

12 Software Development – Part 2 — Answers

Answers

12.1 How the program is broken into modules (subroutines) and the parameters passed between them (the calling hierarchy).

12.2 The states a system can be in and the events that move it between them; e.g. a vending machine, a lock, or a traffic light.

12.3 White-box testing is designed from the code's internal structure (to cover every path); black-box testing is designed only from the specification (inputs → expected outputs).

12.4 (a) normal: e.g. 50; (b) abnormal: e.g. 200 or −10; (c) boundary: 0 or 100 (and the rejected value just outside, like 101).

12.5 That separate modules work correctly together — that the interfaces between them are right.

12.6 Black-box checks the program meets its specification (right outputs for given inputs) but can miss untested code paths; white-box checks every branch of the code runs but might miss a missing feature. Together they cover both “does it meet the spec” and “is all the code exercised”.

12.7 For example: inserting a coin changes **Locked** → **Unlocked**; pushing the bar changes **Unlocked** → **Locked**. (Each event names a state change.)