Java

```java
class ProductOfNumbers {

    public ProductOfNumbers() {

    }

    public void add(int num) {

    }

    public int getProduct(int k) {

    }
}

/**
 * Your ProductOfNumbers object will be instantiated and called as such:
 * ProductOfNumbers obj = new ProductOfNumbers();
 * obj.add(num);
 * int param_2 = obj.getProduct(k);
 */
```

JavaScript

```javascript
var ProductOfNumbers = function() {

};

/**
```

```
 * @param {number} num
 * @return {void}
 */
ProductOfNumbers.prototype.add = function(num) {

};

/**
 * @param {number} k
 * @return {number}
 */
ProductOfNumbers.prototype.getProduct = function(k) {

};

/**
 * Your ProductOfNumbers object will be instantiated and called as such:
 * var obj = new ProductOfNumbers()
 * obj.add(num)
 * var param_2 = obj.getProduct(k)
 */
```

TypeScript

```
class ProductOfNumbers {
    constructor() {

    }

    add(num: number): void {
```

```
    }

    getProduct(k: number): number {

    }
}

/**
 * Your ProductOfNumbers object will be instantiated and called as such:
 * var obj = new ProductOfNumbers()
 * obj.add(num)
 * var param_2 = obj.getProduct(k)
 */
```

C++

```
class ProductOfNumbers {
public:
    ProductOfNumbers() {

    }

    void add(int num) {

    }

    int getProduct(int k) {

    }
};
```

```
/**
 * Your ProductOfNumbers object will be instantiated and called as such:
 * ProductOfNumbers* obj = new ProductOfNumbers();
 * obj->add(num);
 * int param_2 = obj->getProduct(k);
 */
```

C#

```
public class ProductOfNumbers {

    public ProductOfNumbers() {

    }

    public void Add(int num) {

    }

    public int GetProduct(int k) {

    }
}

/**
 * Your ProductOfNumbers object will be instantiated and called as such:
 * ProductOfNumbers obj = new ProductOfNumbers();
 * obj.Add(num);
 * int param_2 = obj.GetProduct(k);
 */
```

Kotlin

```kotlin
class ProductOfNumbers() {

    fun add(num: Int) {

    }

    fun getProduct(k: Int): Int {

    }

}

/**
 * Your ProductOfNumbers object will be instantiated and called as such:
 * var obj = ProductOfNumbers()
 * obj.add(num)
 * var param_2 = obj.getProduct(k)
 */
```

Go

```go
type ProductOfNumbers struct {

}


func Constructor() ProductOfNumbers {
```

```
}


func (this *ProductOfNumbers) Add(num int)  {

}


func (this *ProductOfNumbers) GetProduct(k int) int {

}


/**
 * Your ProductOfNumbers object will be instantiated and called as such:
 * obj := Constructor();
 * obj.Add(num);
 * param_2 := obj.GetProduct(k);
 */
```
