1. **What is C#?**
   C# is a modern, object-oriented programming language developed by Microsoft for building a variety of applications.
2. **What is the .NET Framework?**
   The .NET Framework is a software development platform for building and running applications on Windows.
3. **What is the difference between .NET Core and .NET Framework?**
   .NET Core is a cross-platform, open-source framework, while .NET Framework is Windows-only and proprietary.
4. **What is a value type in C#?**
   A value type holds its data directly and includes types like int, float, and struct.
5. **What is a reference type in C#?**
   A reference type holds a reference to its data and includes types like classes, arrays, and strings.
6. **What is an array in C#?**
   An array is a fixed-size collection of elements of the same type.
7. **What is a List in C#?**
   A List is a dynamically-sized collection of objects that can be accessed by index.
8. **What is the purpose of the `using` statement in C#?**
   The `using` statement ensures that IDisposable objects are disposed of properly when they are no longer needed.
9. **What are constructors in C#?**
   Constructors are special methods called when an instance of a class is created, used for initialization.
10. **What is a destructor in C#?**
    A destructor is a special method called when an object is garbage collected, used for cleanup.

11. **What is encapsulation in C#?**
    Encapsulation is the practice of bundling data and methods that operate on that data within a single unit (class) and restricting access.
12. **What is inheritance in C#?**
    Inheritance is a mechanism where one class can inherit properties and methods from another class.
13. **What is polymorphism in C#?**
    Polymorphism allows methods to be implemented in different ways depending on the object invoking them, typically through method overriding and overloading.
14. **What is an interface in C#?**
    An interface defines a contract that classes can implement, containing method signatures without body implementations.
15. **What is an abstract class in C#?**
    An abstract class is a class that cannot be instantiated and may contain abstract methods that must be implemented by derived classes.
16. **What is the purpose of the static keyword in C#?**
    The static keyword indicates that a member belongs to the type itself rather than to a specific object.
17. **What is a delegate in C#?**
    A delegate is a type that references methods with a specific parameter list and return type, allowing method references to be passed around.
18. **What is an event in C#?**
    An event is a special delegate that allows a class to provide notifications to clients when something of interest occurs.
19. **What is LINQ in C#?**
    LINQ (Language Integrated Query) is a feature that allows querying of collections in a declarative manner using C# syntax.
20. **What is the purpose of the async and await keywords in C#?**
    The async keyword enables asynchronous programming, and await pauses the execution until the awaited task completes.

21. **What is the garbage collector in C#?**
    The garbage collector is an automatic memory management feature that frees up memory by reclaiming objects that are no longer in use.
22. **What is a property in C#?**
    A property is a member of a class that provides a flexible mechanism to read, write, or compute the value of a private field.
23. **What is the virtual keyword in C#?**
    The virtual keyword allows a method to be overridden in a derived class, enabling polymorphic behavior.
24. **What is the override keyword in C#?**
    The override keyword is used to provide a new implementation of a virtual method inherited from a base class.
25. **What is a nullable type in C#?**
    A nullable type can represent all values of its underlying type plus an additional null value, allowing for more flexible data handling.
26. **What is the difference between == and Equals() in C#?**
    == checks for reference equality (for reference types), while Equals() checks for value equality.
27. **What is a struct in C#?**
    A struct is a value type that can contain data members and methods, typically used for small data structures.
28. **What is the readonly keyword in C#?**
    The readonly keyword is used to declare fields that can only be assigned at their declaration or in a constructor.
29. **What is method overloading in C#?**
    Method overloading allows multiple methods in the same class to have the same name but different parameter lists.
30. **What is method overriding in C#?**
    Method overriding allows a derived class to provide a specific implementation of a method that is already defined in its base class.

