“Class 1.5”: about algorithms vs programming

There are several steps to computing, and the most important are not done on the computer:

1) Recognize and clarify a problem to solve.
2) Devise a method for solving it.
3) Carry out the solution.
4) Test, correct and revise the method as needed.

Programming

1) Recognize and clarify a problem to solve.

2) Devise a method for solving it.

3) Carry out the solution.

   - One way of doing this is to write a computer program and run it.

4) Test, correct and revise the method as needed.

   - Using a computer can help with this step too.

Algorithm design

1) Recognize and clarify a problem to solve.
2) Devise a method for solving it.
    - This you have to do with your brain.
3) Carry out the solution.
4) Test, correct and revise the method as needed.

Algorithm design vs programming

- **Algorithm**: the step-by-step procedure for solving the problem.
- **Program**: the implementation in “code” that instructs the computer how to execute the algorithm.

Note: the algorithm could be done by a person instead of a computer. This is actually often a good way of designing an algorithm: solve the problem repetitively and observe what procedure you follow.

Show of hands: which do you find hardest?

1) Recognize and clarify a problem to solve.

2) Devise a method for solving it.

3) Carry out the solution.

   a) Test on paper or in thought.

   b) Program a computer to do it.

   c) Run the program.

4) Test, correct and revise the method as needed.

How to make algorithm design and programming MUCH harder

1) Recognize and clarify a problem to solve.

2) Devise a method for solving it AND

3) Implement the solution, simultaneously. (NO!)

4) Test, correct and revise the method as needed.

Step 2 is MUCH harder if you're trying to do step 3 a the same time. This is probably because step 3 is infinitely harder if you haven't finished step 2.

How to make algorithm design and programming MUCH easier

1) Recognize and clarify a problem to solve.

2) Devise a method for solving it (first).

3) Implement the solution (after step 2).

4) Test, correct and revise the method as needed.

Keep the steps separate.

More advice on algorithm design

- Identify the input data.
- Identify the output data.
- Identify the mathematical relationships between the two.
- Think about the “mechanical” steps you'd go through to produce the output from the input.
- Write it on paper as a flow chart or “pseudo-code” before trying to code.