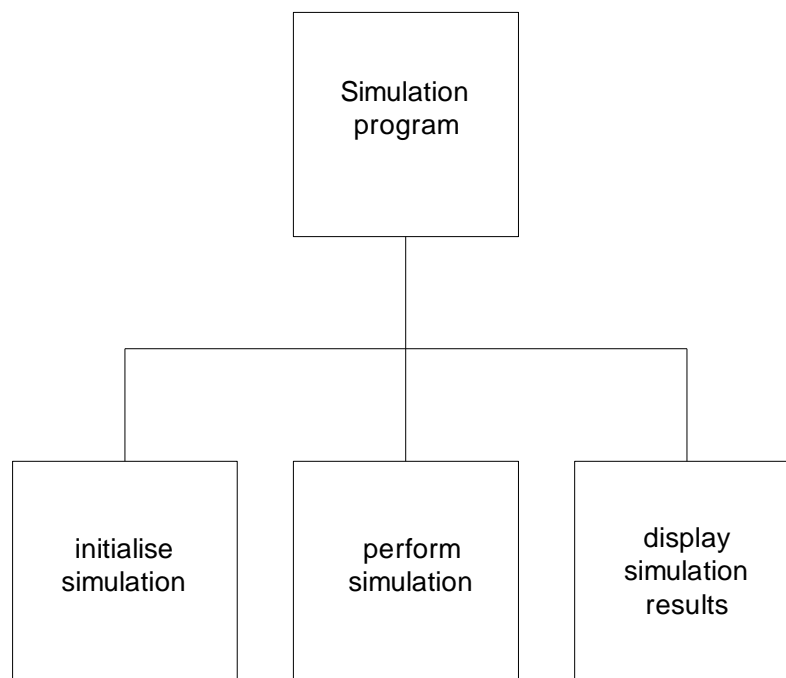


Functional Decomposition

- for designing detailed structure of programs/modules and
- large scale architecture
- starts with a single grand statement of what program is to do. This is next re-written as a series of simpler steps, i.e.
- tackles a problem by decomposing it into sub-problems and the dealing with them in turn
- evolved with structured programming, it is a general purpose flexible design method based on structured programming

Example



Example

make a cup of coffee

can be refined to

```
boil water
get a cup
put coffee in cup
add water
stir
add milk
add sugar to taste
```

Could refine add sugar to taste further as:

```
if sugar required then
    put sugar in cup
    stir
end if
```

Functional decomposition starts with pseudocode and ends with program statements

Uses

- ***top down design*** - break a task into smaller more manageable subtasks
- ***stepwise refinement*** - refine a general or descriptive statement in pseudocode into a more precise one(s) or into a few lines of code
- ***levels of abstraction*** - whereby to understand a solution at one level it is not necessary to understand anything about the workings of the level beneath. Need to know ***what*** they do, not ***how***

1. Program could be refined top down, level by level in a breadth first manner

or

2. take one subtask, decompose it and look at one the sub-subtasks and so on until you arrive at code. Depth first approach. Useful to explore what may be a difficult or interesting part of the program

- Psuedocode design does not lend itself to verify correctness of design in a formal way
- It facilitates an intuitive understanding, approach to a problem
- Because of its flexibility can lead to alternative designs, flexibility = lack of guidance
- No mention of data -idea is that data structures will emerge in the design as actions become more specific, and it should become clear as to how the data will be operated on
- Ignoring data until operations become clear can cause problems later on when a new operation may have to be added. New operation may require data to be structured in a different way. Becomes messy then.
- May not be always possible to delay decision about data structures
- Because of the possibility of many solutions, the developer may be obliged to look at a few of them so as to select the optimal one.
- Well suited to more creative design
- Suited to numerical methods programming where computations are already known in advance from the mathematical theory