

# Pointers - part 1



# Unsafe keyword

unsafe

```
{  
    // code using pointers  
}
```

```
public unsafe void Method (int i)  
{  
    // code using pointers  
}
```



# Pointer declaration

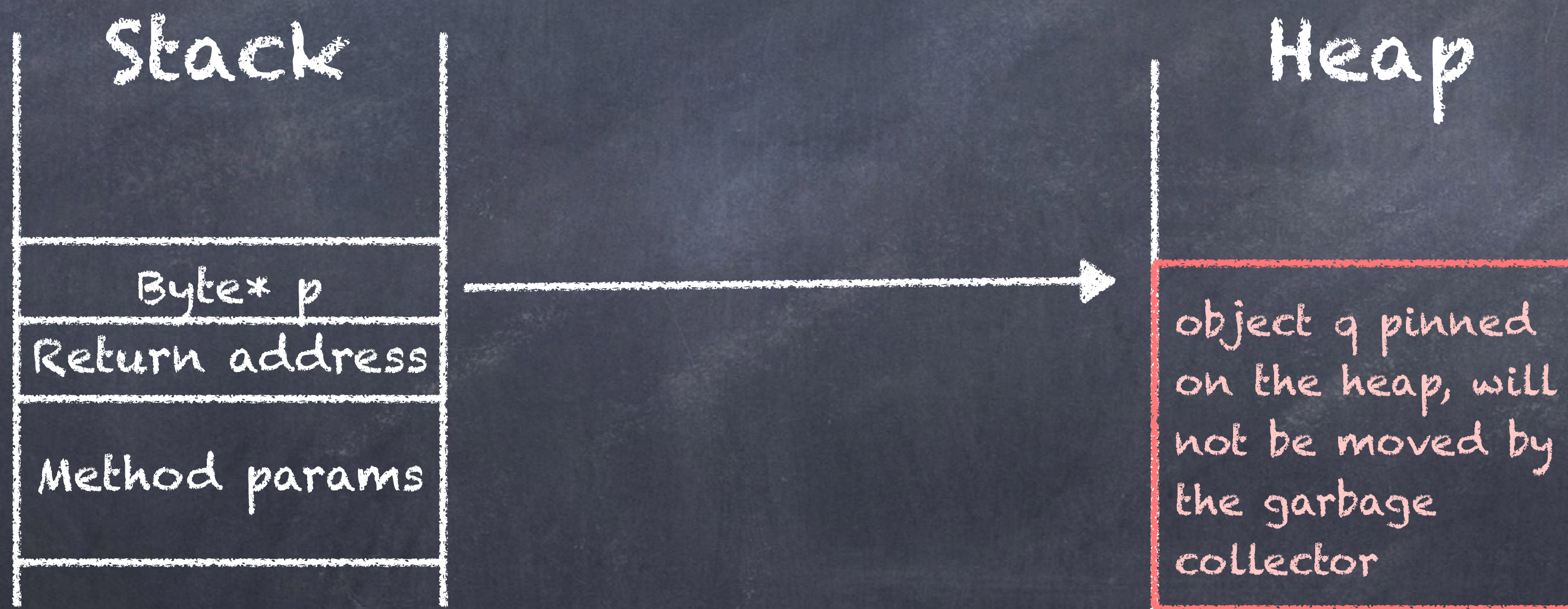
Byte\* p = null;

Generic form: Type\*



# Fixed keyword

**fixed** (Byte\* p = &q) { ... }





# Pointer operations

Byte b = \*p;

Byte b = p[2];

p = p + 3;



# Pointers in C#

- To aid interoperability with unmanaged code
- To support a C or C++ .NET compiler
- To make it easy to port pointer-based algorithms to C#



# Main Takeaway

Always use pointers for image manipulation in .NET



# Guidelines

- Use **high-level** methods to access data in unmanaged memory, to avoid **unsafe** code
- If you need to read or write large blocks of unmanaged data directly, then obtain a **pointer** to the data
- For image manipulation **always use pointers**, because the image data will be too large for any kind of high-level access