CsharpDay04Assignment

Name: Mohamed Waleed Elkady

Part01

Question1: What is the default value assigned to array elements in C#?
Answer: In C#, the default value of an array element depends on the type of the array. For numeric types (int, double, etc.), the default is 0. For bool, it is false. For reference types (like objects or strings), the default is null.

Question2: What is the difference between Array.Clone() and Array.Copy()?
Answer: `Array.Clone()` creates a shallow copy of the entire array, meaning that for reference types, the references are copied but the objects themselves are not duplicated. `Array.Copy()` copies elements from a source array to a destination array, and you can specify which portion to copy and where to place it. `Array.Copy()` can be used for partial copying, while `Clone()` copies the whole array.

Question3: What is the difference between GetLength() and Length for multi-dimensional arrays?
Answer: `Length` returns the total number of elements in the entire multi-dimensional array. `GetLength(dimension)` returns the number of elements along a specific dimension (row or column). For example, for a 3x3 array, `Length` is 9, but `GetLength(0)` is 3 (rows) and `GetLength(1)` is 3 (columns).

Question4: What is the difference between Array.Copy() and Array.ConstrainedCopy()?
Answer: `Array.Copy()` copies elements from one array to another but does not guarantee that all changes are undone if an error occurs during copying. `Array.ConstrainedCopy()` ensures that if an exception occurs during the copy process, the destination array remains unchanged. It is safer for critical operations where partial copying is unacceptable.

Question5: Why is foreach preferred for read-only operations on arrays?
Answer: `foreach` is preferred for read-only operations because it automatically iterates over all

elements without needing an index and prevents accidental modification of the array elements. It also makes the code cleaner and less error-prone.

Question6: Why is input validation important when working with user inputs?
Answer: Input validation is important because user input can be incorrect, malformed, or malicious. Validating ensures that the program behaves correctly, avoids runtime errors, prevents security vulnerabilities, and provides a better user experience.

Question7: How can you format the output of a 2D array for better readability?
Answer: You can format a 2D array by using nested loops to print elements row by row, adding spacing or tab characters (`\t`) between elements, and optionally aligning columns for a clean matrix-like appearance.

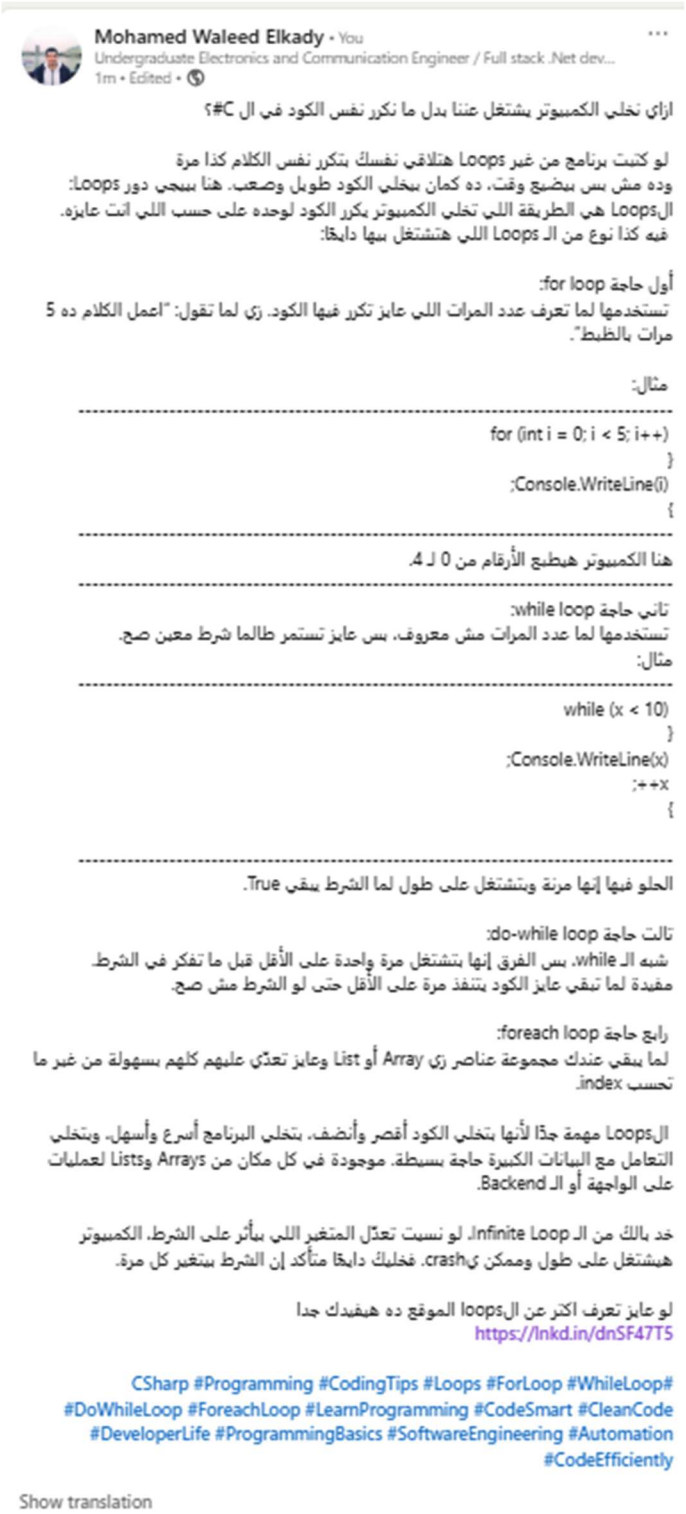
Question8: When should you prefer a switch statement over if-else?
Answer: A `switch` statement is preferred when you need to compare a single variable against multiple constant values because it is more readable, organized, and often more efficient than multiple `if-else` statements. `If-else` is better for complex conditions or ranges.

