

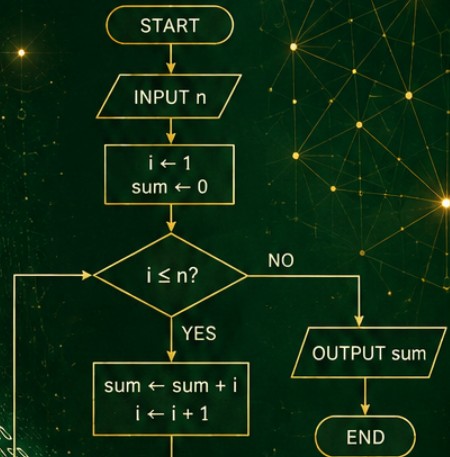
IBDP COMPUTER SCIENCE 100 AI PROMPTS

for Smarter Revision *and* Exam Prep

*Active recall, exam technique, and
mark-scheme thinking -- without cheating.*

CS

```
1 def binary_search(a, target):
2     low = 0
3     high = len(a) - 1
4     while low <= high:
5         mid = (low + high) // 2
6         if a[mid] == target:
7             return mid
8         elif a[mid] < target:
9             low = mid + 1
10        else:
11            high = mid - 1
12    return -1
```



by James R. Martin

© 2026 James R. Martin

All rights reserved.

No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means—electronic, mechanical, photocopying, recording, or otherwise—without prior written permission from the author, except for brief quotations used in reviews.

This book is an independent educational resource and is not affiliated with, endorsed by, or approved by any examination board or awarding organisation.

The author has made use of artificial intelligence tools to assist with drafting, structuring, and generating example material. All educational guidance, explanations, and exam-related advice have been reviewed, edited, and curated by the author. Any resemblance to specific published materials is unintentional.

This book is intended to support revision and exam preparation. It does not replace formal teaching, textbooks, or official specifications. Students are responsible for ensuring that all work submitted for assessment is their own.

ISBN: [TO BE ASSIGNED]

First published 2026

How to Use This Book

For a long time, high-quality tutoring has been a major contributor to elite academic achievement. Used well, AI can now act as a powerful tutor that most students and parents could not previously afford.

This book is a **starting point**, not a rulebook. Each prompt is designed to help you revise, test your understanding, and think more clearly — not to give perfect answers. You are encouraged to **adapt, improve, and remix** these prompts.

You are learning how to think carefully about the questions you ask — a skill that will matter far beyond these exams.

Note on Exam Boards and Syllabi

This guide is designed for students following the International Baccalaureate Diploma Programme (IBDP) Computer Science course. The course is assessed through Paper 1 (short answer and structured response on the core syllabus), Paper 2 (object-oriented programming using a case study), and an Internal Assessment (IA) consisting of a computational solution to a problem. SL students sit Papers 1 and 2 and complete the IA, while HL students cover additional topics including abstract data structures, further resource management, and control systems.

Both Standard Level (SL) and Higher Level (HL) students will benefit from these prompts. SL-specific content covers system fundamentals, computer organisation, networks, and computational thinking. HL-only topics such as abstract data structures (linked lists, stacks, queues, trees, recursion) and advanced resource management are clearly indicated where relevant.

The IB uses specific command terms that students must understand and respond to precisely. In Computer Science, terms such as 'outline', 'describe', 'explain', 'analyse', 'evaluate', 'construct', 'design', and 'distinguish' each require a distinct approach. Many prompts incorporate these command terms to build fluency in exam-appropriate responses.

IB Computer Science is global in its applicability, addressing the social, ethical, and technological dimensions of computing across different cultural and geographic contexts. Students must consider issues such as digital access, data privacy, environmental sustainability, and the societal impact

of emerging technologies in their responses and IA work.

The Internal Assessment requires students to develop a computational solution for a real client. It is assessed on planning, design, development, testing, and evaluation criteria, and it accounts for a significant portion of the overall grade. Several prompts in this guide support IA planning, development, and reflective evaluation.

Contents

How to Use This Book	ii
Note on Exam Boards and Syllabi	iii
• System Fundamentals: System Design and Planning Prompts 1-12	1
• Computer Organisation: Architecture, Logic Gates, and Truth Tables Prompts 13-23	8
• Networks: Protocols, the Internet, and Security Prompts 24-34	14
• Computational Thinking and Problem-Solving: Algorithms and Pseudocode Prompts 35-45	20
• Abstract Data Structures (HL) Prompts 46-56	26
• Resource Management: Operating Systems and Memory Prompts 57-67	32
• Control Systems and Modelling Prompts 68-78	38
• Software Development: SDLC, Testing, and OOP Prompts 79-89	44
• Databases and the Internal Assessment Prompts 90-100	50
Final Closing Note	57
Using AI Beyond This Book	58
About the Author	59
Other Titles in This Series	60

Section 1

System Fundamentals: System Design and Planning

System fundamentals cover the planning, design, and management of computer systems, including system life cycle stages, change management, and the role of stakeholders. This topic provides the foundation for understanding how IT systems are conceived, developed, and maintained.

Key concepts include the systems development life cycle (SDLC), user requirements, system specifications, feasibility studies, and the challenges of transitioning from old to new systems. Students must understand both technical and human dimensions of system implementation.

These prompts develop from basic recall of system planning concepts through to evaluating the success of real-world system implementations, building the analytical depth expected in IB Paper 1 responses.

