

KK03 — Design tools for representing modules

Programming · Comprehensive exam study guide

STUDY DESIGN — YOU MUST KNOW

- data dictionaries
- mock-ups
- object descriptions
- input-process-output (IPO) charts
- pseudocode

Design tools turn requirements into a plan a programmer can build from. You must know the **purpose**, **typical contents** and **when to use** each of the five tools, and be able to read and write them — pseudocode and IPO charts are especially common in the exam.

The five design tools

Tool	Represents	Typical contents
Data dictionary	The data the solution stores/uses	Field/variable name, data type, size/format, description, validation, example.
Mock-up	The look of an interface	Sketch/layout of a screen: controls, labels, placement, navigation.
Object description	The structure of an object/class	Object name, its properties (attributes) and methods (behaviours).
IPO chart	The logic of a module as Input → Process → Output	Three columns: inputs taken, processing steps, outputs produced.
Pseudocode	The step-by-step algorithm	Structured English with sequence, selection and iteration; not language-specific.

Reading & writing pseudocode

Pseudocode uses plain keywords (IF/THEN/ELSE/ENDIF, WHILE/ENDWHILE, FOR/NEXT, INPUT, OUTPUT) and indentation to show control structures. It should be detailed enough to translate into code but free of language-specific syntax.

EXAMPLE

```
BEGIN → INPUT mark → IF mark >= 50 THEN OUTPUT "Pass" ELSE OUTPUT "Fail" ENDIF → END.
```

Data dictionary detail

A data dictionary documents each data element so the team agrees on type, size and validation before coding. It directly supports validation (KK10) and reduces type-mismatch errors.

Key definitions to memorise

Data dictionary

A table documenting each data element: name, data type, size/format, description and validation.

Mock-up	A visual draft of a user interface showing layout, controls and navigation.
Object description	A description of an object/class listing its properties (attributes) and methods (behaviours).
IPO chart	A three-column chart mapping the Inputs, Processing and Outputs of a module.
Pseudocode	A structured, language-independent description of an algorithm using plain keywords and indentation.

Exam technique

EXAM TIP

- If asked to choose a tool, justify by purpose: data → data dictionary; screen → mock-up; logic → pseudocode/IPO.
- When writing pseudocode, show the control structure clearly (indent the body, close with ENDIF/ENDWHILE).
- An IPO chart's Process column lists steps/calculations — not the inputs again.

COMMON MISTAKES

- Putting validation rules in a mock-up — they belong in the data dictionary.
- Writing real Python/code instead of pseudocode (pseudocode is language-independent).
- Confusing an object description (attributes + methods) with a data dictionary (data fields).

Quick revision checklist

- I can name all five design tools and what each represents.
- I can write pseudocode using sequence, selection and iteration.
- I can complete an IPO chart and a small data dictionary for a scenario.

Exam practice — questions & model answers

Q1. Name the design tool best suited to documenting the data a solution will store, and state two things it records.

Name/State · 2 marks

Model answer

- Data dictionary (1).
- Records any two of: field name, data type, size/format, description, validation (1).

Q2. Write pseudocode that inputs a temperature and outputs “Hot” if it is above 30, otherwise “Mild”.

Write · 3 marks

Model answer

- INPUT temperature (1)
- IF temperature > 30 THEN OUTPUT “Hot” (1)
- ELSE OUTPUT “Mild” ENDIF (1).

Q3. Distinguish between an IPO chart and a mock-up.

Distinguish · 2 marks

Model answer

- An IPO chart represents the logic of a module as inputs, processing and outputs (1);
- a mock-up represents the visual layout/appearance of an interface (1).

Q4. Complete an IPO chart for a module that calculates and displays the area of a rectangle.

Apply · 3 marks

Model answer

- Input: length, width (1).
- Process: $\text{area} = \text{length} \times \text{width}$ (1).
- Output: display area (1).