1. What is C#?

C# is an object oriented, type safe and managed language that is compiled by .Net framework to generate Microsoft Intermediate Language.

2. What are the types of comment in C# with examples?

- Single line

```
//This is a Single line comment
```

- Multiple line (/* */)

```
/*This is a multiple line commentWe are in line 2Last line of comment*/
```

- XML Comments (///)

```
/// summary;///  Set error message for multilingual language./// summary
```

3. Can multiple catch blocks be executed?

No, Multiple catch blocks can't be executed. **Once the proper catch code executed, the control is transferred to the finally block and then the code that follows the finally block gets executed.**

4. What is the difference between public, static and void?

- Public declared variables or methods are accessible **anywhere in the application.**
- Static declared variables or methods are globally **accessible without creating an instance of the class.**
- Void is a type modifier that states that the method or variable **does not return any value.**

5. What is an object?

An object is an instance of a class through which <u>we access the methods of that class.</u> "New" keyword is used to create an object.

6. Define Constructors?

A constructor is a member function in a class that has the same name as its class. The constructor is automatically invoked whenever an object class is created. It constructs the values of data members while initializing the class.

7. What is Jagged Arrays?

We can also call jagged array as Array of arrays.

8. What is the difference between ref & out parameters?

- **ref** must be initialized before passing to the method
- **out** parameter needs not to be initialized before passing to a method.

9. What is the use of using statement in C#?

The using block is used to obtain a resource and use it and then automatically dispose of when the execution of block completed.

10. What is serialization?

When we want to transport an object through network then we have to convert the object into a stream of bytes. The process of converting an object into a stream of bytes is called Serialization.

For an object to be serializable, it should implement ISerialize Interface.

11. Can "this" be used within a static method?

We can't use 'This' in a static method because we can only use static variables/methods in a static method.

12. What is difference between constants and read-only?

Constant variables **are declared and initialized at compile time.** The value can't be changed afterwards.

Read only is used only when **we want to assign the value at run time.**

13. What is an interface class?

Interface is an abstract class which has only public abstract methods and the methods only have the declaration and not the definition. These abstract methods must be implemented in the inherited classes.

14. What are value types and reference types?

- Value types are stored in the Stack
- reference types stored on heap.

Value types:

```
int, enum , byte, decimal, double, float, long
```

Reference Types:

```
string , class, interface, object
```

16. What are sealed classes in C#?

We create sealed classes when we want to restrict the class to be inherited.

Sealed modifier used to prevent derivation from a class.

If we forcefully specify a sealed class as base class then a compile-time error occurs.

17. What is method overloading?

Method overloading is creating multiple methods with the same name with unique signatures in the same class. When we compile, the compiler uses overload resolution to determine the specific method to be invoke.

18. What is the difference between Array and Arraylist?

In an array, we can have items of the same type only. The size of the array is fixed. An arraylist is similar to an array but it doesn't have a fixed size.

19. Can a private virtual method be overridden?

No, because they are not accessible outside the class.

20. Describe the accessibility modifier "protected internal".

Protected Internal variables/methods are accessible within the same assembly and also from the classes that are derived from this parent class.

21. What are the differences between System.String and System.Text.StringBuilder classes?

System.String is immutable. When we modify the value of a string variable then a new memory is allocated to the new value and the previous memory allocation released.

System.StringBuilder was designed to have concept of a mutable string where a variety of operations can be performed without allocation separate memory location for the modified string.

22. What's the difference between the System.Array.CopyTo() and System.Array.Clone() ?

Using Clone() method, we creates a new array object containing all the elements in the original array and using CopyTo() method, all the elements of existing array copies into another existing array. Both the methods perform a shallow copy.

23. How can we sort the elements of the array in descending order?

Using Sort() methods followed by Reverse() method.

24. Write down the C# syntax to catch exception?

To catch an exception, we use try catch blocks. Catch block can have parameter of system.Exception type.