Prompt 1: Stages of the system life cycle

Copy this prompt into your AI tool:

List and describe the main stages of the systems development life cycle, including feasibility study, requirements analysis, system design, development and coding, testing with alpha and beta phases, implementation using an appropriate changeover method, and ongoing maintenance. For each stage, explain what key activities occur, what deliverables are produced, and how stakeholders contribute to ensuring functional requirements are met throughout the iterative process.

What this helps you practise:

Recalling and describing the stages of the SDLC with appropriate detail

How to use it well:

Ensure you can name all stages in order and describe each concisely. Paper 1 questions often ask you to outline or describe specific stages.

Prompt 2: Stakeholder roles in system planning

Copy this prompt into your AI tool:

Identify the key stakeholders involved in planning a new IT system for a school, including administrators, teachers, students, IT support staff, and parents.

Explain the role each stakeholder plays in defining requirements, and analyse how conflicting priorities between these groups might be resolved during the planning phase using structured interviews or focus groups.

What this helps you practise:

Identifying stakeholders and analysing conflicting requirements

How to use it well:

Think beyond technical users. IB questions expect you to consider administrators, end users, developers, and management perspectives.

Prompt 3: Methods of obtaining requirements

Copy this prompt into your AI tool:

Compare three methods of obtaining user requirements for a new system, such as interviews, surveys, and direct observation of existing workflows. Evaluate the strengths and limitations of each method, explaining which approach would be most appropriate when designing a student records database for a large secondary school with diverse user needs. At HL, extend your response to cover

recursive implementations and their time complexity analysis.

What this helps you practise:

Comparing requirements-gathering techniques with their trade-offs

How to use it well:

Link each method to specific situations. Paper 1 may ask you to recommend a method for a given scenario and justify your choice.

Prompt 4: System changeover methods

Copy this prompt into your AI tool:

Quiz me on changeover methods: ask me to explain the differences between direct, parallel, phased, and pilot changeover strategies, providing a specific organisational scenario where each method would be most suitable. Probe my understanding of the risks associated with each approach, including potential data loss, system downtime, and staff resistance to change.

What this helps you practise:

Evaluating changeover methods in context

How to use it well:

Application to a specific scenario is key. IB examiners reward responses that justify a recommendation rather than simply listing methods.

Prompt 5: Change management challenges

Copy this prompt into your AI tool:

Analyse the challenges an organisation might face when introducing a new network-based inventory management system to replace a legacy paper-based process. Examine how resistance to change among long-serving employees can be managed through structured training programmes, pilot testing with representative user groups, phased rollout schedules, and clear communication of operational

benefits to all affected stakeholder groups including warehouse staff and management.

What this helps you practise:

Analysing human and organisational challenges in IT system implementation

How to use it well:

Consider user resistance, training needs, data migration, and downtime. The IB expects awareness of both technical and human factors.

Prompt 6: Prototyping and iterative design

Copy this prompt into your AI tool:

Test me on prototyping: ask me to describe the role of prototyping in the system development process for a web-based application, explain how iterative design cycles with user feedback improve the final product, and evaluate whether rapid application development methodologies are more effective than traditional waterfall approaches for client-facing software projects.

What this helps you practise:

Comparing iterative and sequential design methodologies

How to use it well:

Be ready to discuss the trade-offs between flexibility and structure. IB questions may ask you to recommend an approach for a given project.

Prompt 7: User documentation and training

Copy this prompt into your AI tool:

Discuss the importance of user documentation and training when deploying a new database management system within a hospital setting.

Analyse the consequences that arise from inadequate documentation, considering scenarios where medical staff must access patient records efficiently under time pressure, and outline specific

strategies for creating effective technical reference manuals, interactive help systems, and structured hands-on training workshops for different user skill levels.

What this helps you practise:

Explaining the role of documentation and training in successful system deployment

How to use it well:

Connect documentation to usability and adoption. Paper 1 questions on implementation often include documentation as a component.

Prompt 8: Social and ethical considerations in system design

Copy this prompt into your AI tool:

Evaluate the social and ethical considerations that should be addressed when designing a new biometric data-collection system for a school, including privacy of student facial recognition data, informed consent requirements, data protection legislation compliance, potential for surveillance overreach, and the responsibilities of system designers to balance security needs against individual civil liberties.

What this helps you practise:

Evaluating ethical dimensions of system design decisions

How to use it well:

The IB places significant emphasis on social and ethical issues. Be prepared to discuss these in relation to any system design scenario.

Prompt 9: System backup and recovery

Copy this prompt into your AI tool:

Present an IB examination-style question about system backup strategies: ask me to explain why incremental, differential, and full backup procedures

are essential components of disaster recovery planning. Probe my understanding of recovery point objectives, off-site storage using cloud infrastructure, redundancy through RAID configurations, and how organisations determine appropriate backup frequency for mission-critical databases.

What this helps you practise:

Understanding backup strategies and their role in system resilience

How to use it well:

Include full, incremental, and differential backups. Link each to a realistic organisational context for stronger Paper 1 responses.

Prompt 10: Evaluating system success

Copy this prompt into your AI tool:

Evaluate whether a newly implemented online learning management system has been successful by asking me to define measurable success criteria, including user adoption rates, system uptime percentage, response time benchmarks, and stakeholder satisfaction scores. Probe how post-implementation review processes using quantitative metrics and qualitative user feedback determine whether the system meets its original requirements specification.

