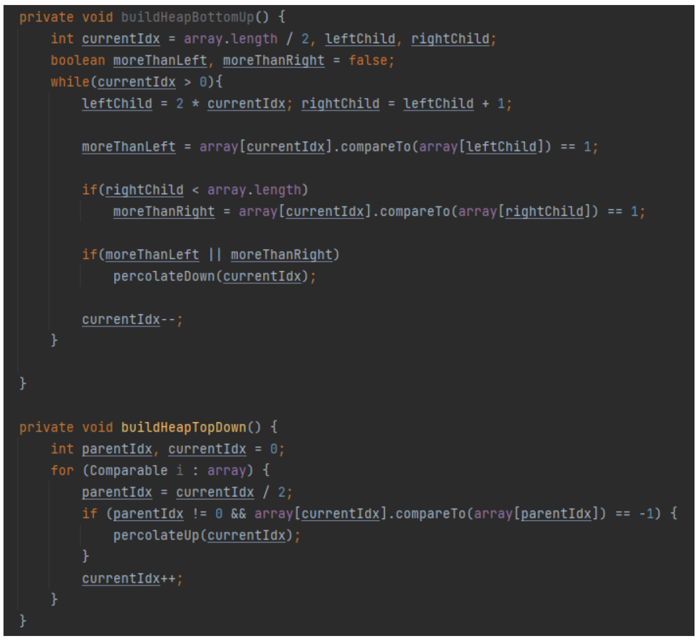
ICS-202 Lab-08 Report

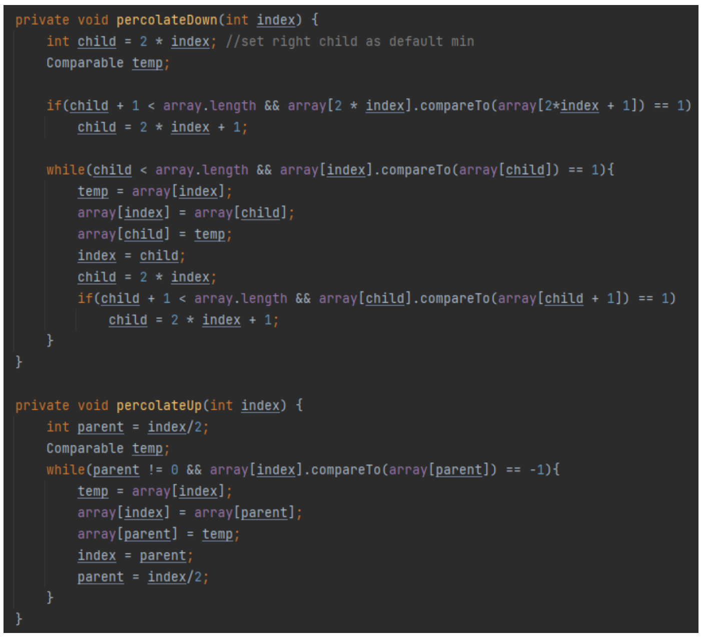
Mohammed Busaleh - 202158210

Exercise 1:

Code:

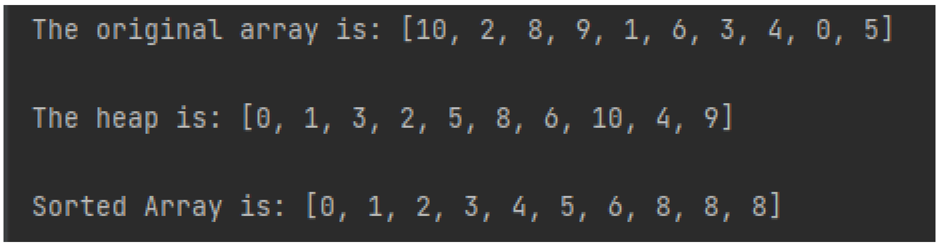


```java
private void buildHeapBottomUp() {
    int currentIdx = array.length / 2, leftChild, rightChild;
    boolean moreThanLeft, moreThanRight = false;
    while(currentIdx > 0){
        leftChild = 2 * currentIdx; rightChild = leftChild + 1;

        moreThanLeft = array[currentIdx].compareTo(array[leftChild]) == 1;

        if(rightChild < array.length)
            moreThanRight = array[currentIdx].compareTo(array[rightChild]) == 1;

        if(moreThanLeft || moreThanRight)
            percolateDown(currentIdx);

        currentIdx--;
    }

}

private void buildHeapTopDown() {
    int parentIdx, currentIdx = 0;
    for (Comparable i : array) {
        parentIdx = currentIdx / 2;
        if (parentIdx != 0 && array[currentIdx].compareTo(array[parentIdx]) == -1) {
            percolateUp(currentIdx);
        }
        currentIdx++;
    }
}
```

