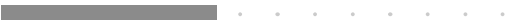


⋮

vector Revisited

- **A container object**
- **Operations (methods)**
 - Report current size
 - Erase all items (empty list)
 - Add new item at the end of the list
 - Access an item by position
 - (More we haven't discussed)



Handwritten notes area with horizontal lines

⋮

vector Member Functions

```
// Example vector usage:
vector<int> vec;
int x = 37;
unsigned int sz = vec.size();
vec.clear();
vec.push_back(x);
vec.push_back(10);
int y = vec[0] + 2;
vec[1] = vec[0] - vec[1];
```



Handwritten notes area with horizontal lines

⋮

Can We Build a vector?

- **vector is a container**
 - Like C++ array, only more
- **How to design a simple one**
 - Only one vector in program
 - Element type must be int
 - No sort() or other fancy stuff

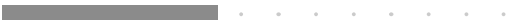


Handwritten notes area with horizontal lines

⋮

Simple Vector Implementation

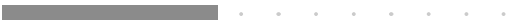
```
// For a vector of 'int' items
const unsigned int max_el = 100;
unsigned int num_el = 0;
int vec [max_el];
// Operations:
unsigned int vec_size();
void vec_clear();
void vec_push_back (int elem);
// Use C++ [] to access elements
```



⋮

Reporting Vector Size

```
unsigned int vec_size()
// Returns: number of elements
// currently in vector.
{
    return num_el;
}
```



⋮

Erasing All Elements

```
void vec_clear()
// Returns: nothing
// Erases all elements from
// vector
{
    num_el = 0;
}
```

Do the element values
really disappear?



⋮

Adding An Element

```

void vec_push_back(int elem)
// Arguments:
//   elem - Element to be added
// Returns: nothing
// If no more room, does nothing
{
  if (num_el < max_el)
  {
    vec[num_el] = elem;
    num_el++;
  }
}

```

Is this correct?

.....

⋮

Simple Vector Example

```

// Similar to real "vector"
int x = 37;
unsigned int sz = vec_size();
vec_clear();
vec_push_back(x);
vec_push_back(10);
int y = vec[0] + 2;
vec[1] = vec[0] - vec[1];

```

.....

⋮

Remove a Single Element?

- **Have a simple vector**
- **Want to remove an element**
 - At the end (back)
 - At the beginning (front)
 - In the middle
- **Can we do it?**

.....

Removing the Last Element

5	2	-3	7	10	??	??	??
0	1	2	3	4	5	6	7

num_el: To "remove" last element, decrement element count

max_el: Last element value still present, but ignored

Removing Middle Element

5	2	-3	7	10	??	??	??
0	1	2	3	4	5	6	7

num_el: To "remove" middle element:

max_el: 1) Copy down values
2) Decrement count

Remove element 2

Removing An Element

```
void vec_remove(unsigned int idx)
// idx - Index of elem to remove
{
  unsigned int j;
  if (idx < num_el)
  {
    for (j=idx; j<num_el-1; j++)
      vec[j] = vec[j+1];
    num_el--;
  }
}
Will this work?
```

⋮

Insert a Single Element?

- **Can add element at end**
 - “Push back” operation
- **What about inserting**
 - At beginning (front)?
 - In the middle?
- **How would you do it?**

