Exercise walk-through

9.1
Computational solution skills

International Baccalaureate · IB Diploma · Computer Science · SL

What makes a good problem

Question	Why it matters
Can I talk to the user several times?	you need their requirements and feedback
Does it need more than a few lines of code?	you must show a range of techniques
Can I finish it in the time given?	an unfinished product cannot be tested

Use the evidence

Running example: the chess coach spends two hours a week pairing players by hand.

Decompose before you code

Part	Design product
data	fields, data types and an example; a file or database layout
algorithms	a flowchart or pseudocode for each non-trivial process, such as pairing
screens	a sketch of each screen, checked with the user
testing	a test plan, written before coding

Use the evidence

Chess program: players and results, rating updates, weekly pairings, the coach's screens.

Explaining a technique

- Name it, show an extract, explain it here, justify it.
- A list of Python features is not an explanation.

```
def make_pairings(players):
    """Pair (name, rating) tuples; return the pairs and the bye."""
    ranked = sorted(players, key=lambda p: p[1], reverse=True)
    bye = None
    if len(ranked) % 2 == 1:
        bye = ranked.pop()      # lowest rating sits out
    pairs = []
    for i in range(0, len(ranked), 2):
        pairs.append((ranked[i][0], ranked[i + 1][0]))
    return pairs, bye
```

Exercises 2.1·2.2

2.1 Rewrite “The program should be fast” as a testable success criterion. [2]

2.2 A criterion says “The program accepts a quantity from 1 to 20.” Give one normal, one boundary and one invalid test value. [3]

Exercises 2.1·2.2

2.1 Rewrite “The program should be fast” as a testable success criterion. [2]

2.1

- names a measurable time, e.g. “within 2 seconds”
- names the action and size, e.g. “pairings for up to 40 players appear after pressing *Pair*”

2.2 A criterion says “The program accepts a quantity from 1 to 20.” Give one normal, one boundary and one invalid test value. [3]

2.2

- normal: any value from 2 to 19, e.g. **10**
- boundary: **1** or **20**
- invalid: **0**, **21**, or a non-number such as “five”

Exercises 2.3·3.1

2.3 A write-up lists “variables, loops, if statements, functions” as its techniques. Explain why this earns little. [2]

3.1 For “age 11 to 18 accepted”, which value is boundary data? [1]

- ☐ **A** 15
- ☐ **B** 18
- ☐ **C** 19
- ☐ **D** “eleven”

Exercises 2.3·3.1

2.3 A write-up lists “variables, loops, if statements, functions” as its techniques. Explain why this earns little. [2]

2.3

- it names features without saying how or why they are used in this product
- it should show a code extract of a key technique and explain what it does and why it was chosen

---

3.1 For “age 11 to 18 accepted”, which value is boundary data? [1]

- ☐ **A** 15
- ☐ **B** 18
- ☐ **C** 19
- ☐ **D** “eleven”

3.1 B

- 18 is the top edge of the allowed range

Exercises 3.2·3.3

3.2 When should the test plan be written? [1]

- **A** after the program is finished
- **B** before coding, from the success criteria
- **C** only if a test fails
- **D** after the evaluation

3.3 Which is an acceptable way to use code from an AI tool? [1]

- **A** paste it in without comment
- **B** acknowledge the tool, show what was changed, and explain the code
- **C** rename the variables so it looks original
- **D** use it only in the test plan

Exercises 3.2·3.3

3.2 When should the test plan be written? [1]

- **A** after the program is finished
- **B** before coding, from the success criteria
- **C** only if a test fails
- **D** after the evaluation

3.2 B

- it fixes what must be shown to work before the product exists

---

3.3 Which is an acceptable way to use code from an AI tool? [1]

- **A** paste it in without comment
- **B** acknowledge the tool, show what was changed, and explain the code
- **C** rename the variables so it looks original
- **D** use it only in the test plan

3.3 B

- acknowledging, adapting and explaining is academic integrity

Exercise 3.4 (a)

3.4 A school canteen wants an app where students pre-order lunch by 10:00, choosing one main dish and up to two sides.

(a) Write **two** testable success criteria for the app. [2]

Exercise 3.4 (a)

3.4 A school canteen wants an app where students pre-order lunch by 10:00, choosing one main dish and up to two sides.

(a) Write **two** testable success criteria for the app. [2]

(a) Any two testable criteria, e.g.:

- an order with one main and up to two sides is saved, and the student sees a confirmation
- orders after 10:00 are refused with a message; the kitchen list prints in under 5 s

Exercise 3.4 (b)

3.4 A school canteen wants an app where students pre-order lunch by 10:00, choosing one main dish and up to two sides.

(b) For the rule “up to two sides”, write a test plan with one normal, one boundary and one invalid test, each with its expected result. [3]

Exercise 3.4 (b)

3.4 A school canteen wants an app where students pre-order lunch by 10:00, choosing one main dish and up to two sides.

(b) For the rule “up to two sides”, write a test plan with one normal, one boundary and one invalid test, each with its expected result. [3]

(b)

- normal: one side, **accepted**
- boundary: two sides, **accepted**
- invalid: three sides, **rejected with a message**

Exercise 3.4 (c)

3.4 A school canteen wants an app where students pre-order lunch by 10:00, choosing one main dish and up to two sides.

(c) The app sorts the day's orders by collection time before printing them for the kitchen. Explain why this technique was chosen. [2]

Exercise 3.4 (c)

3.4 A school canteen wants an app where students pre-order lunch by 10:00, choosing one main dish and up to two sides.

(c) The app sorts the day's orders by collection time before printing them for the kitchen. Explain why this technique was chosen. [2]

(c)

- the kitchen prepares meals in the order they are collected
- so sorting by collection time gives a list that can be worked through from the top

Exercise 3.4 (d)

3.4 A school canteen wants an app where students pre-order lunch by 10:00, choosing one main dish and up to two sides.

(d) In testing, an order placed at 10:00 exactly was rejected, although the requirement says “by 10:00”. Evaluate this result and state what the student should do. [3]

Exercise 3.4 (d)

3.4 A school canteen wants an app where students pre-order lunch by 10:00, choosing one main dish and up to two sides.

(d) In testing, an order placed at 10:00 exactly was rejected, although the requirement says “by 10:00”. Evaluate this result and state what the student should do. [3]

(d)

- the criterion is **not met**: “by 10:00” includes 10:00
- the likely cause is a boundary error, `<` used where `<=` was needed
- fix the comparison, re-run the test and record both results as evidence