Eg:

```
try {    GetAllData();} catch (Exception ex) {}
```

In the above example, we can omit the parameter from catch statement.

25. What's the difference between an interface and abstract class?

Interfaces have all the methods having only declaration but no definition. In an abstract class, we can have some concrete methods. In an interface class, all the methods are public. An abstract class may have private methods.

26. What is the difference between Finalize() and Dispose() methods?

- Dispose() is called when we want for an object to release any unmanaged resources with them.

- Finalize() is used for the same purpose but it doesn't assure the garbage collection of an object.

28. What are generics in C#.NET?

Generics are used to make reusable code classes to decrease the code redundancy, increase type safety and performance.

30. List down the commonly used types of exceptions in .Net?

ArgumentException, ArgumentNullException , ArgumentOutOfRangeException, ArithmeticException, DivideByZeroException ,OverflowException , IndexOutOfRangeException ,InvalidCastException ,InvalidOperationException , IOEndOfStreamException , NullReferenceException , OutOfMemoryException , StackOverflowException etc.

31. What are Custom Exceptions?

Sometimes there are some errors that need to be handeled as per user requirements.

32. What are delegates?

Delegates are same are function pointers in C++ but the only difference is that they are type safe unlike function pointers.

33. How do you inherit a class into other class in C#?

Colon is used as inheritance operator in C#.

```
public class DerivedClass : BaseClass
```

34. What is the base class in .net from which all the classes are derived from?

```
System.Object
```

35. What is the difference between method overriding and method overloading?

In method overriding, we change the method definition in the derived class that changes the method behavior. Method overloading is creating a method with the same name within the same class having different signatures.

36. What are the different ways a method can be overloaded?

Methods can be overloaded using different data types for parameter, different order of parameters, and different number of parameters.

37. Why can't you specify the accessibility modifier for methods inside the interface?

In an interface, we have virtual methods that do not have method definition. All the methods are there to be overridden in the derived class. That's why they all are public.

38. How can we set class to be inherited, but prevent the method from being over-ridden?

Declare the class as public and make the method sealed to prevent it from being overridden.

39. What happens if the inherited interfaces have conflicting method names?

Implement is up to you as the method is inside your own class. There might be problem when the methods from different interfaces expect different data, but as far as compiler cares you're okay.

40. What is the difference between a Struct and a Class?

Structs are value-type variables and classes are reference types.

Structs stored on the stack, causes additional overhead but faster retrieval. Structs cannot be inherited.

41. How to use nullable types in .Net?

Value types can take either their normal values or a null value.

```
Int? someID = null;If(someID.HasVAlue){}
```

42. How we can create an array with non-default values?

We can create an array with non-default values using Enumerable.Repeat.

43. What is difference between is and as operators in c#?

"is" operator is used to check the compatibility of an object with a given type and it returns the result as Boolean.

"as" operator is used for casting of object to a type or a class.

44. What's a multicast delegate?

A delegate having multiple handlers **assigned to it is called multicast delegate**. Each handler is assigned to a method.

45. What are indexers in C# .NET?

Indexers are known as smart arrays in C#. It allows the instances of a class to be indexed in the same way as array.

```
public int this[int index]    // Indexer declaration
```

46. What is difference between the "throw" and "throw ex" in .NET?

"Throw" statement preserves original error stack

 "throw ex" have the stack trace from their throw point.

It is always advised to use "throw" because it provides more accurate error information.

48. How to implement singleton design pattern in C#?

In singleton pattern, a class can only have one instance and provides access point to it globally.

```
Public sealed class Singleton{

Private static readonly Singleton _instance = new Singleton();

}
```

49. What is the difference between directcast and ctype?

DirectCast is used to convert the type of an object that requires the run-time type to be the same as the specified type in DirectCast.

Ctype is used for conversion where the conversion is defined between the expression and the type.

50. Is C# code is managed or unmanaged code?

C# is managed code because Common language runtime can compile C# code to Intermediate language.