What information is needed to represent a vector?

- size (type: size_t)
- capacity (type: size_t)
- array for elements (call this int* data)

Class invariant

① data points to an array of capacity elements.

② data[0], data[1], ... data[size-1]
store the contents of the vector.

We will make sure that ①, ② hold after any of our constructors are called. Then every member function should assume ①, ② upon being called, and must ensure they are still true afterward.

Destructors

opposite of constructor: called when a variable goes out of scope, or is explicitly freed via delete.

E.g.:

```
void f() {
    vector<int> v; // constructor called
    v.push_back(10);
    :
} // destructor for v will be called here.
```

Ex. 2: (kind of contrived)

```
vector<int> * p = new vector<int>; // const.
    :
delete p; // destructor called for (*p).
```

Main job of destructor: free dynamically allocated memory.