Java

```java
class DinnerPlates {

    public DinnerPlates(int capacity) {

    }

    public void push(int val) {

    }

    public int pop() {

    }

    public int popAtStack(int index) {

    }
}

/**
 * Your DinnerPlates object will be instantiated and called as such:
 * DinnerPlates obj = new DinnerPlates(capacity);
 * obj.push(val);
 * int param_2 = obj.pop();
 * int param_3 = obj.popAtStack(index);
 */
```

JavaScript

```javascript
/**
 * @param {number} capacity
 */
var DinnerPlates = function(capacity) {

};

/**
 * @param {number} val
 * @return {void}
 */
DinnerPlates.prototype.push = function(val) {

};

/**
 * @return {number}
 */
DinnerPlates.prototype.pop = function() {

};

/**
 * @param {number} index
 * @return {number}
 */
DinnerPlates.prototype.popAtStack = function(index) {

};

/**
 * Your DinnerPlates object will be instantiated and called as such:
```

```
 * var obj = new DinnerPlates(capacity)
 * obj.push(val)
 * var param_2 = obj.pop()
 * var param_3 = obj.popAtStack(index)
 */
```

TypeScript

```
class DinnerPlates {
    constructor(capacity: number) {

    }

    push(val: number): void {

    }

    pop(): number {

    }

    popAtStack(index: number): number {

    }
}

/**
 * Your DinnerPlates object will be instantiated and called as such:
 * var obj = new DinnerPlates(capacity)
 * obj.push(val)
 * var param_2 = obj.pop()
```

```
 * var param_3 = obj.popAtStack(index)
 */
```

C++

```
class DinnerPlates {
public:
    DinnerPlates(int capacity) {

    }

    void push(int val) {

    }

    int pop() {

    }

    int popAtStack(int index) {

    }
};

/**
 * Your DinnerPlates object will be instantiated and called as such:
 * DinnerPlates* obj = new DinnerPlates(capacity);
 * obj->push(val);
 * int param_2 = obj->pop();
 * int param_3 = obj->popAtStack(index);
 */
```

C#

```
public class DinnerPlates {

    public DinnerPlates(int capacity) {

    }

    public void Push(int val) {

    }

    public int Pop() {

    }

    public int PopAtStack(int index) {

    }
}

/**
 * Your DinnerPlates object will be instantiated and called as such:
 * DinnerPlates obj = new DinnerPlates(capacity);
 * obj.Push(val);
 * int param_2 = obj.Pop();
 * int param_3 = obj.PopAtStack(index);
 */
```

Kotlin

```kotlin
class DinnerPlates(capacity: Int) {

    fun push(`val`: Int) {

    }

    fun pop(): Int {

    }

    fun popAtStack(index: Int): Int {

    }

}

/**
 * Your DinnerPlates object will be instantiated and called as such:
 * var obj = DinnerPlates(capacity)
 * obj.push(`val`)
 * var param_2 = obj.pop()
 * var param_3 = obj.popAtStack(index)
 */
```

Go

```go
type DinnerPlates struct {

}
```

```go
func Constructor(capacity int) DinnerPlates {

}


func (this *DinnerPlates) Push(val int)  {

}


func (this *DinnerPlates) Pop() int {

}


func (this *DinnerPlates) PopAtStack(index int) int {

}


/**
 * Your DinnerPlates object will be instantiated and called as such:
 * obj := Constructor(capacity);
 * obj.Push(val);
 * param_2 := obj.Pop();
 * param_3 := obj.PopAtStack(index);
 */
```