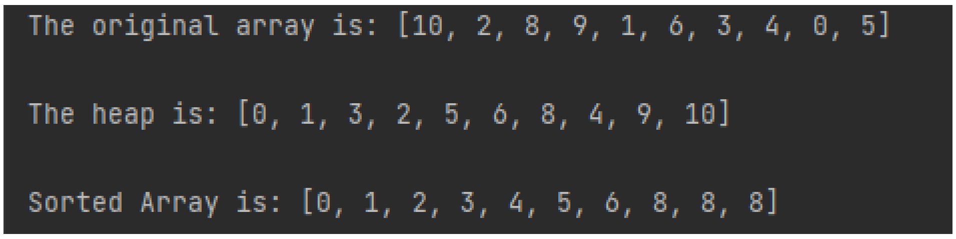


```java
private void percolateDown(int index) {
    int child = 2 * index; //set right child as default min
    Comparable temp;

    if(child + 1 < array.length && array[2 * index].compareTo(array[2*index + 1]) == 1)
        child = 2 * index + 1;

    while(child < array.length && array[index].compareTo(array[child]) == 1){
        temp = array[index];
        array[index] = array[child];
        array[child] = temp;
        index = child;
        child = 2 * index;
        if(child + 1 < array.length && array[child].compareTo(array[child + 1]) == 1)
            child = 2 * index + 1;
    }
}

private void percolateUp(int index) {
    int parent = index/2;
    Comparable temp;
    while(parent != 0 && array[index].compareTo(array[parent]) == -1){
        temp = array[index];
        array[index] = array[parent];
        array[parent] = temp;
        index = parent;
        parent = index/2;
    }
}
```

Output:



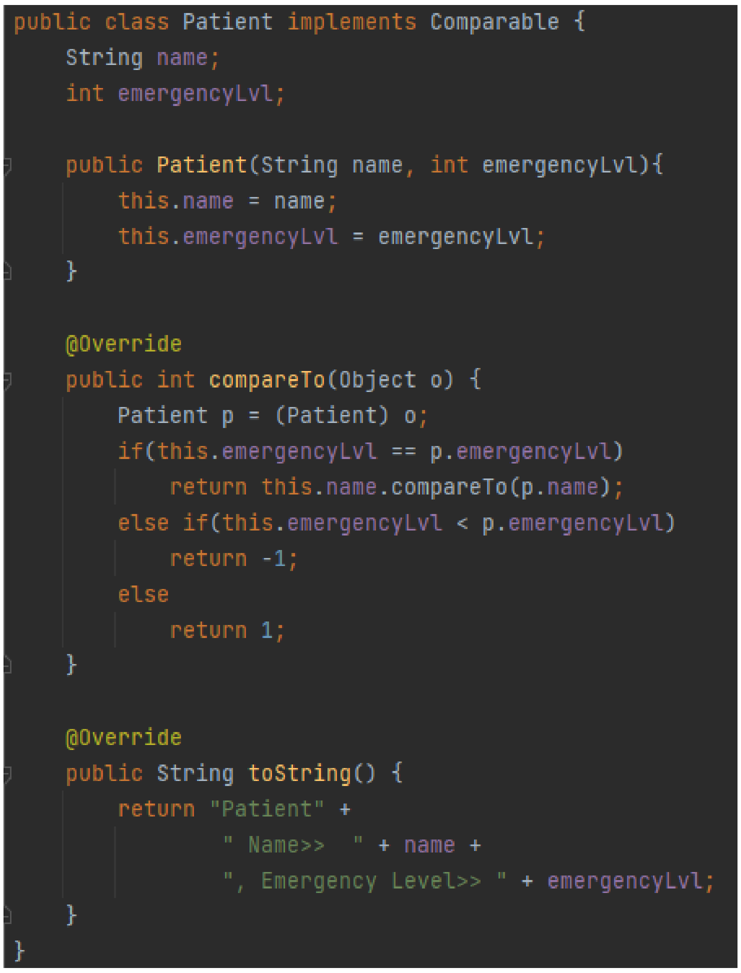
```
The original array is: [10, 2, 8, 9, 1, 6, 3, 4, 0, 5]

The heap is: [0, 1, 3, 2, 5, 8, 6, 10, 4, 9]

Sorted Array is: [0, 1, 2, 3, 4, 5, 6, 8, 8, 8]
```



