

# Top Down Design

Definition: A software development technique that imposes a hierarchical structure on the design of the program. It starts out by defining the solution at the highest level of functionality and breaking it down further and further into small routines that can be easily documented and coded (The Free Dictionary: top-down programming).

Also known as:

- divide and conquer
- stepwise refinement
- structured programming

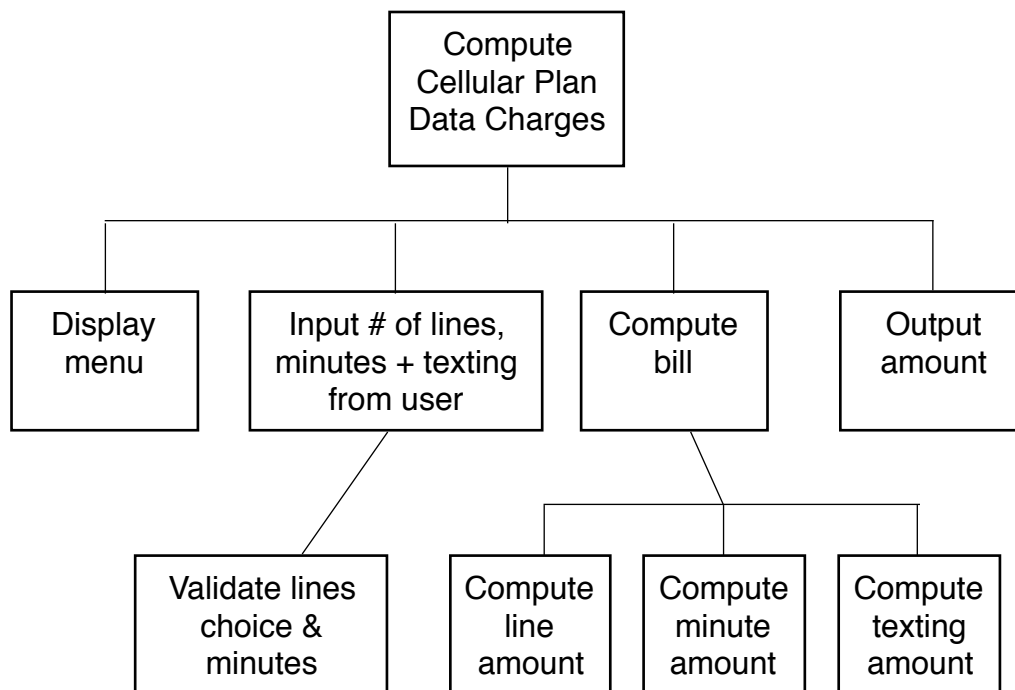
Complex problems can be solved using top-down design, where

- We break the problem into a sequence of smaller tasks
- Then break the sub-tasks into tasks
- Soon, each of the tasks will be easy to do

Advantages to top down design:

- Breaking the problem into parts helps us to clarify what needs to be done.
- At each step of refinement, the new parts become less complicated and, therefore, easier to figure out.
- Parts of the solution may turn out to be reusable.
- Breaking the problem into parts allows more than one person to work on the solution.

Top Down Design usually results in a hierarchy chart that describes the tasks that must be accomplished. Each “leaf” task is a module (often implemented using a function). Once you have the tasks listed, you can start coding each one, and then adding code to sequence them correctly.



The Top Down Design chart does not specify control flow. It does not specify looping or branching. Sometimes the code to carry out each task ends up being interleaved in the final program. In this program the input from the user and output to the file are interleaved. Also, the tasks are performed once for each floor, and the design doesn't specify what kind of loop to use (count-controlled, sentinel-controlled, etc.).

