```cpp
#include "IntVector.h"
#include <stdexcept>

using namespace std;

IntVector::IntVector(const int aArrayOfIntegers[], size_t aNumberOfElements)
{
    fNumberOfElements = aNumberOfElements;      // passing in the number of elements
    fElements = new int[fNumberOfElements];     //creating an integer array fElements

    for (size_t i = 0; i < fNumberOfElements; i++)
    {
        fElements[i] = aArrayOfIntegers[i];     //looping through and populating the array with values
    }
}

IntVector::~IntVector()
{
    delete[] fElements;         //destructor deletes the elements and array
}

size_t IntVector::size() const
{
    return fNumberOfElements;       //gives back the size of the vector
}

const int IntVector::get(size_t aIndex) const
{
        if (aIndex >= fNumberOfElements)
        {
            throw out_of_range("Illegal vector indices.");
        }
        return  (*this)[aIndex];
}

void IntVector::swap(size_t aSourceIndex, size_t aTargetIndex) //the member function swap() takes two indices and, if they are within range, swaps the
                                                               //
                    corresponding array elements in an IntVector object. We need swap() for sorting.
{
    int temp = get(aSourceIndex);
    fElements[aSourceIndex] = get(aTargetIndex);
    fElements[aTargetIndex] = temp;


```

```
44      if (aTargetIndex > fNumberOfElements)
45      {
46          throw out_of_range("Out of range");
47      }
48  }
49
50  const int IntVector::operator[](size_t aIndex) const
51  {
52          if (aIndex >= 0 && aIndex < fNumberOfElements)
53          {
54              return fElements[aIndex];
55          }
56          throw out_of_range("illegal aIndex.");
57  }
58
```