Solutions

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

2.1

- to **plan, manage and control** the project: to build the right product, on time, to good quality

2.2

- **Analysis → Design → Coding → Testing → Maintenance**

2.3

- keeping the released program **working and useful**: fixing faults found in use, and adapting it to new needs

2.4

Acceptance testing; adaptive maintenance.

3.1

- **analysis**: find and document **what** the program must do (the requirements)
- **testing**: run the program against **test data**, compare with expected results, and fix the bugs found

3.2

- waterfall, any one: clear/simple, well documented, easy to manage with stable requirements
- iterative, any one: problems caught earlier, copes with changing requirements, a working partial version appears sooner

3.3

- a suitable model: **RAD** (or iterative)
- because it works in **short repeated cycles with user feedback**
- so changing requirements can be handled without restarting the whole project

3.4

(a) changing the program so it still works in a **new environment**: a new operating system, new hardware, or a change in the law

(b) e.g. the OS is upgraded; a new tax rate / data-protection law; the program must run on new hardware

3.5

- analysis
- design
- coding
- testing
- maintenance

3.6

- **stub testing**
- each unfinished module is replaced by a simple **stub** with the same name and parameters
- the stub only outputs a message, e.g. "Test_A called", or returns a fixed value, so the main program can run and be tested