static Variables  
3. Static Variables  
4. Constant Variables  
5. ReadOnly Variables

C# Operators

1. Arithmatic Operator  
2. Relational Operator  
3. Logical Operator  
4. Assignment Operator  
5. Misc Operator

1. +,-,\*,/, ++,--

2. == (a==b),!= (a!=b),>,<,>=,<=- Out put of these operators can be Boolean Value only( True or False)

3. && (AND), ||, ! (NOT) (A&&B)-> !(T&&T)-> False  
boolean a  
boolean b

4. = (assign this value) a=b;-> Assign the value of variable b to a  
+= C+=A THIS WILL DO C=C+A  
-= C-=A THIS WIL DO C=C-A  
\*= C\*=A -> C=C\*A  
/= C/=A -> C=C/A

5.  
sizeof -> size of Data type -> int a =0; sizeof(a)->  
typeof-> int a =0; typeof(a)-> Int  
?: a>b?:a:b  
is -> if(Ford is car)-> Boolean  
as-> implicit conversion of the a datatype