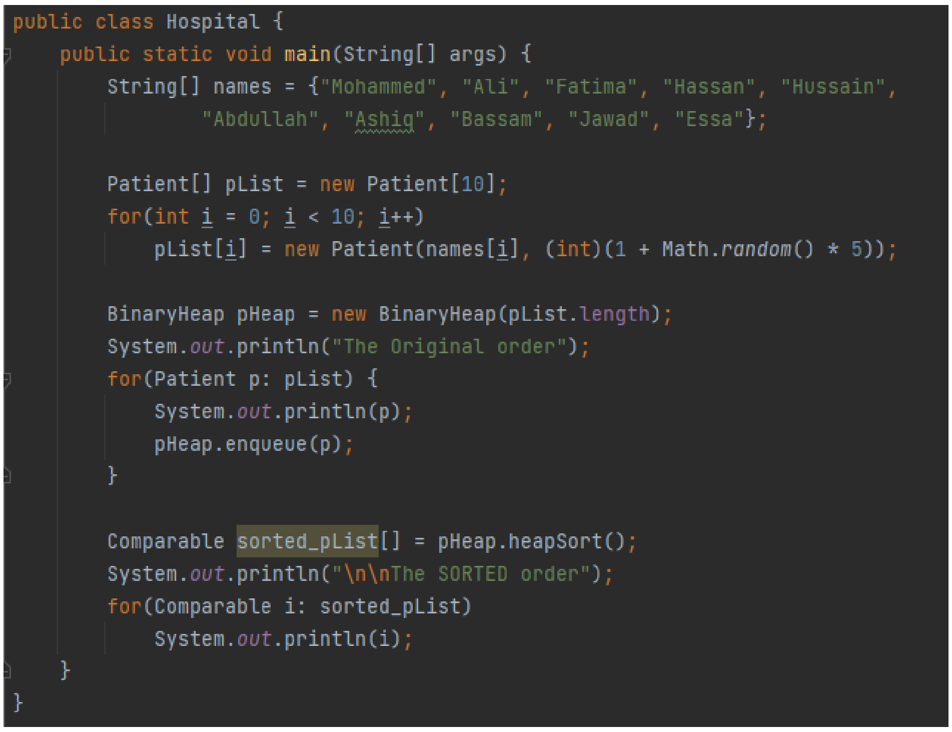
```
The original array is: [10, 2, 8, 9, 1, 6, 3, 4, 0, 5]

The heap is: [0, 1, 3, 2, 5, 6, 8, 4, 9, 10]

Sorted Array is: [0, 1, 2, 3, 4, 5, 6, 8, 8, 8]
```

Exercise 2:

Code:

