Course notes

International Baccalaureate · IB Diploma · Computer Science · SL

IB IB Diploma · Computer Science · SL: teaching notes

Version: First assessment 2027 target

These notes cover selected topics of the course; they are not the whole specification.

Assessment and course boundaries

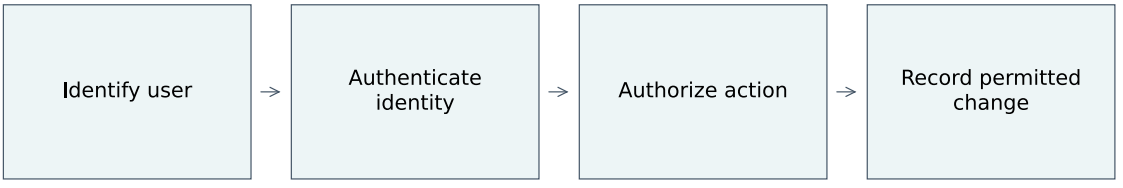
- This ordering records the 2027 A/B structure separately from the 2014 course final assessment in 2026.
- Do not carry forward old Options A–D or the old assessment weights. A.4 machine learning and B.3 OOP are in the new public structure; B.4 is HL-only.
- Current paper marks, case-study assessment differences, approved language conventions and IA rubric remain blocked pending the current guide.
- The project is a school-supervised working computational solution with requirements, tests and user evaluation, never an invented written replacement.

Computing systems, networks and requirements

Official-unit focus: A.1 Computer fundamentals

A school booking system can calculate correctly and still fail if users lose access or records are disclosed. Success includes the system context.

A computer system combines hardware, software, data and people. A network enables communication using protocols. Requirements should distinguish function from constraints such as availability, security and accessibility.



Separate authentication from authorization. Authentication checks identity; authorization determines permitted actions. A threat model connects a valuable asset, a

exercises.lol exercises.com.cn

possible attack and an appropriate control.

Use a fictitious school dataset to define users and permissions. Draw data flows, compare validation and verification, and specify tests for normal, boundary and invalid input. Do not use real credentials or student records in exercises.

■ Checked worked case

Known: a file contains 12 megabytes, where this example defines one megabyte as one million bytes. Bits = bytes×8 = 12×1,000,000×8 = 96,000,000 bits. At 8,000,000 bits per second, ideal time = size/rate = 12 s, excluding overhead.

■ Common error

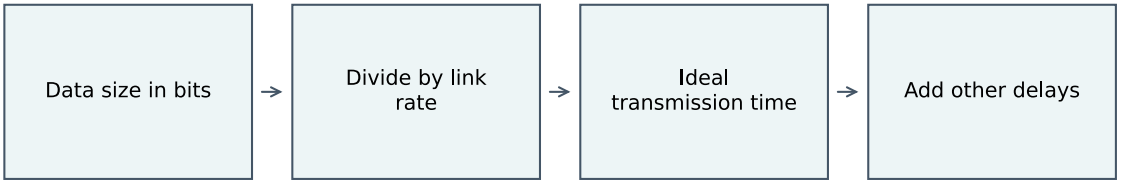
Bandwidth is not actual end-to-end throughput. Encryption does not by itself ensure correct authorization or remove every security risk.

Networks: latency, throughput and layered delivery

Official-unit focus: A.2 Networks

A small message can arrive late on a high-bandwidth link. Capacity and delay measure different properties.

Packets carry addressed data through a network. A layered model separates responsibilities such as application meaning, transport delivery and network routing. Bandwidth describes capacity; throughput is the achieved data rate; latency is delay.



Transmission time depends on data size and rate. Total delay may also include propagation, processing and queueing. Encryption protects content under its assumptions but does not remove congestion or every metadata exposure.

Trace a message route using a documented local model. Record payload size, units and measured time. Use school-approved networks and synthetic messages; do not scan or intercept another user traffic.

exercises.lol exercises.com.cn

■ Checked worked case

Known: an 8 megabit file crosses a 2 megabit/s link. Ideal transmission time = size/rate = $8/2 = 4$ s. Protocol overhead, other users and latency can make observed completion slower.

