

## Stack and Heap Binding Examples

### Example 1: Simple Value Type Binding (Stack)

Value types (like int, float, char) are stored on the stack. When you create a value type variable, it is bound directly to the stack.

STACK	HEAP
a = 10	
b = 20	

```
public class Program
{
    public static void Main()
    {
        int a = 10;

        int b = a; // b is a separate copy of a, stored on the stack

        b = 20;

        Console.WriteLine($"a: {a}, b: {b}"); // Outputs: a: 10, b: 20
    }
}
```

## Stack and Heap Binding Examples

### Example 2: Reference Type Binding (Heap)

Reference types (like class, array) are stored on the heap. The variable holds a reference (or address) to the actual data in the heap.

STACK	HEAP
person1	
person2	Person object with Name = Bob

```
public class Program
{
    public class Person
    {
        public string Name { get; set; }
    }

    public static void Main()
    {
        Person person1 = new Person { Name = "Alice" };
        Person person2 = person1; // person2 holds the reference to the same object on the heap
        person2.Name = "Bob";

        Console.WriteLine($"person1.Name: {person1.Name}, person2.Name: {person2.Name}"); //
Outputs: person1.Name: Bob, person2.Name: Bob
    }
}
```

## Stack and Heap Binding Examples

### Example 3: Value Types in Reference Types

When value types are part of reference types (fields in a class), the value types are stored within the heap allocated space of the reference type.

STACK	HEAP
rect	
rect2	Rectangle object with Width = 15, Height = 10

```
public class Program
```

```
{
```

```
    public class Rectangle
```

```
    {
```

```
        public int Width { get; set; }
```

```
        public int Height { get; set; }
```

```
    }
```

```
    public static void Main()
```

```
    {
```

```
        Rectangle rect = new Rectangle { Width = 5, Height = 10 };
```

```
        Rectangle rect2 = rect;
```

```
        rect2.Width = 15;
```

```
        Console.WriteLine($"rect.Width: {rect.Width}, rect2.Width: {rect2.Width}"); // Outputs:
```

```
rect.Width: 15, rect2.Width: 15
```

```
    }
```

```
}
```

## Stack and Heap Binding Examples

### Example 4: Passing Parameters (Stack and Heap Interaction)

When passing parameters, value types are copied (stack) and reference types pass the reference (heap).

STACK	HEAP
x = 10 a = 20 person	
	Person object with Name = Bob

```
public class Program
{
    public static void Main()
    {
        int x = 10;

        ChangeValue(x); // x is passed by value, original x is not changed

        Console.WriteLine($"x: {x}"); // Outputs: x: 10

        Person person = new Person { Name = "Alice" };

        ChangeName(person); // person is passed by reference, original object is changed

        Console.WriteLine($"person.Name: {person.Name}"); // Outputs: person.Name: Bob
    }

    public static void ChangeValue(int a)
    {
        a = 20;
```

## Stack and Heap Binding Examples

```
}
```

```
public static void ChangeName(Person p)
```

```
{
```

```
    p.Name = "Bob";
```

```
}
```

```
public class Person
```

```
{
```

```
    public string Name { get; set; }
```

```
}
```

```
}
```