### ****Solution Structure****

#### Basket.php

class Basket

{

private $products;

private $deliveryRules;

private $offers;

private $items = [];

public function \_\_construct($products, $deliveryRules, $offers = [])

{

$this->products = $products;

$this->deliveryRules = $deliveryRules;

$this->offers = $offers;

}

public function add($productCode)

{

if (!isset($this->products[$productCode])) {

throw new Exception("Product code {$productCode} not found.");

}

if (!isset($this->items[$productCode])) {

$this->items[$productCode] = 0;

}

$this->items[$productCode]++;

}

public function total()

{

$subtotal = 0;

foreach ($this->items as $code => $qty) {

$price = $this->products[$code];

if ($code === 'R01' && isset($this->offers['R01'])) {

$offerQty = floor($qty / 2);

$fullQty = $qty - $offerQty;

$subtotal += ($fullQty \* $price) + ($offerQty \* ($price / 2));

} else {

$subtotal += $qty \* $price;

}

}

// Apply delivery

$delivery = 0;

foreach ($this->deliveryRules as $limit => $cost) {

if ($subtotal < $limit) {

$delivery = $cost;

break;

}

}

return number\_format($subtotal + $delivery, 2);

}

}

### ****Example Usage: I have index.php File:****

require\_once 'Basket.php';

$catalog = [

'R01' => 32.95,

'G01' => 24.95,

'B01' => 7.95

];

$deliveryRules = [

50 => 4.95,

90 => 2.95,

PHP\_INT\_MAX => 0.00 // Free delivery over $90

];

$offers = [

'R01' => 'BOGO\_HALF' // Buy one get one half

];

$basket = new Basket($catalog, $deliveryRules, $offers);

$basket->add('B01');

$basket->add('B01');

$basket->add('R01');

$basket->add('R01');

$basket->add('R01');

echo "Total: $" . $basket->total(); // Expected: 98.27

### README.md

# Acme Widget Co - Shopping Basket POC

This is a PHP proof-of-concept basket system for Acme Widget Co.

## Features

- Supports product catalog, delivery rules, and promotional offers

- Dynamic calculation of total including discounts and delivery

- Easily extendable

## How to Use

1. Clone the repo

2. Include `Basket.php` in your project

3. Initialize it with a product list, delivery rules, and offers

4. Use `add()` to add items, and `total()` to get final price

## Example

```php

$basket = new Basket($products, $deliveryRules, $offers);

$basket->add('R01');

$basket->add('R01');

echo $basket->total(); // $54.37