Q1: What is a complete binary tree? What is a heap?

Q2: Add the elements 4, 5, 1, 2, 9, and 3 into a heap in this order.

Draw the diagrams to show the heap after each element is added.

Q3: Show the heap after the root in the heap in Question2 is removed.

Q4: Show the steps of creating a heap using {45, 11, 50, 59, 60, 2, 4, 7, 10}.

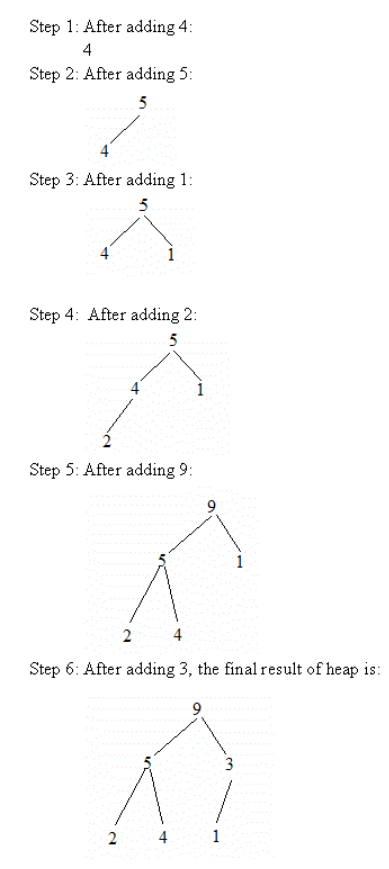
Q5: Which of the following statements are wrong? and explain why.

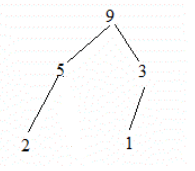
Heap<Object> heap1 = new Heap<>();

Heap<String> heap2 = new Heap<>();

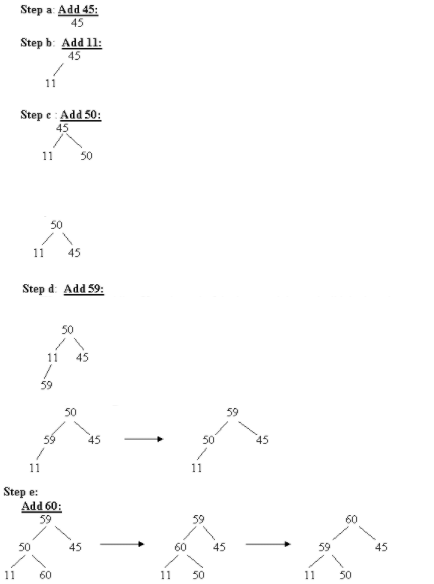
Heap<Calendar> heap3 = new Heap<>();

A1: A complete binary tree is a binary tree in which all the levels are completely filled except possibly the lowest one, which is filled from the left. A heap is a special data structure in Java. A heap is a tree-based data structure and can be classified as a complete binary tree. All the nodes of the heap are arranged in a specific order.

A2:

A3: 

A4:



A5: Heap<Object> heap1 = new Heap<>();

Is the wrong statement, because the class <Object> does not extend the comparable interface. So the compareTo method cannot be implemented on the classes Object and Number.