

1-)Use the Module pattern to create a shopping cart that has one private immutable member: `basket[]`, and the following public methods:

- `upsertItem()` add a product to basket if doesn't exist, or update if exist.
- `getItemsCount()` returns back the number of products in the basket.
- `getTotalPrice()` calculates the total price, taking in consideration the count of products.
- `removeItemById(id)` removes a product from the basket.

Every product item has the following structure:

```
Product = {id: 0, name: 'Coffee', description: 'Coffee Grounds from Ethiopia', price: 9.5, count: 1}
```

Note: One instance of this module should be allowed per application.

2-) Write an implementation for the Observer Pattern where observers have the following format: `{'event': [observers]}`

For example:

```
{
  'eat': [function1, function2],
  'study': [function3, function4, function5]
}
```

This **Observable/Subject** should be used as following:

```
.on('eat', function1) // register an observer
.on('eat', function2) // register an observer
.emit('eat') // all observers (function1, and function2) should be invoked
```

<https://classroom.github.com/a/uu-zG0xu>