| Computer Science II | Lab 18A – HEAP | Lab Value - 100 |
|---|---|---|

**Lab Goal :** This lab was designed to teach you more about a heap.

**Lab Description :** Write a heap program. A heap is a binary tree structure implemented with an array. The root is a spot 0 and the root's children would be a spots 1 and 2. To find any indexes children, you use `i*2+1` and `i*2+2`.

**What does add do?**

Add adds a new item in the very last spot in the tree(array spot length-1). Add then calls swapUp to restructure the tree so that the new item is in the proper location in the heap. swapUp first checks to see if the new item is bigger than its parent. If it is, swapUp swaps the parent and the new item and repeats the same process until the new item is in the root position or it finds that the new item is not larger than its current parent.

**What does remove do?**

Remove copies the root to a variable and then moves the last value in the tree to the root-array spot 0. Remove then calls swapDown to restructure the tree and to check that the tree remains a heap. swapDown looks at the children of the new root and determines which is larger. The larger of the children and the new root are then swapped. This process continues until the bottom of the tree is reached or there are no children larger than the current parent.

## Sample Output :

```
PRINTING THE HEAP!

       75
      17  10
    9  8  2  7
  1  5


PRINTING THE HEAP!

      17
     9  10
   5  8  2  7
 1


PRINTING THE HEAP!

     10
    9  7
  5  8  2  1



PRINTING THE HEAP!

    9
   8  7
 5  1  2
```

**Files Needed ::**

**Heap.java**
**Lab18a.java**

**BONUS :: Write the swapUp() and swapDown() methods recursively.**

```
PRINTING THE HEAP!

   8
  5  7
2  1



PRINTING THE HEAP!

  7
 5  1
2



PRINTING THE HEAP!

 5
2  1



PRINTING THE HEAP!

2
1


PRINTING THE HEAP!

 25
1  2



PRINTING THE HEAP!

  35
 25  2
1


PRINTING THE HEAP!

 25
1  2
```