

### 1-10:

1. **C# is a strongly typed language.**
  - True
  - **Answer: True**
  
2. **In C#, 'null' is considered an object.**
  - True
  - **Answer: False**
  
3. **The 'using' statement in C# is used to include namespaces.**
  - True
  - **Answer: True**
  
4. **C# supports multiple inheritance.**
  - True
  - **Answer: False**
  
5. **A 'struct' in C# is a value type.**
  - True
  - **Answer: True**
  
6. **C# allows you to override static methods.**
  - True
  - **Answer: False**
  
7. **In C#, a 'readonly' field can only be assigned in the constructor.**
  - True
  - **Answer: True**
  
8. **The 'var' keyword in C# is used for dynamic typing.**
  - True

- \*\*Answer: False\*\*

9. \*\*All exceptions in C# are derived from the System.Exception class.\*\*

- True

- \*\*Answer: True\*\*

10. \*\*C# allows operator overloading.\*\*

- True

- \*\*Answer: True\*\*

### 11-20:

11. \*\*The 'dynamic' keyword in C# is resolved at compile time.\*\*

- True

- \*\*Answer: False\*\*

12. \*\*C# allows a method to have optional parameters.\*\*

- True

- \*\*Answer: True\*\*

13. \*\*The 'finally' block in C# will execute even if an exception is thrown.\*\*

- True

- \*\*Answer: True\*\*

14. \*\*In C#, an abstract class can be instantiated.\*\*

- True

- \*\*Answer: False\*\*

15. \*\*The 'sealed' keyword in C# prevents a class from being inherited.\*\*

- True

- \*\*Answer: True\*\*

16. **C# is case-insensitive.**

- True

- **Answer: False**

17. **C# supports both reference and value types.**

- True

- **Answer: True**

18. **The 'lock' statement in C# is used for thread synchronization.**

- True

- **Answer: True**

19. **LINQ queries in C# can only be used with collections.**

- True

- **Answer: False**

20. **C# does not support pointer types.**

- True

- **Answer: False**

### 21-30:

21. **A delegate in C# is a type-safe function pointer.**

- True

- **Answer: True**

22. **C# allows multiple constructors in a class.**

- True

- **Answer: True**

23. **C# arrays are zero-based.**

- True

- \*\*Answer: True\*\*

24. \*\*The 'is' operator in C# checks for reference equality.\*\*

- True

- \*\*Answer: False\*\*

25. \*\*'null' is a keyword in C#.

- True

- \*\*Answer: True\*\*

26. \*\*C# supports both pre- and post-increment operators.

- True

- \*\*Answer: True\*\*

27. \*\*A method marked as 'virtual' in C# must be overridden in a derived class.

- True

- \*\*Answer: False\*\*

28. \*\*The 'default' keyword in C# can be used with switch cases and generics.

- True

- \*\*Answer: True\*\*

29. \*\*C# does not support covariance and contravariance in generic types.

- True

- \*\*Answer: False\*\*

30. \*\*The 'System.Object' class is the base class for all types in C#.

- True

- \*\*Answer: True\*\*

### 31-40:

31. **\*\*The 'const' keyword in C# allows you to declare a constant value that cannot change.\*\***

- True

- **\*\*Answer: True\*\***

32. **\*\*A 'try' block in C# must be followed by at least one 'catch' block.\*\***

- True

- **\*\*Answer: False\*\***

33. **\*\*A method can have a 'params' parameter in C#.\*\***

- True

- **\*\*Answer: True\*\***

34. **\*\*The 'ref' keyword in C# passes arguments by value.\*\***

- True

- **\*\*Answer: False\*\***

35. **\*\*Events in C# are built on top of delegates.\*\***

- True

- **\*\*Answer: True\*\***

36. **\*\*The 'sealed' keyword in C# can be applied to methods.\*\***

- True

- **\*\*Answer: True\*\***

37. **\*\*In C#, a derived class can hide a base class method using the 'new' keyword.\*\***

- True

- **\*\*Answer: True\*\***

38. **\*\*The 'internal' access modifier in C# is more restrictive than 'private'.\*\***

