The 'this' Reference Variable

In this lesson, you will learn about the 'this' reference variable.

WE'LL COVER THE FOLLOWING

The `this` Reference Variable

The `this` reference variable exists for every class. It refers to the current instance of a class. The `this.memberName` specifies that we are accessing the `memberName` of the current object.

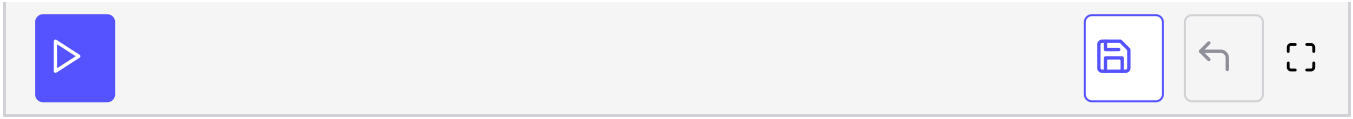
Accessing a Field

We can use the `this` when we have a method argument which has the same name as a **field**. It's always a good convention for the beginners to use the `this` keyword in their class implementation when initializing or accessing the fields. This will help us avoid any confusion or errors.

Let's see it in action:

```
class VendingMachine {

  private int moneyCollected = 70;

  // A simple print function
  public void PrintMoney(int moneyCollected){
    Console.WriteLine("Money Collected using this variable: " + this.moneyCollected);
    Console.WriteLine("Money Collected without using this variable: " + moneyCollected);
  }

}

class Demo {

  public static void Main(string[] args) {
    //passing the parameters
```

```
    var vendingMachine = new VendingMachine(); // Object created with parameterized construct
    vendingMachine.PrintMoney(-10);
  }


}
```



In the above code, we have used the `this` keyword on **line** 7. The purpose of using `this` here is to differentiate between the arguments being passed to the method and the fields of the class. For example, `this.moneyCollected` means we are referring to the field of the class while simply using `moneyCollected` means that we are referring to the argument being passed to the method.

At this point, we know all about the fields and methods of a class. In the next lesson, we will discover an efficient way of declaring fields and how to manipulate these fields using specific methods.