**Name**: Strategy

**Category**: Behavioral

**Description**: Strategy is a Software engineering pattern that seeks to enable end users with the power of choice. We interact with this in our everyday lives constantly. When you purchase something on Amazon you are presented with multiple options: Credit Card, Debit Card, Gift Card, and services such as PayPal or Cash App. Each option is a valid method by which you might complete the transaction. Though the information may differ, by completing all the steps of purchase and confirming your order you will receive the package in the mail. You were presented with a wide array of options (inputs) but regardless of which method is used you will receive the same output. For another example, consider when you need to drive to someplace you have never been. You plug the location into Google Maps and before you even begin your drive you are given 3 different routes as options. Different routes will prioritize different things, things such as speed, traffic, miles, and even scenery. Different choices (routes) heed the same result (destination).

**When to Use**: This method should be employed to broaden the options for an end user, in the case of Amazon it was used to allow for a greater breadth of users to engage with the platform. Nobody will have every single payment method, however, by employing Strategy, they can hopefully provide at least one payment method that each user should have. It should be used in cases when end users do not have the exact same specific desires, limitations, or resources in accordance with the software.

**Advantages**:

* Expands size of potential userbase
* Accounts for the uniqueness of each customer

**Disadvantages**:

* If taken to its extreme it could drastically slow down new feature implementation
* Paradox of choice, you may needlessly overwhelm an end user