|  |
| --- |
| **Skill 29.1 Exercise 1** |
| If an array of Integer objects contain the following elements, what would the array look like after the third pass of selectionSort, sorting from high to low?  public class Sorter {    private Integer[] a;    public Sorter(Integer[] arr){a = arr;}    /\*\*Swap a[j] and a[j] in array a.\*/  private void swap(int i, int j){  /\*implementation not shown\*/  }    /\*\*  \* Sort array a from largest to smallest using selection sort  \* Precondition: a is an array of Integer objects  \*/  public void selectionSort(){  for(int i = 0; i < a.length-1;i++){  //find max element in a[i+1] to a[n-1]  Integer max = a[i];  int maxPos = i;  for(int j = i + 1; j < a.length;j++){  if(max.compareTo(a[j])<0){//max less than a[j]  max = a[j];  maxPos = j;  }  swap(i, maxPos);  }  }  }    } |
|  |

|  |
| --- |
| **Skill 29.1 Exercise 1** |
| An array of Integer objects is to be sorted from biggest to smallest using the insertionSort method. If the array originally contains,  1 7 9 5 4 12  What will it look like after the thired pass of the for loop? |
| public class InsertionSort {  private Integer[] a;    public void insertionSort(){  for(int i = 1; i < a.length;i++){  Integer temp = a[i];  int j = i - 1;  while(j >= 0 && temp.compareTo(a[j]) > 0){  a[j + 1] = a[j];  j--;  }  a[j + 1] = temp;  }  }  } |
|  |

|  |  |
| --- | --- |
| **Skill 29.3 Exercise 1** | |
| Trace the complete execution of the merge sort algorithm when called on the array below. Show the sub-arrays that are created by the algorithm and show the merging of sub-arrays into larger sorted arrays.  int[] numbers = {8, 5, -9, 14, 0, -1, -7, 3};  mergeSort(numbers); | |
| Sub arrays created | Merging of sub arrays |