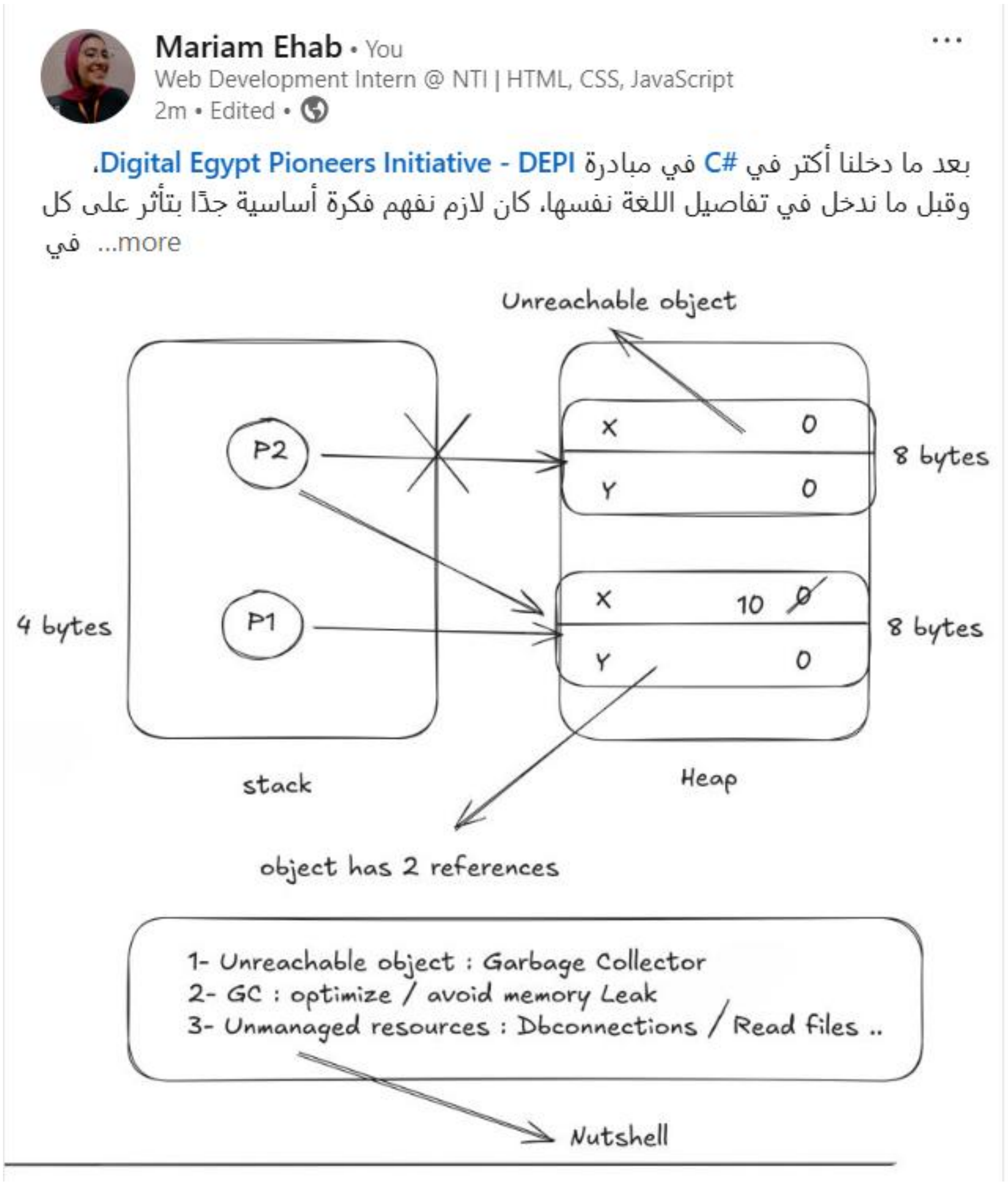
1) LinkedIn article about variables allocation in stack and heap for both value and ref types.



Mariam Ehab • You
Web Development Intern @ NTI | HTML, CSS, JavaScript
2m • Edited •

بعد ما دخلنا أكتر في C# في مبادرة Digital Egypt Pioneers Initiative - DEPI،
وقبل ما ندخل في تفاصيل اللغة نفسها، كان لازم نفهم فكرة أساسية جدًا بتأثر على كل في ...more

1- Unreachable object : Garbage Collector
2- GC : optimize / avoid memory Leak
3- Unmanaged resources : Dbconnections / Read files ..

2) What’s the difference between compiled and interpreted languages and in this way what about CSharp?

Compiled Languages

The source code is translated **once** into machine code, so the execution is fast.
The error are detected at the compile time.

Languages are like: C, C++, Go

Interpreted Languages

The code is executed **line by line** at runtime so that its slower in execution.
Errors appear during execution.

Languages like: Python, JavaScript, PHP

What About C#?

C# is a hybrid language because:

1. C# code is compiled into Intermediate Language (IL).
2. IL runs on the Common Language Runtime (CLR).
3. At runtime, the JIT Compiler converts IL into machine code.

So C# is Compiled to IL, then Just-In-Time compiled at runtime.

---

3) Implicit, Explicit, Convert, and Parse Casting

Implicit Casting: Happens automatically. Safe, so no data loss and we don’t need to use the check block.

Like:

```
int x = 5;
double y = x;
```

Explicit Casting: this must be written manually, and it risks data loss, so we put it on the checked block.

Like:

```
double x = 5.7;
int y = (int) x; // y = 5
```

Convert: Uses Convert class to handle null values safely. It may throw exceptions if conversion fails.

Like:

```
string s = "123";
int x = Convert.ToInt32(s);
```

Parse: Converts string to the value type and throws exception if string is invalid or null.

```
int x = int.Parse("123");
```
