

In this example, I assume that `revenue` and `netProfit` are instances of `Millions`. But isn't `IncomeStatement` an Eloquent model? Glad you asked. It is. And we can write a custom cast to convert floats to `Millions`:

```
class MillionsCast implements CastsAttributes
{
    /**
     * @param float $value
     */
    public function get($model, $key, $value, $attributes)
    {
        return new Millions($value);
    }

    /**
     * @param Millions $millions
     */
    public function set($model, $key, $millions, $attributes) {
        return [
            $key => $millions->value,
        ];
    }
}
```

It can be used in an Eloquent model, and here's how it works:

- When you're accessing an attribute on the model, the `get` method will be called. So `$incomeStatement->revenue` will return an instance of `Millions`.
- When you're setting an attribute on the model, the `set` method will be called. So `$incomeStatement->revenue = new Millions(1000)` will insert the `value` property (1000) from the `Millions` instance.