What this helps you practise:

Defining and applying success criteria for system evaluation

How to use it well:

Consider functionality, usability, efficiency, and stakeholder satisfaction. Evaluation questions test your ability to apply criteria and make judgements.

Prompt 11: Legacy systems and migration

Copy this prompt into your AI tool:

Examine the challenges associated with migrating structured and unstructured data from a legacy relational database system to a modern cloud-based platform. Analyse the risks involved in data mapping, format conversion, ensuring referential integrity during transfer, and maintaining business continuity throughout the migration process, including rollback strategies if critical errors are detected.

What this helps you practise:

Analysing data migration risks and mitigation strategies

How to use it well:

Consider data integrity, compatibility, downtime, and user disruption. This topic connects system fundamentals to real-world IT challenges.

Prompt 12: Feasibility studies

Copy this prompt into your AI tool:

Ask me to explain the purpose of a feasibility study in the system planning process for a proposed e-commerce platform, covering technical feasibility of the server infrastructure, economic feasibility including total cost of ownership calculations, legal feasibility regarding data protection regulations, operational feasibility considering staff capabilities, and schedule feasibility for meeting the project delivery timeline.

What this helps you practise:

Evaluating feasibility factors that determine whether a system project should proceed

How to use it well:

Paper 1 may present a scenario and ask whether a proposed system is feasible. Practise weighing multiple feasibility dimensions to reach a justified recommendation.

Section 2

Computer Organisation: Architecture, Logic Gates, and Truth Tables

Computer organisation covers the internal architecture of computer systems, including the CPU, memory hierarchy, machine instruction cycle, and binary representation. Students must understand how hardware components work together to execute programs.

Logic gates and truth tables form the foundation of digital circuit design. Students learn to construct and interpret logic diagrams using AND, OR, NOT, NAND, NOR, and XOR gates, and to verify circuit behaviour using truth tables.

These prompts progress from identifying hardware components and constructing basic truth tables through to analysing the design decisions behind computer architectures and evaluating performance trade-offs.

Prompt 13: CPU components and functions

Copy this prompt into your AI tool:

Identify the main components of a central processing unit, including the arithmetic logic unit, control unit, registers such as the program counter and memory address register, and the system bus architecture. Explain the function of each component within the fetch-decode-execute cycle, and analyse how these components interact during the execution of a machine-level instruction.

What this helps you practise:

Describing CPU architecture and the machine instruction cycle

How to use it well:

Know the ALU, control unit, registers, and buses.
Paper 1 frequently tests knowledge of CPU components and the fetch-decode-execute cycle.

Prompt 14: Primary and secondary memory

Copy this prompt into your AI tool:

Distinguish between primary memory types including random access memory and read-only memory, and secondary storage devices such as solid-state drives, hard disk drives, and optical media. Explain why a memory hierarchy incorporating cache, RAM, and virtual memory is necessary, and analyse how access speed, volatility, and cost per gigabyte influence system design decisions.

What this helps you practise:

Explaining the memory hierarchy and its performance implications

How to use it well:

Include cache memory in your discussion. IB questions may ask you to explain why different types of memory are needed.

Prompt 15: Binary representation

Copy this prompt into your AI tool:

Convert the following values between binary, denary, and hexadecimal number systems: the binary number 10110011, the denary value 175, and the hexadecimal value 2F. Show your working for each conversion step, explain why hexadecimal representation is preferred by programmers when examining memory addresses, and demonstrate how two's complement represents negative integers in an eight-bit system.

What this helps you practise:

Performing number base conversions accurately

How to use it well:

Practise conversions until they are automatic. Paper 1 includes calculation questions where accuracy is essential for full marks.

Prompt 16: Logic gates: AND, OR, NOT

Copy this prompt into your AI tool:

Draw the standard logic gate symbols for AND, OR, and NOT gates and construct the truth table for each gate showing all possible input combinations.

Then design a combinational logic circuit that implements the Boolean expression $Q \text{ equals } A \text{ AND } (B \text{ OR NOT } C)$, verifying your design by constructing the complete truth table for all eight input combinations. SL students should focus on core concepts, while HL students should also address object-oriented design patterns and UML representations.

What this helps you practise:

Constructing logic circuits and truth tables from Boolean expressions

How to use it well:

Be able to move between Boolean expressions, circuit diagrams, and truth tables. Paper 1 may present any of these and ask for the others.

Prompt 17: NAND, NOR, and XOR gates

Copy this prompt into your AI tool:

Quiz me on advanced logic gates: ask me to explain the behaviour of NAND, NOR, and XOR gates by constructing truth tables for each. Then challenge me to justify why NAND gates are considered universal gates, demonstrate how any Boolean function can be constructed using only NAND gates, and simplify a given Boolean expression using De Morgan's theorem.

What this helps you practise:

Understanding advanced logic gates and the concept of universality

How to use it well:

Know how NAND can replicate other gates. This is a common Paper 1 extension question that rewards deeper understanding.

Prompt 18: Constructing circuits from specifications

Copy this prompt into your AI tool:

A security system should activate an alarm when either sensor A or sensor B detects motion AND the system is armed via input C. Design the combinational logic circuit using AND, OR, and NOT gates, write the Boolean expression, construct the complete truth table for all eight input combinations, and then implement the same function using only NAND gates to demonstrate gate universality.

