For Each… Pro's and Con's

For Each

```
int [] nums = { 5, 6, 2, 7, 3, 1 };


for( int x : nums )
{
        System.out.print( x );
}
```

When you see this code… you should say to yourself "for each integer x in the nums array do this…"

It will automatically iterate across the array on the right and put values into the variable on the left, one at a time.

Note – the for each loop eliminates the need for an index variable and the brackets we would normally use to access values.

Why use a For Each loop?

- Much easier to read
- Less chance for error – no loop bounds and low level indices to worry about
- Higher level of abstraction for a common task like iterating over a collection

When can I not use a For Each loop?

- To assign new values to the array.
  - Unless there is a mutator method available for that particular collection that does what you want… you have no way to assign new values to it.
- To compare values at different indices.
  - No index available