**1. What is an indexer in C#?**

A) A method that returns an array  
B) A special type of property that allows objects to be indexed like an array  
C) A class used for managing database indexes  
D) A function that maps values to keys  
**Answer:** B) A special type of property that allows objects to be indexed like an array

**2. How is an indexer defined in C#?**

A) Using the index keyword  
B) Using the this keyword  
C) Using the array keyword  
D) Using the indexer keyword  
**Answer:** B) Using the this keyword

**3. What accessors are typically used in an indexer?**

A) Only get  
B) Only set  
C) get and set  
D) Neither get nor set  
**Answer:** C) get and set

**4. Which of the following is true about indexers?**

A) They must have a name like a property  
B) They are defined using the this keyword  
C) They cannot be overloaded  
D) They cannot have different data types for the index parameter  
**Answer:** B) They are defined using the this keyword

**5. What is the purpose of an indexer in C#?**

A) To access elements of a class using an index similar to an array  
B) To replace the need for properties  
C) To store multiple class instances  
D) To perform arithmetic operations  
**Answer:** A) To access elements of a class using an index similar to an array

**6. What will happen if an index value is out of range in an indexer?**

A) The program will crash  
B) A default value or an empty string can be returned  
C) An exception will always be thrown  
D) The compiler will catch the error before execution  
**Answer:** B) A default value or an empty string can be returned

**7. Can indexers in C# be overloaded?**

A) Yes, by defining multiple indexers with different parameter types  
B) No, indexers cannot be overloaded  
C) Yes, but only within an interface  
D) No, but they can be inherited  
**Answer:** A) Yes, by defining multiple indexers with different parameter types

**8. What data types can be used as an index in an indexer?**

A) Only integers  
B) Only strings  
C) Any type including integers and strings  
D) Only floating-point numbers  
**Answer:** C) Any type including integers and strings

**9. Which of the following statements about indexers is false?**

A) They can be defined in a struct  
B) They can be static  
C) They can be inherited  
D) They can return different types of values  
**Answer:** B) They can be static

**10. In the example below, what will names["Nuha"] return?**

IndexedNames names = new IndexedNames();

names[0] = "Zara";

names[1] = "Riz";

names[2] = "Nuha";

names[3] = "Asif";

Console.WriteLine(names["Nuha"]);

A) 2  
B) "Nuha"  
C) "Zara"  
D) An error  
**Answer:** A) 2