What this helps you practise:

Translating real-world requirements into logic circuits and truth tables

How to use it well:

Paper 1 often presents scenario-based logic questions. Practise translating plain-language conditions into Boolean expressions first.

Prompt 19: Data representation: text and images

Copy this prompt into your AI tool:

Test me on data representation: ask me to explain how text characters are encoded using ASCII and Unicode standards, how bitmap images are represented through pixel grids with colour depth and resolution parameters, and how increasing colour depth from eight-bit to twenty-four-bit affects file size. Probe my understanding of how sampling

rate and bit depth determine the quality and storage requirements of digital sound files.

What this helps you practise:

Explaining binary representation of text (ASCII/Unicode) and images (resolution, colour depth)

How to use it well:

Link resolution, colour depth, and file size quantitatively. IB questions may ask you to calculate file sizes based on given parameters.

Prompt 20: Operating system functions

Copy this prompt into your AI tool:

Outline the key functions of an operating system, including process scheduling using algorithms such as round-robin and priority-based scheduling, memory management through paging and virtual memory allocation, peripheral device management via device drivers, and file system organisation. Analyse how the OS manages competing resource demands when multiple applications run concurrently on a multi-core processor system.

What this helps you practise:

Describing the resource management role of operating systems

How to use it well:

Cover memory management, process scheduling, and I/O management. Paper 1 expects clear, structured descriptions of OS functions.

Prompt 21: Comparing architectures

Copy this prompt into your AI tool:

Compare the von Neumann architecture, which uses a single shared bus for data and instructions, with the Harvard architecture, which employs separate buses and memory spaces. Analyse the advantages of each approach, explain why embedded systems

and digital signal processors often use Harvard architecture, and evaluate how modern processors incorporate elements of both designs through modified Harvard approaches.

What this helps you practise:

Comparing computer architectures and their trade-offs

How to use it well:

Understand the key difference regarding shared versus separate memory for data and instructions. This is a standard Paper 1 comparison question.

Prompt 22: Performance factors

Copy this prompt into your AI tool:

Analyse the factors that affect the performance of a computer system, including processor clock speed measured in gigahertz, number of execution cores, cache memory size and levels, bus width and transfer rates, and the type of storage device used. Evaluate how upgrading each component would affect overall system performance for specific workloads such as video rendering versus database queries.

What this helps you practise:

Evaluating the interplay of hardware factors in system performance

How to use it well:

Avoid treating each factor in isolation. IB examiners reward responses that discuss how factors interact to determine overall performance.

Prompt 23: Emerging hardware technologies

Copy this prompt into your AI tool:

Evaluate the potential impact of quantum computing on future computing capabilities, analysing how quantum superposition and entanglement could revolutionise cryptographic security and

optimisation problems. Consider the TOK question of whether mathematical models in quantum mechanics constitute genuine knowledge of physical reality or merely useful predictive tools, and discuss the practical barriers including qubit stability and decoherence that currently prevent widespread deployment.

What this helps you practise:

Assessing the implications of emerging technologies for computing

How to use it well:

Stay current with technology trends. Paper 1 may include forward-looking questions that require you to evaluate new developments.

Section 3

Networks: Protocols, the Internet, and Security

Networking covers the fundamentals of how computer systems communicate, including network types, protocols, the internet, and security. Students must understand how data is transmitted, routed, and protected across networks.

Key concepts include network topologies, the OSI and TCP/IP models, common protocols (HTTP, FTP, SMTP, DNS), and the mechanisms that protect data in transit. Security topics include encryption, firewalls, authentication, and common threats.

These prompts progress from describing network components and protocols through to evaluating security strategies and analysing the social implications of networked systems.

Prompt 24: Network types and topologies

Copy this prompt into your AI tool:

Distinguish between local area networks, wide area networks, wireless local area networks, and virtual private networks, explaining the typical scale, transmission media, and use case for each. Describe the star and mesh network topologies using labelled diagrams, and evaluate the advantages and disadvantages of each topology in terms of fault tolerance, scalability, installation cost, and data throughput.

What this helps you practise:

Classifying network types and comparing topologies

How to use it well:

Be precise with definitions. Paper 1 questions often ask you to distinguish between network types or recommend a topology for a given scenario.

Prompt 25: Network protocols

Copy this prompt into your AI tool:

Present an IB examination-style question about network protocols: ask me to explain the role of protocols such as TCP, IP, HTTP, HTTPS, FTP, and SMTP in enabling standardised network communication. Probe my understanding of how protocol layering in the TCP/IP model ensures interoperability, and challenge me to explain why protocols like DHCP and DNS are essential for automatic network configuration and domain name resolution.

What this helps you practise:

Describing common network protocols and their applications

How to use it well:

Link each protocol to a real-world use case. Paper 1 may ask you to identify which protocol is appropriate for a given scenario.

Prompt 26: The TCP/IP model

Copy this prompt into your AI tool:

Outline the four layers of the TCP/IP model, specifically the application layer handling HTTP and FTP requests, the transport layer managing TCP and UDP connections, the internet layer routing IP packets, and the network access layer controlling physical transmission. Explain how data is encapsulated at each layer during transmission and decapsulated upon receipt by the destination host.

What this helps you practise:

Describing the TCP/IP layered model and data transmission process

How to use it well:

Be able to trace data through the layers with a

concrete example. This is a frequently examined topic in Paper 1.