- True

- **\*\*Answer: False\*\***

39. **\*\*A C# 'enum' can contain both string and integer values.\*\***

- True

- **\*\*Answer: False\*\***

40. **\*\*C# allows recursion in methods.\*\***

- True

- **\*\*Answer: True\*\***

### 41-50:

41. **\*\*The 'readonly' keyword in C# allows a field to be modified after object construction.\*\***

- True

- **\*\*Answer: False\*\***

42. **\*\*In C#, interfaces can define fields.\*\***

- True

- **\*\*Answer: False\*\***

43. **\*\*'Partial classes' in C# allow a class definition to be split across multiple files.\*\***

- True

- **\*\*Answer: True\*\***

44. **\*\*C# supports conditional compilation with preprocessor directives.\*\***

- True

- **\*\*Answer: True\*\***

45. **\*\*The 'default' access modifier in C# is 'protected'.\*\***

- True

- **\*\*Answer: False\*\***

46. **\*\*The 'protected internal' access modifier in C# is more restrictive than 'internal'.\*\***

- True
- \*\*Answer: False\*\*

47. \*\*In C#, all exceptions must be caught or declared to be thrown.\*\*

- True
- \*\*Answer: False\*\*

48. \*\*C# supports asynchronous programming using the 'async' and 'await' keywords.\*\*

- True
- \*\*Answer: True\*\*

49. \*\*In C#, the 'out' keyword requires the argument to be initialized before passing it to a method.\*\*

- True
- \*\*Answer: False\*\*

50. \*\*An 'interface' in C# can inherit from multiple interfaces.\*\*

- True
- \*\*Answer: True\*\*

### 51-60:

51. \*\*C# has built-in support for dynamic typing using the 'dynamic' keyword.\*\*

- True
- \*\*Answer: True\*\*

52. \*\*In C#, the 'new' keyword can be used to create instances of a type.\*\*

- True
- \*\*Answer: True\*\*

53. \*\*The 'extern' keyword in C# is used to declare a method that is implemented externally.\*\*

- True

- \*\*Answer: True\*\*

54. \*\*C# does not support operator overloading for user-defined types.\*\*

- True

- \*\*Answer: False\*\*

55. \*\*C# supports nullable value types.\*\*

- True

- \*\*Answer: True\*\*

56. \*\*In C#, you cannot override a non-virtual method in a derived class.\*\*

- True

- \*\*Answer: True\*\*

57. \*\*The 'goto' statement in C# can be used to jump to another method.\*\*

- True

- \*\*Answer: False\*\*

58. \*\*An abstract class in C# can contain a constructor.\*\*

- True

- \*\*Answer: True\*\*

59. \*\*C# allows you to define extension methods for existing types.\*\*

- True

- \*\*Answer: True\*\*

60. \*\*The 'protected' access modifier in C# allows access from within the same assembly.\*\*

- True

- \*\*Answer: False\*\*

### 61-70:



61. **\*\*In C#, the 'throw' keyword is used to raise exceptions.\*\***

- True

- **\*\*Answer: True\*\***

62. **\*\*A 'delegate' in C# cannot be used to invoke multiple methods.\*\***

- True

- **\*\*Answer: False\*\***

63. **\*\*The 'volatile' keyword in C# ensures that the value of a field is always read from memory, not from a cache.\*\***

- True

- **\*\*Answer: True\*\***

64. **\*\*C# supports anonymous methods using the 'delegate' keyword.\*\***

- True

- **\*\*Answer: True\*\***

65. **\*\*The 'readonly' keyword in C# can be applied to methods.\*\***

- True

- **\*\*Answer: False\*\***

66. **\*\*In C#, a 'struct' can inherit from another struct.\*\***

- True

- **\*\*Answer: False\*\***

67. **\*\*C# supports method overloading.\*\***

- True

- **\*\*Answer: True\*\***

68. **\*\*In C#, the 'params' keyword**

allows a method to accept a variable number of arguments.\*\*

- True

- \*\*Answer: True\*\*

69. \*\*The 'System.String' class in C# is mutable.\*\*

- True

- \*\*Answer: False\*\*

70. \*\*In C#, 'int' and 'System.Int32' are the same type.\*\*

- True

- \*\*Answer: True\*\*

### 71-80:

71. \*\*The 'checked' keyword in C# is used to enable overflow checking for arithmetic operations.\*\*

