

Solutions

Each answer shows the steps, then the **result**.

2.1

- the **hierarchical decomposition** of a program into **modules/subroutines** and the **parameters** passed between them

2.2

- the **parameters** passed **into** a module and the **results returned** from it

2.3

- the **states** a system can be in and the **events** (transitions) that move it between them

2.4

- `insert coin` (the coin event)

3.1

- `MakeTea` at the top with lines down to `BoilWater` and `AddMilk`
- `Cups` shown as a parameter **in** on the `AddMilk` link

3.2

- three states `Idle`, `Paid`, `Dispensing`
- `Idle` $\rightarrow$ (coin) `Paid`
- `Paid` $\rightarrow$ (select) `Dispensing`
- `Dispensing` $\rightarrow$ (done) `Idle`

3.3

- any one: it makes **missing or invalid transitions** easy to spot; it documents the system's behaviour clearly for all events

3.4

- the two modules are called repeatedly in a **loop** — it shows **iteration**

3.5

(a) `Idle` + coin inserted $\rightarrow$ `Collecting`

- `Collecting` + enough money and item available $\rightarrow$ `Dispensing`
- `Collecting` + sold out $\rightarrow$ `OutOfStock`
- `Dispensing` + item delivered $\rightarrow$ `Idle`

(b) it shows the **states** the system can be in and which **events** move it between them

- the behaviour over time, including what is not allowed, which a structure chart cannot express because it only shows which module calls which

(c) a curved arrow means **iteration**: the modules it encloses are called **repeatedly**

- for as long as the condition on the loop holds
- (*a diamond, by contrast, marks selection*)

(d) parameters are drawn as small arrows beside the line joining the two modules

- the parameter passed **down** to **CheckStock** — the item code — points from **ProcessSale** towards it
- the Boolean returned points **back up** towards **ProcessSale**, and each arrow is labelled with the data it carries

(e) the structure chart shows how the program is **decomposed** into modules and what data passes between them

- the state-transition diagram shows the **behaviour** of the system as a whole
- the flowchart shows the **detailed logic inside one module**