Prompt 27: Packet switching and data transmission

Copy this prompt into your AI tool:

Challenge me to explain packet switching: ask me how data is divided into packets with headers containing source and destination IP addresses, sequence numbers, and error-checking checksums, then transmitted across the internet through multiple routers. Probe my understanding of how packets may take different routes and be reassembled at the destination, and compare packet switching with circuit switching for voice communication.

What this helps you practise:

Explaining packet-based data transmission and routing

How to use it well:

Include concepts of packet headers, routing tables, and reassembly. Paper 1 expects you to explain the process, not just name it.

Prompt 28: Wireless networking

Copy this prompt into your AI tool:

Compare wired Ethernet and wireless IEEE 802.11 network connections in terms of bandwidth, latency, reliability, mobility, and installation cost. Analyse the security implications of wireless networking, including vulnerabilities to packet sniffing and unauthorised access, and evaluate how WPA3 encryption, MAC address filtering, and hidden SSIDs contribute to securing a wireless campus network. For HL, include discussion of abstract data types and their formal specifications alongside practical implementations.

What this helps you practise:

Evaluating the trade-offs and security challenges of wireless networks

How to use it well:

Consider speed, convenience, interference, and encryption standards. IB questions may ask you to recommend a network type for a specific context.

Prompt 29: Network security threats

Copy this prompt into your AI tool:

Identify and explain four common network security threats: malware including ransomware and trojans, social engineering attacks such as phishing and spear-phishing, distributed denial of service attacks that overwhelm server resources, and man-in-the-middle attacks that intercept unencrypted communications. For each threat, describe a specific countermeasure and evaluate its effectiveness in protecting organisational data.

What this helps you practise:

Identifying and explaining common cybersecurity threats

How to use it well:

Be specific about how each attack functions. Paper 1 questions test understanding of mechanisms, not just awareness of threat names.

Prompt 30: Encryption and authentication

Copy this prompt into your AI tool:

Ask me to distinguish between symmetric encryption using a shared secret key with algorithms like AES, and asymmetric encryption using public-private key pairs with RSA. Explain how TLS/SSL protocols protect data transmitted over HTTPS connections, and evaluate the role of digital certificates and certificate authorities in authenticating the identity of web servers during secure communication.

What this helps you practise:

Distinguishing encryption methods and explaining their role in security

How to use it well:

Include public/private key concepts and practical examples like HTTPS. Understanding encryption is essential for both Paper 1 and IA security discussions.

Prompt 31: Firewalls and network protection

Copy this prompt into your AI tool:

Present questions on network protection: ask me to describe how stateful packet inspection firewalls analyse incoming traffic against security rules, how proxy servers act as intermediaries to filter web requests and cache content, and how intrusion detection systems monitor network traffic for suspicious patterns. Evaluate which combination of these defences provides the most robust protection for a corporate network.

What this helps you practise:

Explaining network protection mechanisms and their limitations

How to use it well:

Avoid treating firewalls as complete solutions. IB examiners reward awareness that security requires multiple layers of protection.

Prompt 32: Cloud computing and distributed systems

Copy this prompt into your AI tool:

Analyse the advantages and disadvantages of cloud computing models including Infrastructure as a Service, Platform as a Service, and Software as a Service compared to traditional local hosting. Evaluate the security and privacy concerns including data sovereignty, vendor lock-in, service level

agreements, and shared responsibility models that organisations must consider when migrating mission-critical applications to cloud infrastructure.

What this helps you practise:

Evaluating cloud computing trade-offs including security and privacy

How to use it well:

Consider scalability, cost, reliability, and data sovereignty. This topic connects to social and ethical issues emphasised in the IB syllabus.

Prompt 33: Social and ethical issues in networking

Copy this prompt into your AI tool:

Discuss the social and ethical implications of widespread internet access, including the digital divide between developed and developing nations, data privacy concerns arising from mass surveillance and data harvesting by technology companies, issues of online anonymity enabling cyberbullying and misinformation, and the environmental impact of global data centre energy consumption on carbon emissions targets.

What this helps you practise:

Analysing the broader social impact of networked technologies

How to use it well:

The IB places strong emphasis on social and ethical considerations. Be prepared to discuss these issues in Paper 1 and in your IA context.

Prompt 34: Network design for a scenario

Copy this prompt into your AI tool:

A small business with thirty employees across two floors needs a network for file sharing, internet access, and a customer-facing web server. Design the network topology specifying appropriate

hardware including switches, routers, wireless access points, and a firewall. Justify your choice of cabled versus wireless segments, recommend an IP addressing scheme using subnetting, and outline the security measures required to protect customer data.

What this helps you practise:

Applying network knowledge to design a solution for a realistic scenario

How to use it well:

Paper 1 may present scenario-based design questions. Practise justifying each design decision with reference to the requirements and constraints given.

Section 4

Computational Thinking and Problem-Solving: Algorithms and Pseudocode

Computational thinking involves decomposition, pattern recognition, abstraction, and algorithmic design. This topic focuses on the ability to analyse problems, design solutions, and express them using pseudocode and flowcharts.

Students must be able to trace algorithms, construct pseudocode solutions for a range of problems, and analyse the efficiency of different approaches. Standard algorithms such as searching and sorting are core knowledge for both SL and HL.

These prompts develop problem-solving skills from basic algorithm tracing through to designing and evaluating complex solutions, preparing students for Paper 1 algorithmic questions and the IA development process.