■ Common error

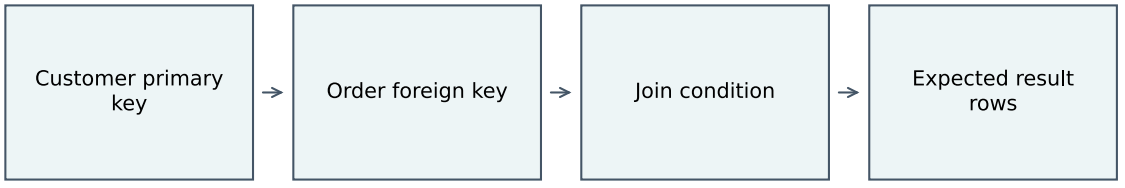
A megabyte is eight megabits before considering overhead. A faster rated link does not guarantee low latency or secure endpoints.

Relational databases: a key and a relationship

Official-unit focus: A.3 Databases

Duplicating a user address in every order can produce conflicting records when the address changes.

A relational table contains rows and attributes. A primary key identifies a row; a foreign key links to a referenced key in another table. Normalization can reduce avoidable duplication and update anomalies while preserving meaningful relationships.



A join combines rows according to a specified condition. The result count depends on relationship cardinality and filters. A foreign key constraint enforces a relationship rule; it does not automatically encrypt personal data.

Design a small synthetic user-order dataset with no real personal information. Declare keys, test duplicate and missing-reference inserts, then check a query result against a hand-worked expected table.

■ Checked worked case

Known: four customers have respectively 2, 0, 3 and 1 orders. An inner join of customer to order returns $2+0+3+1 = 6$ rows. A left join also retains the customer with no order as one null-matched row, giving 7 here.

exercises.lol exercises.com.cn

■ Common error

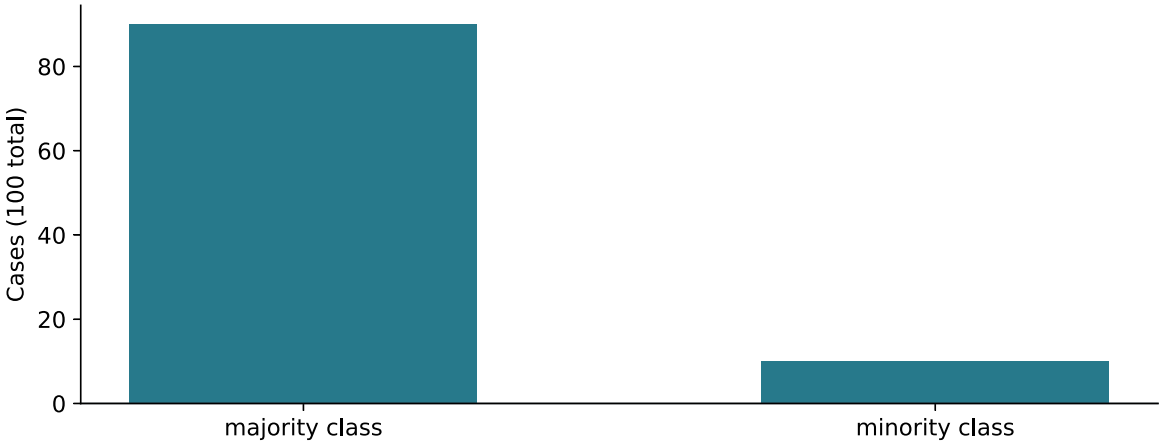
A primary key need not be a person name. A join is not a simple concatenation of tables. Use parameterized queries for untrusted input rather than building SQL from raw strings.

Data models and responsible machine learning

Official-unit focus: A.4 Machine learning

A model can score well by learning a shortcut in the training data. Evaluation must test the intended task on appropriate unseen examples.