31. **What is a namespace in C#?**
    A namespace is a container that holds a set of related classes, structs, enums, and interfaces to organize code.
32. **What is the purpose of the foreach loop in C#?**
    The foreach loop is used to iterate over collections, such as arrays or lists, without needing an index variable.
33. **What is exception handling in C#?**
    Exception handling is the process of responding to exceptional conditions or errors during program execution using try, catch, and finally blocks.
34. **What is the try-catch-finally construct in C#?**
    This construct is used to catch exceptions, execute cleanup code, and ensure that certain code runs regardless of exceptions.
35. **What is a static constructor in C#?**
    A static constructor initializes static members of a class and is called once before any static member is accessed.
36. **What is a dynamic type in C#?**
    The dynamic type allows variables to bypass static type checking, enabling flexibility at runtime.
37. **What is the difference between an abstract class and an interface in C#?**
    An abstract class can provide implementation for some methods, while an interface cannot provide any method implementation.
38. **What is a lambda expression in C#?**
    A lambda expression is a concise way to represent an anonymous method that can be used to create delegates or expression tree types.
39. **What is the params keyword in C#?**
    The params keyword allows a method to accept a variable number of arguments as a single array.
40. **What is a Tuple in C#?**
    A Tuple is a data structure that can hold a fixed number of items of different types, allowing for multiple return values.

41. **What is the purpose of the lock statement in C#?**
    The lock statement is used to ensure that a block of code runs by only one thread at a time, preventing race conditions.
42. **What is a thread in C#?**
    A thread is a lightweight process that can run concurrently with other threads, allowing parallel execution of code.
43. **What is Task Parallel Library (TPL) in C#?**
    The TPL is a set of APIs for simplifying parallel programming by using tasks instead of threads.
44. **What is the volatile keyword in C#?**
    The volatile keyword indicates that a field can be accessed by multiple threads and prevents compiler optimizations.
45. **What is a Singleton pattern in C#?**
    A Singleton pattern restricts a class to a single instance and provides a global access point to that instance.
46. **What is Dependency Injection in C#?**
    Dependency Injection is a design pattern that allows a class to receive its dependencies from an external source rather than creating them internally.
47. **What is an attribute in C#?**
    An attribute is a declarative tag that is added to a class, method, or property to provide metadata that can be inspected at runtime.
48. **What is reflection in C#?**
    Reflection is the ability to inspect and interact with types, members, and attributes at runtime.
49. **What is a generic type in C#?**
    A generic type allows you to define classes, interfaces, and methods with a placeholder for the type, providing type safety and reusability.
50. **What is an enumerator in C#?**
    An enumerator is a construct that allows you to iterate over a collection, such as an array or a list.

51. **What is a finalizer in C#?**
    A finalizer is a special method that is called by the garbage collector before an object is reclaimed, allowing for cleanup.
52. **What is the purpose of the `new` keyword in C#?**
    The `new` keyword is used to create instances of objects or to hide a member inherited from a base class.
53. **What is a property initializer in C#?**
    A property initializer allows you to assign a value to a property at the point of declaration.
54. **What is a switch expression in C#?**
    A switch expression is a concise way to evaluate multiple conditions and return values based on those conditions.
55. **What is a record type in C#?**
    A record type is a reference type designed for immutable data and provides built-in functionality for equality comparison.
56. **What is a static member in C#?**
    A static member belongs to the class itself rather than to any specific instance, shared across all instances.
57. **What is the `const` keyword in C#?**
    The `const` keyword defines a constant field or local variable whose value cannot be modified.
58. **What is the `readonly` keyword in C#?**
    The `readonly` keyword defines a field that can only be assigned during declaration or in a constructor.
59. **What is the `dynamic` keyword in C#?**
    The `dynamic` keyword allows variables to be dynamically typed at runtime, enabling late binding.
60. **What is a `Task` in C#?**
    A `Task` represents an asynchronous operation that can be awaited, allowing for concurrent execution.

61. **What is the purpose of IEnumerable in C#?**
    IEnumerable provides a way to iterate over a collection without exposing the underlying collection structure.
62. **What is a LINQ query syntax in C#?**
    LINQ query syntax is a declarative way to query collections using a SQL-like syntax.
63. **What is an implicit type in C#?**
    An implicit type is defined using the var keyword, allowing the compiler to infer the type at compile time.
64. **What is a lambda statement in C#?**
    A lambda statement is a concise way to represent an anonymous function in a single line.
65. **What is a List<T> in C#?**
    List<T> is a generic collection that dynamically resizes, allowing for the storage of objects of a specified type.
66. **What is a Dictionary<TKey, TValue> in C#?**
    A Dictionary<TKey, TValue> is a collection of key-value pairs, providing fast lookup by keys.
67. **What is a Stack<T> in C#?**
    A Stack<T> is a last-in-first-out (LIFO) collection that allows adding and removing elements from the top.
68. **What is a Queue<T> in C#?**
    A Queue<T> is a first-in-first-out (FIFO) collection that allows adding elements to the end and removing from the front.
69. **What is an IQueryable in C#?**
    IQueryable represents a queryable collection that can be executed against a data source, supporting LINQ queries.
70. **What is a using directive in C#?**
    A using directive allows the use of types from a namespace without needing to specify the full namespace.

