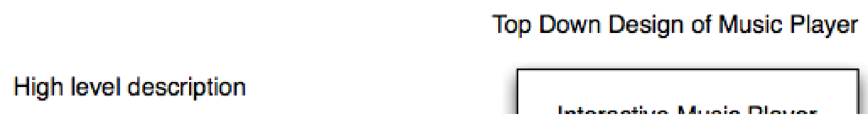
[0406] Top-Down Design

One of the key reasons that Structure Programming was seen as such big improvement over the spaghetti code that came before it was that it also included an easy-to-use method for creating really large programs. That method was called "Top-Down Design" and it sprung directly from the modularity concepts we just discussed.

The basic idea behind Top-Down Design is that you first describe your problem, in English, at a high level that a human can easily understand and then you break it down into a relatively small collection of somewhat more detailed steps. Then you break down each of those steps into still smaller steps, and so on until you have a sequence of steps that can be easily implemented as programming code.

At each stage of the decomposition process, you give each step a name, a collection of inputs and define the single output that it generates. That makes it easy to combine all of the smaller modules together into you program once all the implementation is done (at least in theory!). Here's a diagram that tries to illustrate the process:



One of the key questions that needs to be addressed when doing top-down design is "What is the right size for each of my modules? When should I break something down into smaller steps versus just adding more steps to the current module?"

The answer is "It depends..." which isn't super-helpful. This question gets to the heart of an issue programmers face every day - "When should I create a new method/function?" We will be exploring that question throughout the rest of this chapter.