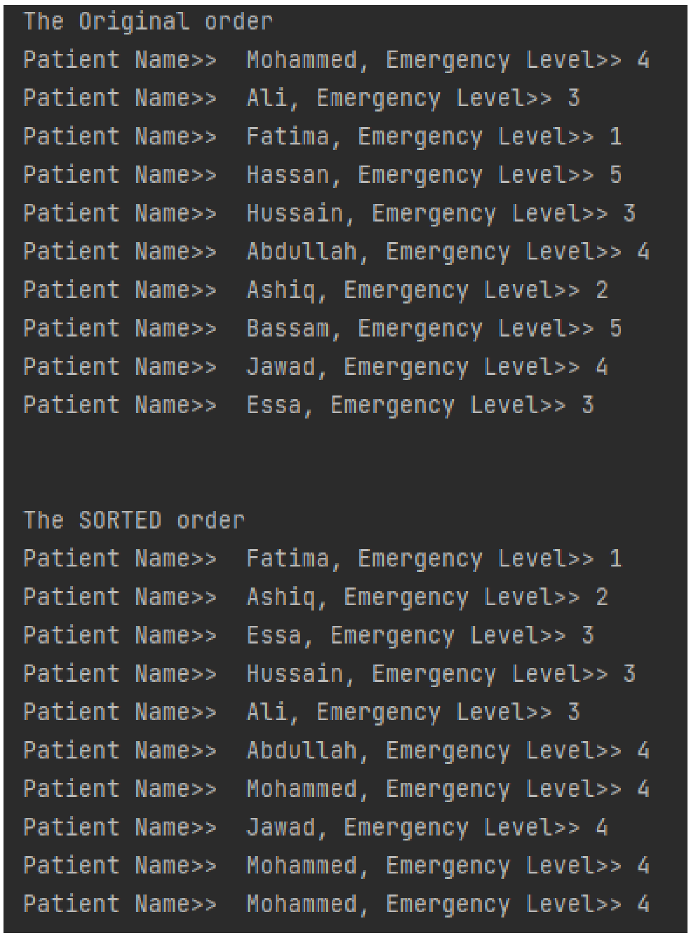


```java
public class Patient implements Comparable {
    String name;
    int emergencyLvl;

    public Patient(String name, int emergencyLvl){
        this.name = name;
        this.emergencyLvl = emergencyLvl;
    }

    @Override
    public int compareTo(Object o) {
        Patient p = (Patient) o;
        if(this.emergencyLvl == p.emergencyLvl)
            return this.name.compareTo(p.name);
        else if(this.emergencyLvl < p.emergencyLvl)
            return -1;
        else
            return 1;
    }

    @Override
    public String toString() {
        return "Patient" +
                " Name>>  " + name +
                ", Emergency Level>> " + emergencyLvl;
    }
}
```