71. **What is an extension method in C#?**
    An extension method allows adding new methods to existing types without modifying the original type.
72. **What is the purpose of the base keyword in C#?**
    The base keyword refers to the base class from which the current class derives, allowing access to its members.
73. **What is the this keyword in C#?**
    The this keyword refers to the current instance of a class or struct.
74. **What is a Nullable<T> type in C#?**
    Nullable<T> allows value types to be assigned a null value, representing the absence of a value.
75. **What is a thread-safe collection in C#?**
    A thread-safe collection is designed to be safely accessed by multiple threads simultaneously.
76. **What is an anonymous type in C#?**
    An anonymous type is a type created on the fly without a name, often used to hold a set of values.
77. **What is the purpose of the await keyword in C#?**
    The await keyword is used to asynchronously wait for a task to complete, without blocking the calling thread.
78. **What is the nameof operator in C#?**
    The nameof operator returns the name of a variable, type, or member as a string, improving code maintainability.
79. **What is the purpose of the params keyword in a method signature in C#?**
    The params keyword allows passing a variable number of arguments to a method, represented as an array.
80. **What is a using block in C#?**
    A using block ensures that resources are disposed of once the block of code is executed.

81. **What is an IEnumerable<T> in C#?**
    IEnumerable<T> represents a collection of objects that can be enumerated, providing a way to iterate through the collection.
82. **What is a Func<T> delegate in C#?**
    A Func<T> delegate represents a method that returns a value and can take parameters, allowing for higher-order functions.
83. **What is an Action<T> delegate in C#?**
    An Action<T> delegate represents a method that takes parameters but does not return a value.
84. **What is a ValueTuple in C#?**
    A ValueTuple is a lightweight alternative to a Tuple that can hold multiple values and supports named fields.
85. **What is a dynamic object in C#?**
    A dynamic object bypasses compile-time type checking, allowing for flexible interactions with objects.
86. **What is the Async/Await pattern in C#?**
    The Async/Await pattern simplifies asynchronous programming, allowing developers to write code that looks synchronous.
87. **What is a TaskCompletionSource<T> in C#?**
    A TaskCompletionSource<T> represents a task that can be completed manually, allowing more control over task completion.
88. **What is a try-finally block in C#?**
    A try-finally block ensures that the code in the finally block runs regardless of whether an exception is thrown.
89. **What is a Region in C#?**
    A region is a code block that can be expanded or collapsed in the IDE, improving code organization.
90. **What is an awaitable type in C#?**
    An awaitable type is a type that has a method named GetAwaiter() to enable the use of the await keyword.

91. **What is the purpose of the async modifier in a method?**
The async modifier enables the use of the await keyword inside the method, allowing for asynchronous execution.

92. **What is a property setter in C#?**
A property setter is a method that assigns a value to a property and can include additional logic for validation.

93. **What is a property getter in C#?**
A property getter is a method that retrieves the value of a property and can include logic to compute the value.

94. **What is a Task.Delay() in C#?**
Task.Delay() creates a task that completes after a specified time interval, often used for simulating asynchronous delays.

95. **What is the purpose of the yield keyword in C#?**
The yield keyword is used to return each element in a collection one at a time in an iterator.

96. **What is a FileStream in C#?**
A FileStream provides a way to read from and write to files, allowing for file input/output operations.

97. **What is a MemoryStream in C#?**
A MemoryStream is a stream that uses memory as its backing store, allowing for in-memory file operations.

98. **What is an ActionResult in ASP.NET MVC?**
An ActionResult represents the result of an action method, determining what response is sent to the client.

99. **What is a model in ASP.NET MVC?**
A model represents the data and business logic in an ASP.NET MVC application, often used for data transfer between the controller and view.

100. **What is a view in ASP.NET MVC?**
A view is a template that defines how data from the model is presented to the user, typically using HTML.