**1. What is an Anonymous Method in C#?**

A) A method without parameters  
B) A method without a return type  
C) A method without a name  
D) A method that runs automatically

**Answer:** C) A method without a name

**2. Which keyword is used to declare an Anonymous Method?**

A) delegate  
B) anonymous  
C) lambda  
D) var

**Answer:** A) delegate

**3. What is the main advantage of using Anonymous Methods?**

A) They increase performance  
B) They allow defining inline methods without explicitly declaring them  
C) They use less memory  
D) They cannot be assigned to delegates

**Answer:** B) They allow defining inline methods without explicitly declaring them

**4. How is an Anonymous Method assigned to a delegate?**

A) delegate void NumberChanger(int n) { Console.WriteLine(n); };  
B) NumberChanger nc = delegate(int x) { Console.WriteLine(x); };  
C) Anonymous nc = new delegate(int x) { Console.WriteLine(x); };  
D) void nc = delegate(int x) { Console.WriteLine(x); };

**Answer:** B) NumberChanger nc = delegate(int x) { Console.WriteLine(x); };

**5. Can an Anonymous Method access local variables from its enclosing method?**

A) Yes, but only if they are constants  
B) Yes, but they must be declared as static  
C) No, Anonymous Methods cannot access local variables  
D) Yes, they can access local variables

**Answer:** D) Yes, they can access local variables

**6. What is the return type of an Anonymous Method?**

A) The return type must always be void  
B) The return type is inferred from the return statement inside the method  
C) It must be explicitly defined  
D) Anonymous Methods do not return any value

**Answer:** B) The return type is inferred from the return statement inside the method

**7. Which of the following is NOT true about Anonymous Methods?**

A) They can be assigned to delegates  
B) They can contain multiple statements  
C) They can have parameters  
D) They can be overloaded

**Answer:** D) They can be overloaded

**8. Can an Anonymous Method be used with built-in delegates like Func<T> and Action<T>?**

A) No, only named methods can be assigned to these delegates  
B) Yes, Anonymous Methods can be assigned to Func<T> and Action<T>  
C) Only if they have a return statement  
D) Only inside a class

**Answer:** B) Yes, Anonymous Methods can be assigned to Func<T> and Action<T>

**9. How do Anonymous Methods handle multiple parameters?**

A) They do not support parameters  
B) Parameters must be declared inside the method body  
C) They accept parameters just like normal methods  
D) They must be passed using ref or out

**Answer:** C) They accept parameters just like normal methods

**10. What is the difference between Anonymous Methods and Lambda Expressions?**

A) Anonymous Methods are faster than Lambda Expressions  
B) Lambda Expressions provide a more concise syntax than Anonymous Methods  
C) Anonymous Methods can be used in LINQ queries, but Lambda Expressions cannot  
D) There is no difference between them

**Answer:** B) Lambda Expressions provide a more concise syntax than Anonymous Methods