- True

- \*\*Answer: True\*\*

72. \*\*In C#, the 'base' keyword is used to refer to the current instance of a class.\*\*

- True

- \*\*Answer: False\*\*

73. \*\*A 'static' constructor in C# cannot have parameters.\*\*

- True

- \*\*Answer: True\*\*

74. \*\*In C#, the 'continue' statement can be used in a 'switch' statement.\*\*

- True

- \*\*Answer: False\*\*

75. \*\*The 'foreach' loop in C# can be used to iterate over an array.\*\*

- True

- \*\*Answer: True\*\*

76. \*\*The 'object' keyword in C# is an alias for System.Object.\*\*

- True

- \*\*Answer: True\*\*

77. \*\*C# supports both synchronous and asynchronous exception handling.\*\*

- True

- \*\*Answer: False\*\*

78. \*\*In C#, 'double' and 'System.Double' are different types.\*\*

- True

- \*\*Answer: False\*\*

79. \*\*The 'nameof' operator in C# returns the name of a variable as a string.\*\*

- True

- \*\*Answer: True\*\*

80. \*\*In C#, 'int?' is a nullable integer type.\*\*

- True

- \*\*Answer: True\*\*

### 81-90:

81. \*\*The 'protected' keyword in C# allows access to class members only within the same class.\*\*

- True

- \*\*Answer: False\*\*

82. \*\*The 'throw' keyword in C# can only be used within a catch block.\*\*

- True

- \*\*Answer: False\*\*

83. **\*\*In C#, an interface can contain constructors.\*\***

- True

- **\*\*Answer: False\*\***

84. **\*\*C# allows methods to be marked as 'partial'.\*\***

- True

- **\*\*Answer: True\*\***

85. **\*\*The 'is' operator in C# is used for pattern matching.\*\***

- True

- **\*\*Answer: True\*\***

86. **\*\*The 'System.Nullable' type in C# can be used with reference types.\*\***

- True

- **\*\*Answer: False\*\***

87. **\*\*In C#, a method marked as 'abstract' must be implemented in derived classes.\*\***

- True

- **\*\*Answer: True\*\***

88. **\*\*The 'global::' prefix in C# is used to refer to the global namespace.\*\***

- True

- **\*\*Answer: True\*\***

89. **\*\*A 'ref' parameter in C# does not require the argument to be initialized before being passed to a method.\*\***

- True

- **\*\*Answer: False\*\***

90. **\*\*The 'default' keyword in C# can be used to initialize any type.\*\***

- True

- \*\*Answer: True\*\*

### 91-100:

91. \*\*In C#, 'out' parameters must be initialized before returning from a method.\*\*

- True

- \*\*Answer: True\*\*

92. \*\*The 'new' keyword in C# can be used to hide a base class method.\*\*

- True

- \*\*Answer: True\*\*

93. \*\*In C#, a 'class' can inherit from multiple classes.\*\*

- True

- \*\*Answer: False\*\*

94. \*\*C# does not allow defining methods inside methods.\*\*

- True

- \*\*Answer: False\*\*

95. \*\*A method in C# can be both 'static' and 'abstract'.\*\*

- True

- \*\*Answer: False\*\*

96. \*\*C# allows defining constants using the 'const' keyword.\*\*

- True

- \*\*Answer: True\*\*

97. \*\*The 'await' keyword in C# can only be used within an 'async' method.\*\*

- True

- \*\*Answer: True\*\*

98. **\*\*In C#, the 'virtual' keyword is used to define methods that can be overridden in derived classes.\*\***

- True

- **\*\*Answer: True\*\***

99. **\*\*In C#, a 'switch' statement can be used with string values.\*\***

- True

- **\*\*Answer: True\*\***

100. **\*\*The 'volatile' keyword in C# can be applied to local variables.\*\***

- True

- **\*\*Answer: False\*\***