Separation of concerns

Term used by Edger Dijkstra in discussing the development of algorithms and mathematical proofs.

Rarely discussed, but its use is crucial in some situations.

Illustrate with the development of bubble down.

```java
/** class invariant – array b, integer size, HashMap map – is true, except that b[k] may be out of place.
 * Bubble b[k] down. */

public void bubbleDown(int k)
{
    int cR = 2 * k + 2;  int cL = 2 * k + 1;
    while (cL < size && cR < size + 1) {
        if (compareTo(cL, cR) == 0) // same priority for left and right
            if (compareTo(k, cL) < 0) {
                swap(cR, k); k = cR;
                cR = 2 * k + 2;  cL = 2 * k + 1;
            } else if (compareTo(k, cL) >= 0) { return; }
        else if (compareTo(cL, cR) > 0 || !map.containsValue(cR) && …) {
            // left has higher priority, or just left exists
            if (compareTo(k, cL) < 0) {
                swap(cL, k); k = cL;
                cR = 2 * k + 2;  cL = 2 * k + 1;
            } else if (compareTo(k, cL) >= 0) { return; }
        } else if (compareTo(cR, cL) > 0 || !map.containsValue(cL) && …) {
            // right has higher priority, or just right exists
            if (compareTo(k, cR) < 0) {
                swap(cR, k); k = cR;
                cR = 2 * k + 2;  cL = 2 * k + 1;
            } else if (compareTo(k, cR) >= 0) { return; }
        }
    }
}
```

Concerns:
- When to stop bubbling:
  1. k has no children.
  2. k belongs below a child.

Which child to bubble with?
- 1. k has no children.
- 2. k belongs below a child.

How to do the bubbling?
Separation of concerns

// inv: class invariant is true except k may be out of place
while (2*k+1 < size) {
    //Set c to the child to bubble with
    int c = 2*k+1;  // left child
    if (c+1 < size && compareTo(c+1, c) >= 0) c = c+1;

    if (compareTo(k, c) >= 0) return;
}

When to stop bubbling:  Which child to bubble with?
1. k has no children.
2. ! k belongs below a child.

Which child to bubble with?
How to do the bubbling?

// inv: class invariant is true except k may be out of place
while (2*k+1 < size) {
    //Set c to the child to bubble with
    int c = 2*k+1;  // left child
    if (c+1 < size && compareTo(c+1, c) >= 0) c = c+1;

    if (compareTo(k, c) >= 0) return;
    swap(k, c);
}

When to stop bubbling:  Which child to bubble with?
1. k has no children.
2. ! k belongs below a child.

How to do the bubbling?