

Solution Review: Calling a Constructor from a Constructor

This review provides insight to the solution of the 'Calling a Constructor from a Constructor' challenge.

WE'LL COVER THE FOLLOWING ^

- Solution
 - Explanation

Solution

```
class VendingMachine {  
  
    // Private Fields  
    private string _count;  
    private string _capacity;  
    private string _moneyCollected;  
  
    // Default Constructor  
    public VendingMachine() {  
  
        this._count = "";  
        this._capacity = "";  
        this._moneyCollected = "";  
    }  
  
    // Parameterized Constructor 1  
    public VendingMachine(string count, string capacity) {  
  
        this._count = count;  
        this._capacity = capacity;  
    }  
  
    // Parameterized Constructor 2  
    public VendingMachine(string count, string capacity, string money)  
    : this(count, capacity)  
    {  
        this._moneyCollected = money;  
    }  
  
    // Method to return vending machine details  
    public string GetDetails() {  
  
        return this._count + ". " + this._capacity + ". " + this._moneyCollected;
```



```

    return this._count + ", " + this._capacity + ", " + this._moneyCollected;
}

}

class Demo {

    public static void Main(string[] args) {
        VendingMachine vendy = new VendingMachine("Count: 45", "Capacity: 50", "Money Collected:");
        Console.WriteLine(vendy.GetDetails());
    }

}

```



Explanation

The solution is pretty straightforward.

- **Line 25:** Parameterized constructor with two parameters is called using `this()`.
- **Line 27:** Third parameter is assigned to the respective field inside the constructor.