Question9: What is the time complexity of Array.Sort()?
Answer: The time complexity of `Array.Sort()` in C# is O(n log n) on average, as it typically uses a variation of the QuickSort or IntroSort algorithm internally.

Question10: Which loop (for or foreach) is more efficient for calculating the sum of an array, and why?
Answer: For simple arrays of value types like integers, a `for` loop is slightly more efficient than `foreach` because it avoids the overhead of the enumerator object used by `foreach`. However, the difference is minimal and often negligible in practice.

Part02

1- The linkedin article:-



Mohamed Waleed Elkady • You
Undergraduate Electronics and Communication Engineer / Full stack .Net dev...
1m • Edited •

ازاي نخلي الكمبيوتر بشتغل عننا بدل ما نكرر نفس الكود في ال C#؟

لو كتبت برنامج من غير Loops هتلاقي نفسك بتكرر نفس الكلام كذا مرة
وده مش بس بيضيع وقت، ده كمان بيخلي الكود طويل وصعب. هنا بييجي دور Loops:
الLoops هي الطريقة اللي تخلي الكمبيوتر يكرر الكود لوحده على حسب اللي انت عايزه.
فيه كذا نوع من الـ Loops اللي هتشتغل بيها دايمًا:

أول حاجة for loop:
تستخدمها لما تعرف عدد المرات اللي عايز تكرر فيها الكود. زي لما تقول: "اعمل الكلام ده 5 مرات بالظبط".

مثال:

```
for (int i = 0; i < 5; i++)
{
    Console.WriteLine(i);
}
```

هنا الكمبيوتر هيطبع الأرقام من 0 لـ 4.

تاني حاجة while loop:
تستخدمها لما عدد المرات مش معروف، بس عايز تستمر طالما شرط معين صح.
مثال:

```
while (x < 10)
{
    Console.WriteLine(x);
    ++x;
}
```

الحلو فيها إنها مرنة وبتشتغل على طول لما الشرط يبقى True.

تالت حاجة do-while loop:
شبه الـ while، بس الفرق إنها بتشتغل مرة واحدة على الأقل قبل ما تفكر في الشرط.
مفيدة لما تبقى عايز الكود يتنفذ مرة على الأقل حتى لو الشرط مش صح.

رابع حاجة foreach loop:
لما يبقى عندك مجموعة عناصر زي Array أو List وعايز تعدّي عليهم كلهم بسهولة من غير ما تحسب index.

الLoops مهمة جدًا لأنها بتخلي الكود أقصر وأنضف، بتخلي البرنامج أسرع وأسهل، وبتخلي التعامل مع البيانات الكبيرة حاجة بسيطة. موجودة في كل مكان من Arrays وLists لعمليات على الواجهة أو الـ Backend.

خد بالك من الـ Infinite Loop، لو نسيت تعدّل المتغير اللي بيأثر على الشرط، الكمبيوتر هيشتغل على طول وممكن يcrash. فخليك دايمًا متأكد إن الشرط بيتغير كل مرة.

لو عايز تعرف اكتر عن الloops الموقع ده هيفيدك جدا
https://lnkd.in/dnSF47T5

CSharp #Programming #CodingTips #Loops #ForLoop #WhileLoop#
#DoWhileLoop #ForeachLoop #LearnProgramming #CodeSmart #CleanCode
#DeveloperLife #ProgrammingBasics #SoftwareEngineering #Automation
#CodeEfficiently

Show translation

2- The code in the Github Respository

3-Question: What happens if the user enters a value outside the range of 1 to 7?

Answer: If the user enters a value outside the range of 1 to 7, `Enum.Parse` will throw a `System.ArgumentException` because the input does not match any defined value in the enum. This means the program will crash unless the input is properly validated or handled using a `try-catch` block. Proper input validation ensures that only valid integers corresponding to enum values are processed