Problem Solving with the Computer

The first thing a programmer must be able to do is to solve the problem themselves. If you can't do it yourself then how can you possible tell a stupid thing like a computer how to go about solving the problem? Remember, the only thing a computer can do is to calculate and evaluate Boolean expressions (you know like if a<10) when told in the correct language (syntax). Once the programmer is able to solve the problem in most cases mathematically they are then ready to proceed.

Concrete Problem:

If we already understand the problem we're going to solve, the next step is to figure out a plan of attack which we will then break down into steps small enough that they can be expressed in Java. This is called **stepwise refinement**, since we start out with a "coarse" solution and refine it until the steps are within the capability of the Java language. For a complex problem, this may take several intermediate steps; but let's start out with a simple example. Suppose we want to know how much various sizes of concrete pads will cost.

1. First we need to get the cost/yd of the concrete and the width, length and depth of the pad.
2. We need to calculate the volume of the pad and compute the number of yards.
3. Next we need to multiply the number of yards by the cost.
4. ..and finally display the result.

This in turn can be broken down further, as follows:

1. Get the INPUT.
    a. Assign the width to a variable
    b. Assign the length to a variable
    c. Assign the depth to a variable
    d. Assign the cost of the concrete to a variable
2. Multiply the width times the length times the depth
3. Multiply that result b y the cost of the concrete.
4. Display result.

This looks okay, except that the measurements may be in feet and/or inches and the cost of the concrete is in cubic yards. So some additional steps would be required.

What steps would now be necessary? Write out the steps you would take and submit your response in the concrete submission link.

You've probably noticed that this is a much more detailed description than would be needed to tell a human being what you want to do. That's because the computer is extremely stupid and literal: it does only what you tell it to do, not what you meant to tell it to do. Unfortunately, it's very easy to get one of the steps wrong, especially in a complex program. In that case, the computer will do something ridiculous and you'll have to figure out what you did wrong. This **debugging**, as it's called, is one of the hardest parts of programming. Actually, it shouldn't be too difficult to understand why that is the case; after all, you're looking for a mistake you've made yourself. If you knew exactly what you were doing, you wouldn't have made the mistake in the first place.

I hope that this brief discussion has made the process of programming a little less mysterious. In the final analysis, it's basically just logical thinking.

In Summary: Almost all programs can be broken down into 3 tasks that need to be accomplished. Task #1 is to get the required data. Call this INPUT. This can be as we have seen so far in the form of an assignment statement(s). Task #2 is to explain in logical steps how to solve the problem. This is called the algorithm or PROCESSING. Task #3 is to report the findings in a readable form for the user, call OUTPUT.

So keep in mind: INPUT→ PROCESSING → OUTPUT

Change Problem:

Additional challenge: Pretend you need to write a program for a cash register so it can tell it's human how much change to return on a transaction. (Change must be made up of the smallest number of bills and coins possible. List the directions you would give the computer. Remember it's really accurate when it comes to calculating but it must be told exactly how to perform those calculations. Submit your solution using the change submission link.