Prompt 35: Decomposition and abstraction

Copy this prompt into your AI tool:

Test me on computational thinking: ask me to explain the concepts of decomposition, pattern recognition, and abstraction using the example of designing a school timetable scheduling algorithm.

Probe whether I can identify how breaking the problem into sub-problems such as room allocation, teacher availability, and student option choices simplifies the overall design, and how abstraction removes unnecessary detail.

What this helps you practise:

Applying decomposition and abstraction to problem analysis

How to use it well:

Use a concrete example to demonstrate each concept. Paper 1 may ask you to explain these thinking approaches in context.

Prompt 36: Flowchart construction

Copy this prompt into your AI tool:

Design a detailed flowchart for a program that takes a student's percentage score as input, validates that it falls within the range zero to one hundred, and outputs the corresponding IB grade using the boundaries: seven for eighty percent and above, six for seventy to seventy-nine, and so on down to one. Include decision symbols, input/output shapes, and appropriate error handling for invalid inputs.

What this helps you practise:

Representing algorithmic logic using flowcharts

How to use it well:

Use correct flowchart symbols for processes, decisions, and input/output. Paper 1 may ask you to construct or interpret flowcharts.

Prompt 37: Writing pseudocode: basic structures

Copy this prompt into your AI tool:

Write pseudocode for a program that reads ten integer values from the user into an array, performs input validation to reject non-numeric entries, then calculates and outputs the arithmetic average, the highest value, and the lowest value. Use meaningful variable names, include appropriate loop structures, and add comments explaining each section of your algorithm.

What this helps you practise:

Writing pseudocode using sequence, selection, and iteration

How to use it well:

Follow IB pseudocode conventions. Paper 1 expects you to write clear, well-structured pseudocode that another programmer could follow.

Prompt 38: Tracing an algorithm

Copy this prompt into your AI tool:

Given a pseudocode sorting algorithm, trace through the execution with the input array containing values five, three, eight, one, and four. Show the complete state of all variables and the array contents after each iteration of the outer and inner loops, recording each comparison and swap operation in a structured trace table with clearly labelled columns.

What this helps you practise:

Tracing algorithm execution with specific input values

How to use it well:

Use a trace table to track variable values systematically. Trace questions appear regularly in Paper 1 and are a reliable source of marks.

Prompt 39: Linear and binary search

Copy this prompt into your AI tool:

Compare linear search and binary search algorithms by writing pseudocode for each. Explain the precondition that binary search requires a sorted collection, analyse the time complexity of each algorithm in terms of best-case, worst-case, and average-case scenarios, and evaluate under what conditions linear search might outperform binary search despite its higher theoretical time complexity.

What this helps you practise:

Implementing and comparing standard search algorithms

How to use it well:

Know the prerequisite for binary search (sorted data) and understand time complexity differences. These are frequently examined algorithms.

Prompt 40: Sorting algorithms

Copy this prompt into your AI tool:

Quiz me on sorting algorithms: ask me to describe how bubble sort and selection sort work step by step, trace each algorithm on the array containing seven, two, nine, four, and one, and compare their time complexity in big-O notation. Challenge me to explain why more efficient algorithms like merge sort with $O(n \log n)$ complexity are preferred for large datasets. At SL level, focus on fundamental network architecture, while HL students should extend to protocol stack analysis and subnetting calculations.

What this helps you practise:

Tracing and comparing sorting algorithm performance

How to use it well:

Be able to trace each sort step by step. Paper 1 may ask you to trace a sort or compare algorithms. Accuracy in tracing is essential.

Prompt 41: Arrays and collections

Copy this prompt into your AI tool:

Write pseudocode to manage a collection of student names stored in a one-dimensional array with a maximum capacity of fifty entries. Include operations to add a name at the next available position, search for a name using linear search returning its index, delete a name by shifting subsequent elements, and display all stored names. Implement boundary checking for array overflow and underflow conditions.

What this helps you practise:

Implementing basic data operations on arrays

How to use it well:

Arrays are fundamental to IB Computer Science. Practise common operations until they are fluent, as they underpin many Paper 1 questions.

Prompt 42: String manipulation

Copy this prompt into your AI tool:

Test me on writing pseudocode for a program that takes a string as input, uses a loop to reverse the character sequence, and checks whether the original string matches the reversed version to determine if it is a palindrome. Include handling for case-insensitive comparison, explain the time complexity of your solution, and describe how a stack data structure could provide an alternative approach.

What this helps you practise:

Applying string manipulation techniques in pseudocode

How to use it well:

Know standard string operations (length, substring, character access). Paper 1 may present string-based problems requiring these skills.

Prompt 43: Modular design and sub-programs

Copy this prompt into your AI tool:

Act as an IB examiner: ask me to explain why modular design using functions and procedures is important in programming for code reusability, maintainability, and team collaboration. Then present a monolithic pseudocode solution for a student grade calculator and challenge me to refactor it into well-defined sub-programs with appropriate parameters, return values, and local variable scope.

What this helps you practise:

Refactoring code into modular sub-programs with clear interfaces

How to use it well:

Modularity is assessed in both Paper 1 and the IA. Your IA solution should demonstrate effective use of sub-programs and clear module boundaries.

