Name : Onkar Shinde
Batch : C2
Roll no : COSC26

Assignment 9

Input Code : public class HeapSort<T extends

Comparable<T>>

{

```
   private T data[];
private int length;

   HeapSort(T data[]) {
this.data = data;
      this.length = this.data.length;
   }

   public T[] buildMaxHeap(T[] tempData) {        for(int
i = (int) Math.floor(length/2); i >= 0 ; i--) {
         tempData = heapify(tempData, i);
      }
      return tempData;
   }

   public T[] heapify(T[] tempData, int node) {
int leftNode = node*2+1;        int rightNode =
node*2+2;        int maxNode = node;

      if(leftNode < length) {
         if(tempData[leftNode].compareTo(tempData[maxNode]) > 0){
maxNode = leftNode;
         }
      }

      if(rightNode < length) {
         if((tempData[rightNode].compareTo(tempData[maxNode])) > 0) {
maxNode = rightNode;
         }
      }

      if(maxNode != node) {           T temp =
tempData[node];           tempData[node] =
```

```
tempData[maxNode];
tempData[maxNode] = temp;
          tempData = heapify(tempData, maxNode);
       }
       return tempData;
   }

   public void sort(){
      this.data = buildMaxHeap(this.data);
while(length > 0) {           this.length--;

          T temp = this.data[0];
this.data[0] = this.data[length];
this.data[length] = temp;

          this.data = heapify(this.data, 0);

           }
   }

   public void printData() {
for(T i : this.data) {
System.out.print(i + " ");
      }

          System.out.println();
}

   public static void main(String[] args) {        /*
Sorting Integer Data using Heap Sort */
      Integer[] dataToBeSorted = {2, 8, 5, 3, 9, 1};
     HeapSort heapSort = new HeapSort<Integer>(dataToBeSorted);


       System.out.println("Given Data - ");
heapSort.printData();
heapSort.sort();
       System.out.println("Sorted Data - ");
heapSort.printData();

          }
```

}

Output :

```
Given Data -
2 8 5 3 9 1
Sorted Data -
1 2 3 5 8 9

=== Code Execution Successful ===
```