# .NET MCQ

1. What happens when you compile this code?

```
var query = from e in employees
            where e.Age > 30
            select e.Name, e.Age;
```

- A. It works fine and returns Name and Age together.

- B. It only works if you write using System.Tuple;.

- C. You can replace the comma with into to fix it.

- D. It gives an error because select must return a single object.

- Answer: D

2. You are working on a banking application that saves `Customer` objects to disk. The `Customer` class is given below. When serializing a `Customer` object, which field/property will **NOT** be saved to disk?

```
[Serializable]
public class Customer
{
    public string Name { get; set; }
    public int Age { get; set; }
    [NonSerialized] private string _password;
    public string TemporarySessionToken { get; set; }
}
```

- A. `Name`

- B. `_password` & `TemporarySessionToken`

- C. `_password`

- D. `TemporarySessionToken`

- Answer: C

---

3. What happens when you compile this code?

```csharp
public class ConfigLoader {
    var _config = LoadConfig();   // Line 1
    public void Init() {
        var config = _config;     // Line 2
        var = "test";             // Line 3
        var timeout = 30;         // Line 4
        Console.WriteLine(timeout.GetType().Name);
    }
    private Dictionary<string, string> LoadConfig() => new();
}
```

- A. Output: Int32

- B. Compile error at Line 1: "var cannot be used here" and Line 3: "Invalid expression"

- C. Output: Dictionary`2

- D. Compile error at Line 1 and runtime error at Line 4

- Answer: B

---

4. Which of the following statements about module initializers is FALSE?

- A. Module initializers execute before any other method in the assembly, including static constructors.

- B. A single assembly can contain multiple module initializer methods.

- C. Module initializers can be defined as instance methods in non-static classes.

- D. If a module initializer throws an exception, the application fails immediately

- Answer: C

---

5. With nullable reference types enabled What does the compiler do?

```
string nonNullableString = "Hello";
var inferredString = nonNullableString;
inferredString = null;
```

- A. No warnings – var infers non-nullable string.

- B. Warning – var infers string? (nullable).

- C. Error – var does not allow null assignments.

- D. No warnings – var acts like dynamic.

- Answer: A

---

6. In a WinForms app, the auto-generated MainForm.Designer.cs declares a partial method:
  - partial void InitializeCustomComponents();
  - A developer implements it in MainForm.cs:

```
partial void InitializeCustomComponents() {
    // ...
}
```

- Later, another developer adds a second implementation in MainForm.Experimental.cs. What happens?

- A. Compile error – Partial methods allow only one implementation

- B. Runtime error – Duplicate method execution

- C. Silent merge – Both implementations run

- D. Warning only – Code works with a compiler warning

- Answer: A

---

7. Which of the following statements about extension methods is true?

- A. An extension method declared in a static class can override an instance method on the same type if it has the same signature.

- B. Extension methods can be declared as static methods in any class, even if the class itself is not static, as long as the method is static and public.

- C. If an extension method and an instance method have the same name and signature, the compiler will always prefer the extension method.

- D. Extension methods must be static methods declared inside a static class, and the first parameter must be prefixed with "this" to specify the type they extend.

- Answer: D

---

8. What will be the output of the following code snippet?

```
class Program {
    static void Main(string[] args) {
```

```
        int[] nums = {3, 1, 2, 5, 4};
        var ltAvg = from n in nums
                        let x = nums.Average()
                        where n < x
                        select n;
        Console.WriteLine("The average is " + nums.Average());
    }
}
```

- A. Run time error

- B. 3

- C. 1, 2

- D. Compile time error

- Answer: B

---

9. Consider this StreamWriter constructor - Which statement about the parameters is FALSE?
   - new StreamWriter(string path, bool append)

- A. path must be a full absolute path – relative paths always throw ArgumentException

- B. When append is true, the writer preserves original content but does not guarantee atomic writes.

- C. If the file at path is locked by another process, this will throw IOException regardless of append value.

- D. Path can be a UNC network path (e.g., `\\server\file.txt`).

- Answer: A

---

10. Which of the following statements about the ArgumentException class is correct?

- A. ArgumentException is thrown when a method receives a null argument that is not allowed; therefore, it is the same as ArgumentNullException.

- B. ArgumentException is in the System.Collections namespace.

- C. ArgumentException indicates that at least one of the arguments passed to a method does not meet the parameter requirements.

- D. ArgumentException can only be thrown by the .NET runtime and cannot be thrown explicitly in your code.

- Answer: C