**8 kyu**

**Playing with cubes II**

30780% of 262597 of1,750[Jotha](https://www.codewars.com/users/Jotha" \o "This kata's Sensei)

C#

* [TRAIN AGAIN](https://www.codewars.com/kata/playing-with-cubes-ii/train/csharp)
* [NEXT KATA](https://www.codewars.com/trainer/csharp)

Details

[Solutions](https://www.codewars.com/kata/playing-with-cubes-ii/solutions/csharp)

[Forks (1)](https://www.codewars.com/kata/playing-with-cubes-ii/forks/csharp)

[Discourse (47)](https://www.codewars.com/kata/playing-with-cubes-ii/discuss/csharp)

* Add to Collection
* |
* Share this kata:

Hey Codewarrior!

You already implemented a **Cube** class, but now we need your help again! I'm talking about constructors. We don't have one. Let's code two: One taking an integer and one handling no given arguments!

Also we got a problem with negative values. Correct the code so negative values will be switched to positive ones!

The constructor taking no arguments should assign 0 to Cube's Side property.

<https://www.codewars.com/kata/playing-with-cubes-ii/csharp>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

class Program

{

//public static int[] ArrayPreviousLess(int[] arr)

//{

// //coding and coding..

// int[] res = new int[arr.Length];

// for (int i = arr.Length - 1; i >= 0; i--)

// {

// int j;

// for (j = i - 1; j >= 0; j--)

// {

// if (arr[j] < arr[i])

// {

// res[i] = arr[j];

// break;

// }

// }

// if (res[i] == 0) res[i] = -1;

// }

// return res;

//}

public class Cube

{

private int Side;

public int GetSide()

{

//if (this.Side < 0) return -Side;

return Math.Abs(-this.Side);

}

public void SetSide(int num)

{

this.Side = Math.Abs(num);

}

public Cube()

{

this.Side = 0;

}

public Cube(int side)

{

this.Side = Math.Abs(side);

}

}

static void Main(string[] args)

{

//int[] arr = new int[] { 3, 5, 2, 4, 5 };

//int[] res = ArrayPreviousLess(arr);

// //{ -1, 3, -1, 2, 4}

//foreach (int item in res)

//{

// Console.Write(item + " ");

//}

//Console.WriteLine(AreEquallyStrong(10, 15, 15, 10));

Cube cube = new Cube(56);

Console.WriteLine(cube.GetSide());

Console.ReadLine();

}

}

}