Prompt 44: Algorithm efficiency and Big O

Copy this prompt into your AI tool:

Challenge me on algorithm efficiency: ask me to explain what is meant by time complexity and space complexity, compare the performance of linear search at $O(n)$ with binary search at $O(\log n)$ and bubble sort at $O(n \text{ squared})$ with merge sort at $O(n \log n)$. Probe my understanding of big-O notation, and ask me to determine the complexity of a given nested loop algorithm.

What this helps you practise:

Understanding and comparing algorithm time complexity

How to use it well:

You do not need to calculate Big O formally, but understanding relative efficiency helps you choose appropriate algorithms in Paper 1 and the IA.

Prompt 45: Designing a solution for a complex problem

Copy this prompt into your AI tool:

Design an algorithmic solution for a library book-lending system that tracks which books are borrowed, records borrower names and due dates, and identifies overdue items. Specify the data structures required, including arrays or linked lists for the book catalogue and a queue for reservation requests. Write pseudocode for the key operations:

borrow, return, search, and generate overdue report.

What this helps you practise:

Applying computational thinking to design a multi-module solution

How to use it well:

This mirrors the type of problem-solving required for the IA. Practise decomposing real-world problems into algorithmic components.

Section 5

Abstract Data Structures (HL)

Abstract data structures are studied at Higher Level and include linked lists, stacks, queues, binary trees, and the concept of recursion. Students must understand how these structures store and organise data and when each is appropriate. [HL only]

Key skills include tracing operations on these data structures, implementing them in pseudocode, and comparing their suitability for different applications. HL Paper 1 questions frequently require students to trace stack and queue operations or traverse a binary tree.

These prompts build from understanding the structure and operations of each data type through to evaluating which structure is most appropriate for a given problem, supporting the full range of HL assessment demands.

Prompt 46: Stacks: structure and operations

Copy this prompt into your AI tool:

Test me on the stack data structure: ask me to describe its last-in-first-out principle and the role of the top pointer, then trace the operations push five, push three, pop, push seven, peek, and pop on an initially empty stack. Challenge me to explain stack overflow and underflow conditions, and to describe how the call stack manages function calls and local variable storage during program execution.

What this helps you practise:

Tracing stack operations and understanding LIFO behaviour

How to use it well:

Trace questions on stacks are common in HL Paper

1. Practise with different sequences of push and pop until you can trace confidently.

Prompt 47: Queues: structure and operations**Copy this prompt into your AI tool:**

Quiz me on queues: ask me to explain the queue data structure and its first-in-first-out principle, then trace the operations enqueue ten, enqueue twenty, dequeue, enqueue thirty, and front on an initially empty queue. Probe my understanding of circular queue implementation using an array with front and rear pointers, and evaluate why circular queues avoid the problem of wasted memory space in linear implementations.

What this helps you practise:

Tracing queue operations and understanding FIFO behaviour

How to use it well:

Be clear on the difference between stacks and queues. Paper 1 may ask you to choose the appropriate structure for a given scenario.

Prompt 48: Linked lists**Copy this prompt into your AI tool:**

Present an IB examination-style question about linked lists: ask me to describe how a singly linked list stores data using nodes containing a data field and a next pointer, how traversal follows the pointer chain from the head node, and how insertion and deletion operations work at the beginning, middle, and end of the list. Challenge me to compare linked list memory allocation with contiguous array storage.

What this helps you practise:

Understanding linked list structure and insertion operations

How to use it well:

Draw diagrams to visualise pointer changes. HL questions may ask you to trace insertions or deletions, so practise both.

Prompt 49: Comparing arrays and linked lists

Copy this prompt into your AI tool:

Compare arrays and linked lists as data storage structures, analysing their differences in memory allocation, access time complexity for random access versus sequential traversal, and efficiency of insertion and deletion operations. Under what circumstances would you choose a linked list over an array, considering scenarios such as implementing a dynamic playlist versus storing fixed-size sensor readings?. SL students should address basic control structures, while HL students must also consider computational complexity and Big-O notation.

What this helps you practise:

Evaluating trade-offs between static and dynamic data structures

How to use it well:

This comparison tests deeper understanding. Paper 1 may present a scenario and ask you to justify your choice of data structure.

Prompt 50: Binary trees: structure and traversal

Copy this prompt into your AI tool:

Draw a binary search tree from the following insertion order: fifteen, eight, twenty-two, four, eleven, nineteen, twenty-seven. Then perform in-order, pre-order, and post-order traversals, listing the nodes visited in each sequence. Explain how the binary search tree property enables $O(\log n)$ search operations, and analyse what happens to

performance when insertions arrive in already sorted order.

What this helps you practise:

Constructing binary search trees and performing tree traversals

How to use it well:

Tree traversal is a high-frequency HL topic. Practise all three traversal methods until they are automatic.

Prompt 51: Applications of stacks and queues

Copy this prompt into your AI tool:

Provide two real-world applications of stacks, such as undo functionality in text editors and browser back-button history, and two applications of queues, such as print job scheduling and customer service ticket systems. For each application, explain why the specific data structure's ordering principle makes it the most appropriate choice and analyse what problems would arise from using the wrong structure.

What this helps you practise:

Connecting abstract data structures to practical applications

How to use it well:

Consider undo functionality, expression evaluation (stacks), and print queues, scheduling (queues). Application questions test understanding beyond definition.

Prompt 52: Recursion: concept and tracing

Copy this prompt into your AI tool:

Challenge me on recursion: ask me to explain what recursion is, identify the two essential components of a recursive function, namely the base case that terminates recursion and the recursive case that reduces the problem size, then trace the execution of a recursive factorial function for factorial of five.

