

Array

즐겁고 알찬 자료구조 튜터링

Array

Array

- 동일한 타입의 데이터를 저장하는 **순서 리스트**
- **index**를 사용하여 원소에 접근
- 메모리에 **연속적인 공간으로 할당**
- **배열의 크기**는 생성시기에 정해짐

Memory Location									
200	201	202	203	204	205	206			
U	B	F	D	A	E	C			
0	1	2	3	4	5	6			
Index									

RealArray Class



```
1 class RealArray {
2 private:
3     int *arr;
4
5 public:
6     RealArray(int capacity) {
7         arr = new int[capacity];
8     }
9     int &operator[](int index) {
10         return *(arr + index);
11     }
12     ~RealArray() {
13         delete[] arr;
14     }
15 };
```

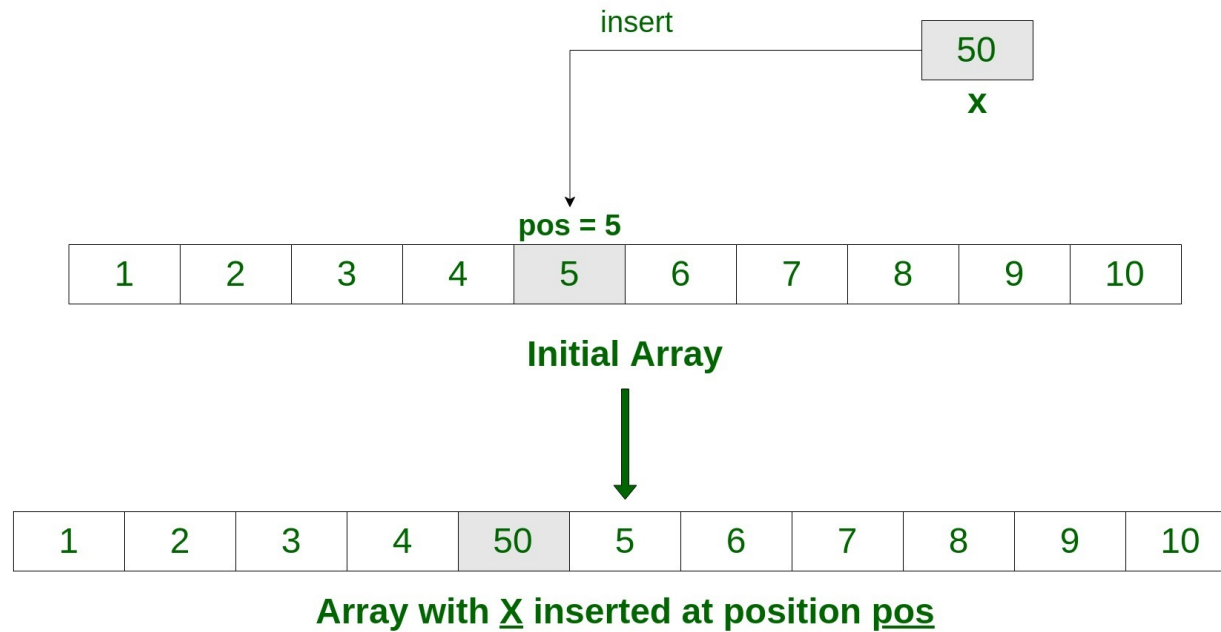
ArrayObject Class



```
1  class ArrayObject : public RealArray {
2  private:
3      int capa;
4      int cur;
5
6  public:
7      ArrayObject(int capacity) : RealArray(capacity) {
8          capa = capacity;
9          cur = 0;
10     }
11
```

배열에서 삽입 연산

Insert an element at a specific position in an Array.



배열에서 삽입 연산



```
1 void insert(int index, int data) {  
2     for (int i = capa - 1; i > index; i--) {  
3         (*this)[i] = (*this)[i - 1];  
4     }  
5     (*this)[index] = data;  
6  
7     if (cur < capa) {  
8         cur++;  
9     }  
10 }
```

배열에서 삭제 연산



```
1 void erase(int index) {  
2     for (int i = index; i < capa - 1; i++) {  
3         (*this)[i] = (*this)[i + 1];  
4     }  
5     cur--;  
6 }
```


배열의 장단점

- 장점
 - 원소에 빠른 접근이 가능
 - 구현이 쉬움
- 단점
 - 크기가 정해져 있음
 - 삽입 및 삭제 연산에서 오버헤드가 발생