

(for week starting 20th March)

1. For each of the following three scenarios, follow the design principle of “Know Thy User” and write down all the user characteristics you can think of. Can you think how the information you’ve listed would affect the design?
 - (a) A web site for a shop selling pure wool sweaters. The shop is located on the Isle of Skye.
 - (b) A touch screen located in a large Glasgow shopping mall. The screen displays information about the various shops in the mall.
 - (c) A simple computer puzzle game, developed in Britain, for the 5-7 year old age group.

2. Given that we have an array called `numbers` created like this:

```
int[] numbers = { 4, 15, 99, 6, 21, 12, 88, 67, 3 };
```

- (a) The Java code below is the core algorithm in Quicksort. Trace through the following fragment of Java *in detail* to see what effect it has on the array:

```
1)      int i = 0, j = numbers.length-1;
2)      int pivot = numbers[i + (j-i)/2];
3)      while (i <= j)
4)      {
5)          while (numbers[i] < pivot)
6)          {
7)              i++;
8)          }
9)          while (numbers[j] > pivot)
10)         {
11)             j--;
12)         }
13)         if (i <= j)
14)         {
15)             int tmp = numbers[i];
16)             numbers[i] = numbers[j];
17)             numbers[j] = tmp;
18)             i++;
19)             j--;
20)         }
21)     }
```

- (b) What overall task do you think that this has achieved? If n is the length of `numbers`, how many steps does the algorithm take?
- (c) The “pivot” value has an essential role. In the algorithm above, the **central value in the array** is chosen as the pivot – this is quick to find, but does not guarantee the best overall Quicksort performance. Quicksort works best if the pivot value is the **median** value in the array – that is the value which, as closely as possible, has *the same number of values smaller than it in the array as the number of values larger than it* (**15 in the array above**). Why do you think that this would give the best behaviour? Try to design an algorithm (in principle, not in Java) to find the median value in an array (hard!).