AP Computer Science A

1. Log in
2. Unit 5
   - Array Review and Swapping Elements
   - Selection Sort
   - Bubble Sort
   - Insertion Sort
   - Merge Sort
   - Sequential Search
   - Binary Search

Oct 16-7:52 AM

Selection Sort ...

1. Find the smallest element in the array
2. Swap it to the "front" of the array
3. Repeat (skipping swapped smallest element)

Example of Selection Sort

Original Array 5 4 2 3

1st Pass will search for the smallest number in the list ...

Nov 13-12:26 PM

Selection Sort ...

1. Find the smallest element in the array
2. Swap it to the "front" of the array
3. Repeat (skipping swapped smallest element)

Example of Selection Sort

Original Array 5 4 2 3

Once the smallest element is found, swap it to the front ...

Nov 13-12:26 PM

Selection Sort ...

1. Find the smallest element in the array
2. Swap it to the "front" of the array
3. Repeat (skipping swapped smallest element)

Example of Selection Sort

Array after 1st Pass 2 4 5 3

Repeat this process (ignoring the "last items placed")

Nov 13-12:26 PM

Selection Sort ...

1. Find the smallest element in the array
2. Swap it to the "front" of the array
3. Repeat (skipping swapped smallest element)

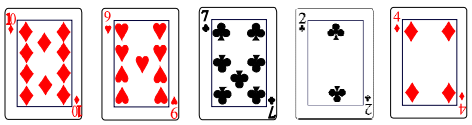
The Complete Process

Original Array	5	4	2	3	// swap 2 to front
After 1st Pass	2	4	5	3	// swap 3 to "front"
After 2nd Pass	2	3	5	4	// swap 4 to "front"
After 3rd Pass	2	3	4	5	// Done!

Nov 13-12:26 PM

Selection Sort Example with Cards ...

* Count the passes
* Notice the last pass?!?



Nov 13-12:26 PM

Things to note with Selection Sort ...

1. With each pass we have that many elements guaranteed to be in order (2 passes = 2 elements) (after ***n*** passes we have ***n*** elements in order)
2. If there are ***n*** elements then we only need ***n-1*** passes
3. After the ***k*** th pass through an array with n elements, we have ***(n-1)-k*** elements to check.

Nov 13-12:26 PM

Now ... to code selection sorting ...

1. Find the smallest number in the array

```
int smallest = theArray[0];                 // 1st element
int smallestIndex = 0;                      // Where is smallest?

for(int i=1 ; i<theArray.length ; i++)      // identify the smallest
{
  if(theArray[i] < smallest){               // if smaller,save it
      smallest = theArray[i];
      smallestIndex = i;
  }
}
```

Nov 13-12:26 PM

Now ... to code selection sorting ...

2. Swap the smallest to the front of the array ...

```
int temp = theArray[0];                     // Store 1st element
theArray[0] = smallest;                     // Overwrite 1st element
theArray[smallestIndex]=temp;               // Move 1st element
```

Nov 13-12:26 PM

But, now we need to run this "find-and-swap" again ... ignoring the 1st element!

This involves another for-loop to track which pass we are on!

```
for(int j=0 ; j<theArray.length ; j++)
{
    int smallest = theArray[ j ];           // 1st element
    int smallestIndex = j;                  // Where is smallest?

    for(int i=j+1 ; i<theArray.length ; i++)    // identify the smallest
    {
      if(theArray[i] < smallest){           // if smaller,save it
          smallest = theArray[i];
          smallestIndex = i;
      }
    }

    int temp = theArray[ j ];               // Swap process
    theArray[ j ] = smallest;
    theArray[smallestIndex]=temp;
}
```

Nov 13-12:26 PM

Here's the final code for a possible selection sort class ...

```
package Unit6;
// author: Hendermonger

public class selectionSort {
    public static void main(String[] args) {

    int[] theArray = {5,4,2,3,1,8,9,6};

    for(int i=0; i<theArray.length ; i++)
        System.out.print(theArray[i]+" ");
    System.out.print(" -> Before Swap");

    for(int j=0 ; j<theArray.length ; j++)
    {
        int smallest = theArray[j];             // 1st element
        int smallestIndex = j;                  // Where is smallest?

            for(int i=j+1 ; i<theArray.length ; i++)    // identify the smallest
            {
                if(theArray[i] < smallest){             // if smaller,save it
                        smallest = theArray[i];
                smallestIndex = i;
            }
            }
        int temp = theArray[j];                 // Swap process
        theArray[j] = smallest;
        theArray[smallestIndex]=temp;
    }
        System.out.println("\n");
        for(int i=0; i<theArray.length ; i++)
            System.out.print(theArray[i]+" ");
        System.out.print(" -> After Swap\n\n");
    }
}
```

Jan 17-8:44 AM

Things to do ...

1. Wrap Up Unit 6 WS 01
2. Work on Unit 6 WS02 Arrays and Selection Sort

Oct 16-9:12 AM