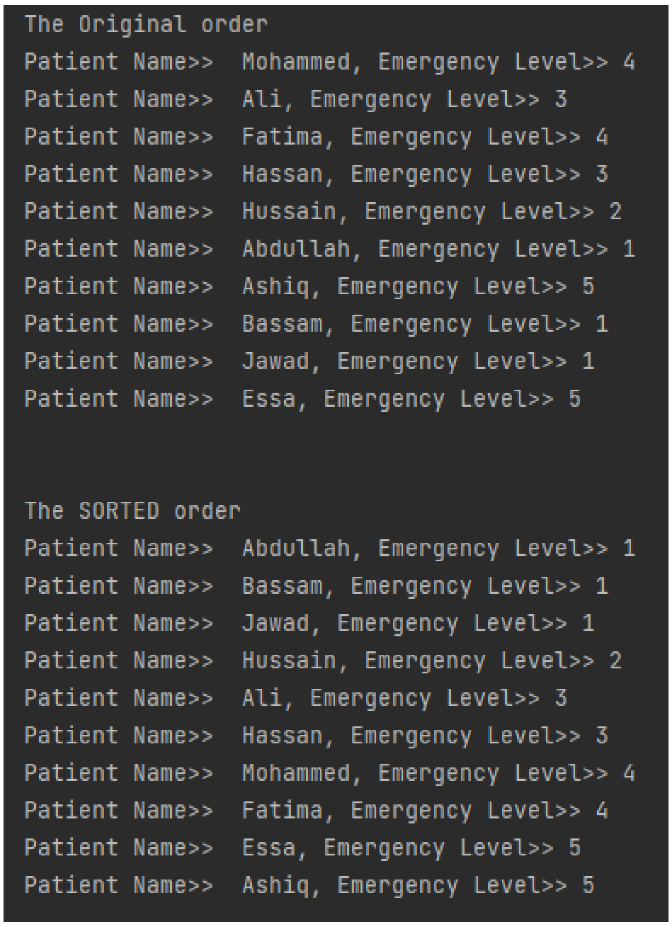


```java
public class Hospital {
    public static void main(String[] args) {
        String[] names = {"Mohammed", "Ali", "Fatima", "Hassan", "Hussain",
                "Abdullah", "Ashiq", "Bassam", "Jawad", "Essa"};

        Patient[] pList = new Patient[10];
        for(int i = 0; i < 10; i++)
            pList[i] = new Patient(names[i], (int)(1 + Math.random() * 5));

        BinaryHeap pHeap = new BinaryHeap(pList.length);
        System.out.println("The Original order");
        for(Patient p: pList) {
            System.out.println(p);
            pHeap.enqueue(p);
        }

        Comparable sorted_pList[] = pHeap.heapSort();
        System.out.println("\n\nThe SORTED order");
        for(Comparable i: sorted_pList)
            System.out.println(i);
    }
}
```

Output:



```
The Original order
Patient Name>>  Mohammed, Emergency Level>> 4
Patient Name>>  Ali, Emergency Level>> 3
Patient Name>>  Fatima, Emergency Level>> 1
Patient Name>>  Hassan, Emergency Level>> 5
Patient Name>>  Hussain, Emergency Level>> 3
Patient Name>>  Abdullah, Emergency Level>> 4
Patient Name>>  Ashiq, Emergency Level>> 2
Patient Name>>  Bassam, Emergency Level>> 5
Patient Name>>  Jawad, Emergency Level>> 4
Patient Name>>  Essa, Emergency Level>> 3


The SORTED order
Patient Name>>  Fatima, Emergency Level>> 1
Patient Name>>  Ashiq, Emergency Level>> 2
Patient Name>>  Essa, Emergency Level>> 3
Patient Name>>  Hussain, Emergency Level>> 3
Patient Name>>  Ali, Emergency Level>> 3
Patient Name>>  Abdullah, Emergency Level>> 4
Patient Name>>  Mohammed, Emergency Level>> 4
Patient Name>>  Jawad, Emergency Level>> 4
Patient Name>>  Mohammed, Emergency Level>> 4
Patient Name>>  Mohammed, Emergency Level>> 4
```



```
The Original order
Patient Name>>  Mohammed, Emergency Level>> 4
Patient Name>>  Ali, Emergency Level>> 3
Patient Name>>  Fatima, Emergency Level>> 4
Patient Name>>  Hassan, Emergency Level>> 3
Patient Name>>  Hussain, Emergency Level>> 2
Patient Name>>  Abdullah, Emergency Level>> 1
Patient Name>>  Ashiq, Emergency Level>> 5
Patient Name>>  Bassam, Emergency Level>> 1
Patient Name>>  Jawad, Emergency Level>> 1
Patient Name>>  Essa, Emergency Level>> 5


The SORTED order
Patient Name>>  Abdullah, Emergency Level>> 1
Patient Name>>  Bassam, Emergency Level>> 1
Patient Name>>  Jawad, Emergency Level>> 1
Patient Name>>  Hussain, Emergency Level>> 2
Patient Name>>  Ali, Emergency Level>> 3
Patient Name>>  Hassan, Emergency Level>> 3
Patient Name>>  Mohammed, Emergency Level>> 4
Patient Name>>  Fatima, Emergency Level>> 4
Patient Name>>  Essa, Emergency Level>> 5
Patient Name>>  Ashiq, Emergency Level>> 5
```