A relational database uses tables, keys and relationships to organize data. Machine learning estimates patterns from data. Training and evaluation data serve different roles.



A primary key uniquely identifies a record; a foreign key relates records. In prediction, leakage can expose information unavailable at the real decision time. Accuracy alone may hide an unbalanced target distribution.

Use fictional booking records to identify entities, attributes and relationships. For a learning exercise, use a public non-sensitive dataset, separate training and test data, and describe who may be affected by errors. The current 2027 CS objective scope awaits the guide.

■ Checked worked case

Known: a classifier makes 90 correct predictions among 100 cases. Accuracy = correct/total×100 = 90%. If 90 cases belong to one class, always predicting that class gives the same accuracy and can still fail every minority-class case.

■ Common error

High accuracy is not evidence of fairness or causal understanding. A database primary key is not simply whichever field looks important.

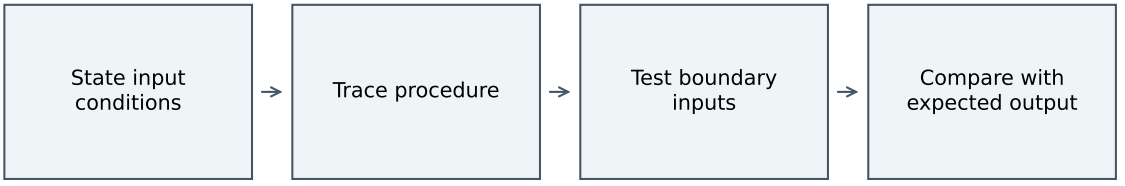
exercises.lol exercises.com.cn

Algorithms, traces and correctness evidence

Official-unit focus: B.1 Computational thinking; B.2 Programming; IA Computational solution

An algorithm can work for one example and fail at a boundary. Testing should be designed from the specification, not only the happy path.

An algorithm is a finite, unambiguous procedure for a task. A trace records state changes. A loop invariant describes a property preserved by each iteration and helps justify correctness.



State input conditions and expected outputs. Use boundary cases, empty collections where allowed, duplicates and invalid values. Distinguish a wrong algorithm from a wrong implementation or an incomplete requirement.

Trace a search over a small fictional sorted list. State the indexing convention. For binary search, update bounds so the remaining interval shrinks and reject unsorted input unless sorting is part of the task.

■ Checked worked case

Known: a linear search of an eight-item list can require eight comparisons when the sought item is last or absent. Doubling the list length doubles the worst-case comparison count under this model. Binary search reduces the interval by roughly half each step but requires a suitable ordered structure.

■ Common error

A successful sample test does not prove correctness for all valid inputs. Do not import Cambridge-specific pseudocode syntax into an IB course without a course source.

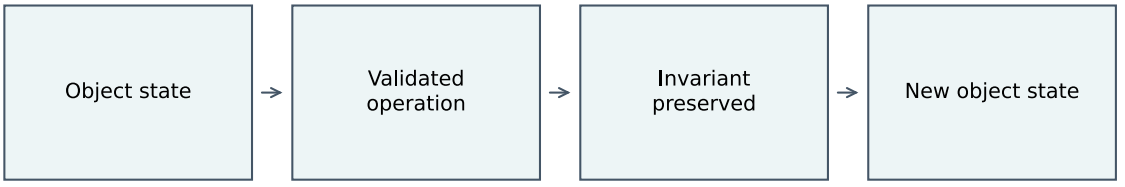
exercises.lol exercises.com.cn

Objects: state, behaviour and an interface

Official-unit focus: B.3 Object-oriented programming

Two bank-account model objects can respond to the same deposit operation while holding different balances.

A class defines a type with state and behaviour. An object is an instance. Encapsulation controls access through an interface so operations can preserve invariants. Inheritance models an appropriate type relationship; composition models an object containing or using another object.



A method call acts on a particular instance. State changes should satisfy preconditions and postconditions. Polymorphism lets code use a common interface with different implementations when the contract is respected.

