Solutions

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

2.1

- removing details that do not affect the problem, keeping only those that do

2.2

- any four relevant sub-problems, e.g. lend books, return books, manage members, search the catalogue

2.3

- kept: e.g. temperature, chance of rain, day
- left out: e.g. air pressure at every height, the satellite used

2.4

- “a little” is ambiguous: it has no single meaning
- e.g. “add 50 ml of water”

3.1 B

- the same structure found in several places

3.2 C

- only what a traveller needs is kept

3.3 C

- an algorithm is independent of any language

3.4

(a)

- three sensible sub-problems, e.g. booking, payment, reminders
- two tasks under booking, e.g. show free courts, reserve a court
- two under payment, e.g. calculate fee, take card payment
- two under reminders, e.g. schedule message, send message

(b)

- e.g. booking and reminders both look up a member’s details and a time slot
- one routine can be written once and reused, saving time and errors

(c)

- include: e.g. court number, the times it is free
- include: e.g. the sport it is for, the price per hour
- leave out: e.g. the colour of the floor, as it does not affect a booking

(d)

- the sub-problems can be shared among team members and built at the same time
- each part can be tested on its own, so errors are easier to find