KK02 — Characteristics of functional and non-functional requirements, constraints and scope

Programming · Comprehensive exam study guide

These four terms underpin the whole study. Functional and non-functional requirements describe **what the solution must do and how well**; constraints are the **limits** you work within; scope sets the **boundaries** of what is and isn't included. Exam questions frequently ask you to classify a statement or give examples in a scenario.

Requirements

Type	Meaning	Example (a canteen ordering app)
Functional	A task or function the solution **must perform**.	“Calculate the total price of an order”; “allow a student to log in”.
Non-functional	A **quality or property** describing how well the solution performs — often about usability, performance, reliability, security.	“Each screen loads within 2 seconds”; “usable by Year 7 students”; “data is encrypted”.

Constraints

A **constraint** is a limitation or restriction that influences how the solution can be developed. Constraints are commonly grouped as economic, legal, social and technical.

- Economic — budget, cost of tools/licences, time available.
- Technical — hardware, software, programming language, device capabilities.
- Social — user expectations, accessibility, ethical concerns.
- Legal — privacy, copyright, data-protection obligations.

Scope

Scope defines the boundaries of the solution — what it **will** and **will not** do, and for a given version. Clear scope prevents ‘scope creep’ (uncontrolled growth of features).

A good scope statement lists inclusions and explicit exclusions, e.g. “Version 1 lets students order food and pay; it does **not** manage staff rosters.”

Key definitions to memorise

Functional requirement	A specific task or behaviour the solution must perform.
Non-functional requirement	A quality or attribute describing how well the solution must perform (e.g. speed, usability, security, reliability).
Constraint	A limitation or restriction (economic, legal, social or technical) that influences how a solution can be developed.
Scope	The defined boundaries of a solution — the features and functions that are and are not included.

Exam technique

EXAM TIP

- To classify: ask ‘is this a thing the system DOES (functional) or a quality of HOW it does it (non-functional)?’
- When giving examples, write them as testable statements, not vague aims.
- If asked about scope, mention BOTH inclusions and exclusions — boundaries cut both ways.

COMMON MISTAKES

- Calling 'the app must be easy to use' a functional requirement — it is non-functional.
- Confusing a constraint (a limit you can't change) with a requirement (something the solution must do).
- Describing scope only as 'what it does', forgetting the exclusions that set the boundary.

Quick revision checklist

- ■ I can define and give a testable example of a functional and a non-functional requirement.
- ■ I can name the four constraint categories and give an example of each.
- ■ I can write a scope statement with inclusions and exclusions.

Exam practice — questions & model answers

Q1. Define a functional requirement and a non-functional requirement, giving an example of each.

Define · 4 marks

Model answer

- Functional = a task the solution must perform, e.g. ‘calculate the total cost’ (2).
- Non-functional = a quality of how it performs, e.g. ‘load each screen within two seconds’ or ‘usable by Year 7 students’ (2).

Q2. Identify the type of each requirement: (a) the system encrypts stored passwords; (b) the system lets a user reset a password.

Identify · 2 marks

Model answer

- (a) Non-functional — a security quality of the solution (1).
- (b) Functional — a task the system performs (1).

Q3. Describe two constraints that could influence the development of a school event-booking app.

Describe · 4 marks

Model answer

- Economic — a limited budget restricts paid tools/hosting, so free libraries must be used (2).
- Technical — it must run on the school's existing tablets, limiting frameworks/features that can be used (2). (Legal/social also acceptable.)

Q4. Explain why clearly defining the scope of a solution is important.

Explain · 2 marks

Model answer

- It sets agreed boundaries of what will and won't be built (1),
- which prevents scope creep and keeps the project achievable within time and budget (1).