Implement a small synthetic account or inventory model. Test two independent instances, rejected invalid operations and boundary values. Keep the model away from real financial accounts and credentials.

■ Checked worked case

Known: object A starts at 40 units and receives 15; object B starts at 20 and receives 5. Final balances are 55 and 25. Their total is 80, but neither individual object balance is 80.

■ Common error

A class is not the same thing as one instance. Inheritance is not automatically better than composition. Merely hiding a field does not prove that all operations preserve valid state.

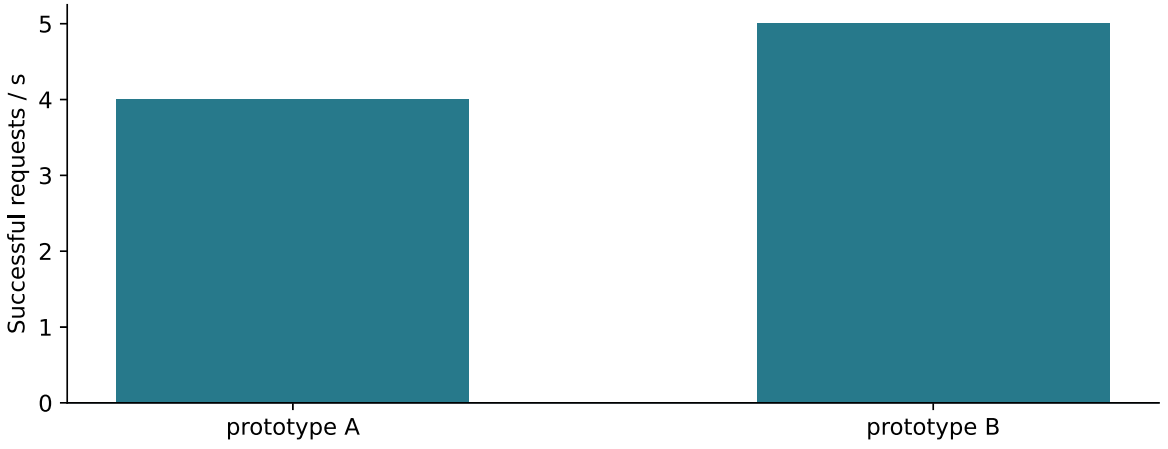
exercises.lol exercises.com.cn

Computing case analysis: evaluate a proposed system

Official-unit focus: Case study Case study

A proposed school booking system looks efficient, but its recommendation depends on users, constraints and evidence. A plausible technical term alone is not a justified decision.

A case study places computational choices within a stated scenario. Identify stakeholders, requirements, available evidence and constraints before recommending an architecture or algorithm. Distinguish a scenario fact from an assumption and a prediction. Trade-offs can involve performance, reliability, accessibility, privacy, maintenance and cost.



A recommendation should connect a requirement to a mechanism and an observable test. For example, a concurrency problem requires a strategy that prevents conflicting updates, plus tests demonstrating the invariant. A claim about faster response requires comparable workload measurements rather than only a complexity label. Evaluate alternatives under the same stated conditions.

Use an original school-approved scenario and synthetic data. Build a matrix of claim, supporting scenario evidence, technical explanation, limitation and acceptance test. Mark missing facts explicitly and show how the recommendation would change if an assumption failed. This prepares case-based reasoning without reproducing an unavailable IB assessment case study or inventing its examination rubric.

■ Checked worked case

Known: a fictional prototype processes 120 successful requests during a 30-second observation. Throughput is 120/30=4 successful requests/s. A second prototype processes 150 in 30 seconds, or 5/s. The second has 25% greater recorded throughput, but no conclusion about latency, failure rate or performance under a larger workload follows without those measurements.

exercises.lol exercises.com.cn

■ Common error

Throughput is not response time. A strong recommendation states the conditions in which it is expected to work and the evidence that could disconfirm it. Current official CS case-study requirements must be checked against the applicable assessment version.

exercises.lol exercises.com.cn