Solutions

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

2.1

(a)

- at rest; physical protection of the drive, or encryption so a stolen disk is unreadable

(b)

- in transit; protect the link, or encrypt so an intercepted packet is unreadable

(c)

- in use; access control limiting who may view or edit

2.2

- the data must be **unencrypted** to be processed, so encryption cannot protect it at that moment and the only defence left is access control over who or what may use it

2.3

(a)

- so that the storage, transmission and handling rules the law requires can be applied to exactly those data

(b)

- any two of: how the data must be stored; how they may be transmitted; who may handle them; retention or disposal rules; breach reporting

2.4

- cryptography policy - any two of: approved algorithms for each use; minimum or maximum key lengths; key-generation requirements; key-storage requirements. Web application security policy - any two of: when an application is subject to assessment; how the assessment is carried out (tools or framework); timelines for remediating vulnerabilities by risk level

2.5

(a)

- RuBAC / rule-based

(b)

- RBAC / role-based

(c)

- DAC / discretionary

(d)

- MAC / mandatory

2.6

(a)

- allowed - reading down, permitted by the Simple Security Property, which only forbids reading *above* your level

(b)

- not allowed - the Simple Security Property forbids reading above your level

(c)

- allowed - writing up, permitted; the Star Security Property only forbids writing *below* your level

2.7

(a)

- 640

(b)

- 755

(c)

- 600

2.8

(a)

- 764

(b)

- `chmod g-w` followed by the filename

(c)

- it changes only the one permission asked for, so an owner or others setting cannot be altered by mistake while re-typing three digits

3.1 B

- Access follows the role, so RBAC

3.2 B

- Owner rw = 6, group r = 4, others none = 0

3.3 C

- Writing up is allowed; the Star Property forbids writing *below* your level, and the Simple Property governs reading, not writing

3.4 B

- A statement of a required parameter is a rule about what must be in place - managerial. The other three act on traffic or data directly

3.5

(a)

- data **in transit**; encrypt the transfer, and protect the physical campus link

(b)

- **RBAC**. Subjects are the three groups of users, objects are the data files, operations are read and modify. Each of the three kinds of user maps cleanly onto a role, so access is granted once per role rather than per person, and a new undergraduate inherits the correct access on the day they are added, with nothing to revoke individually when the statistician's contract ends. Accept DAC only if the answer argues the owner should decide *and* notes the loss of central consistency

(c)

- **RuBAC**. A rule-based layer evaluates the condition - weekday - and allows or denies on top of the role the subject already holds, so the two work together rather than replacing each other

(d)

- least privilege = each entity is given exactly the access it needs to do its job and no more. `-rwxrwxrwx` grants **everyone** write and execute, so an undergraduate who should only read can alter or delete the research data, and any compromised account on the system can too. Set **640**: owner 6 = read+write for the researcher who maintains the file; group 4 = read only, which is all the undergraduates need; others 0, because nobody outside the project should reach it at all. Accept 750/740 with justification tied to whether the file must be executed

(e)

- any two parameters: approved algorithm, minimum key length, key-generation requirements, key-storage requirements. A policy is still needed because installed software can be configured weakly - a short key, a deprecated algorithm, or a key left beside the backup it protects - and the policy is what makes the required setting checkable and auditable. Encryption is a technical control; the policy is the managerial control that says what "encrypted" must mean here