Figure 12.3 *Arrays* vs. *Vectors*.

Let's review the things that make Vectors different from arrays:

1. A Vector can grow. An array *cannot*. You can determine the size of an array at runtime, but once set, it cannot change:

   ```
   Student[] course =
       new Student[Integer.parseInt(keyb.readLine())];
   ```

   If the user types 5 as the input, then throughout this run of the program, **course** *always* has five elements.

2. A Vector is just another class. An array is a "fundamental part" of the Java language.

3. A Vector can contain a mixture of types. An array can contain only one type of element.

4. Vector elements *must* be objects. An array's element type can be either a class type or a primitive type.

And one last comment on Vector efficiency …

Current implementations of the Vector class can be very inefficient, especially when a Vector object gets long. Implementations in newer versions of Java are getting better.

In the long run, it's probably a good idea to use a Vector instead of an array because the Vector class and its methods are designed to *help you* maintain your data structure, whereas if you use an array you must worry about many of the maintenance details yourself.

A Vector, essentially, is a higher-level $\boxed{}$ than an array.