|  |  |
| --- | --- |
| Boxing and Unboxing |  |

### ****What is Boxing?****

**Boxing** is the process of **converting a value type to an object type** (or any interface type it implements).

* Value types: int, float, bool, struct, etc.
* Reference types: object, string, class, etc.

**Boxing wraps the value inside a reference type (object)** and stores it on the heap.

int num = 10;

object obj = num; // Boxing

### ****What is Unboxing?****

**Unboxing** is the process of **extracting the value type from an object**.

You must explicitly cast the object back to the value type.

object obj = 10;

int num = (int)obj; // Unboxing

### ****Boxing vs Unboxing****

| **Feature** | **Boxing** | **Unboxing** |
| --- | --- | --- |
| Direction | Value type ➝ Object | Object ➝ Value type |
| Explicit? | Implicit | **Explicit cast required** |
| Memory | Value copied to heap | Value copied from heap to stack |
| Performance | Slower than direct value types | Slower due to casting & heap access |

Code

|  |
| --- |
| class Program  {  static void Main()  {  int x = 42; // Value type  object obj = x; // Boxing  int y = (int)obj; // Unboxing  Console.WriteLine($"Boxed value: {obj}");  Console.WriteLine($"Unboxed value: {y}");  }  } |

Diagram: Boxing and Unboxing

|  |
| --- |
| ┌────────────┐ Boxing ┌─────────────┐  │ Stack │ ───────────────▶ │ Heap │  │ int x=10 │ │ object obj │  └────────────┘ ◀────────── └─────────────┘  Unboxing |

 In **Boxing**, the int value is moved from the stack to the heap.

 In **Unboxing**, it is extracted from the heap and returned to the stack.

 **Boxing creates a new object** on the heap.

 **Unboxing requires a cast** — if the object isn’t the correct type, it throws an InvalidCastException.

 **Avoid unnecessary boxing/unboxing** in performance-critical code.

When Does It Happen Implicitly?

|  |
| --- |
| ArrayList list = new ArrayList();  list.Add(10); // Boxing happens here (int to object)  int num = (int)list[0]; // Unboxing here |

### Summary

| **Term** | **Meaning** |
| --- | --- |
| Boxing | Value → Object (implicit) |
| Unboxing | Object → Value (explicit) |
| Memory | Heap allocation for boxed value |
| Performance | Slower due to conversion & GC overhead |