Probe my understanding of how each recursive call creates a new stack frame and how stack overflow can occur with missing base cases.

What this helps you practise:

Understanding and tracing recursive function execution

How to use it well:

Identify the base case and recursive case clearly. Trace questions on recursion appear regularly in HL Paper 1 and require step-by-step accuracy.

Prompt 53: Recursion versus iteration

Copy this prompt into your AI tool:

Compare recursive and iterative approaches to solving the Fibonacci sequence problem by writing pseudocode for both versions. Analyse the advantages and disadvantages of each approach in terms of code readability, memory usage from stack frame allocation, and time complexity, explaining why the naive recursive solution has exponential $O(2^n)$ complexity while the iterative version achieves linear $O(n)$ performance.

What this helps you practise:

Evaluating recursive versus iterative solutions for the same problem

How to use it well:

Understand stack overflow risks and redundant calculations in naive recursion. This comparison demonstrates the evaluative thinking HL expects.

Prompt 54: Dynamic data structures and memory

Copy this prompt into your AI tool:

Present questions on dynamic data structures: ask me to explain how linked lists and binary trees manage memory differently from static arrays through dynamic node allocation and pointer-based

connections. Probe my understanding of heap memory versus stack memory, how garbage collection reclaims unused nodes, and evaluate the trade-offs between the flexibility of dynamic structures and the cache-friendly access patterns of arrays.

What this helps you practise:

Analysing memory management differences between static and dynamic structures

How to use it well:

Consider heap allocation, pointer overhead, and fragmentation. This connects abstract data structures to resource management topics.

Prompt 55: Choosing the right data structure

Copy this prompt into your AI tool:

A music streaming application needs to manage a playlist where users can add songs, remove songs, skip to the next track, and return to the previous track. Analyse which data structure or combination of structures, such as a doubly linked list or a combination of stack and queue, would most efficiently support all these operations. Justify your choice by comparing time complexity for each operation.

What this helps you practise:

Selecting and justifying appropriate data structures for a given problem

How to use it well:

Consider doubly linked lists, stacks, and queues. Justify your choice by matching data structure properties to requirements.

Prompt 56: Implementing a stack or queue in pseudocode

Copy this prompt into your AI tool:

Write pseudocode to implement a stack using an array as the underlying storage, including push, pop, peek, isEmpty, and isFull operations. Implement boundary checking that throws appropriate error messages for stack overflow when pushing to a full stack and stack underflow when popping from an empty stack. Trace your implementation with a sequence of eight operations to verify correctness.

What this helps you practise:

Implementing abstract data structures in pseudocode with error handling

How to use it well:

HL Paper 1 may ask you to write pseudocode for data structure operations. Practise implementations for both stacks and queues.

Section 6

Resource Management: Operating Systems and Memory

Resource management covers how operating systems manage hardware resources including the CPU, memory, storage, and peripheral devices. SL students learn the basics of OS resource management, while HL students explore these concepts in greater depth.

Key topics include process scheduling, memory management techniques (paging, segmentation, virtual memory), disk scheduling, and the role of the OS in multitasking. HL students must also understand resource contention, deadlock, and the limitations of resource management techniques.

These prompts build from describing OS functions through to evaluating the trade-offs in resource management strategies, reflecting the analytical demands of IB Paper 1.

Prompt 57: Roles of the operating system

Copy this prompt into your AI tool:

Outline the main roles of an operating system in managing system resources, including process scheduling using algorithms such as round-robin and shortest job first, memory allocation through paging and segmentation, file system management with directory hierarchies and access permissions, and peripheral device coordination via interrupt handling. Analyse how the OS acts as an intermediary between user applications and hardware.

What this helps you practise:

Describing the resource management functions of an operating system

How to use it well:

Cover process management, memory management, file management, and device management. Paper 1 expects a structured overview of OS roles.

Prompt 58: Process scheduling

Copy this prompt into your AI tool:

Ask me to analyse process scheduling: probe why an operating system needs to schedule processes for efficient CPU utilisation and fair resource allocation, then have me compare two scheduling algorithms such as first-come-first-served and round-robin with time quantum. Challenge me to evaluate which algorithm minimises average waiting time for a given set of processes with specified burst times.

What this helps you practise:

Comparing process scheduling algorithms and their performance trade-offs

How to use it well:

Consider fairness, throughput, and response time. Paper 1 may ask you to recommend a scheduling approach for a given scenario.

Prompt 59: Memory management: paging and segmentation

Copy this prompt into your AI tool:

Distinguish between paging, which divides physical memory into fixed-size frames, and segmentation, which divides logical memory into variable-size segments corresponding to program structures. Analyse what problems each technique solves, including external fragmentation in segmentation and internal fragmentation in paging, and evaluate

how modern operating systems combine both approaches in segmented paging.

What this helps you practise:

Explaining and comparing memory management techniques

How to use it well:

Understand the difference between fixed-size pages and variable-size segments. HL students should be able to discuss these in detail.

Prompt 60: Virtual memory

Copy this prompt into your AI tool:

Test me on virtual memory: ask me to explain how virtual memory uses disk storage to extend available RAM by swapping pages between physical memory and a swap file on secondary storage. Probe my understanding of page tables, translation lookaside buffers, page fault handling, and thrashing that occurs when excessive page swapping degrades system performance below acceptable levels.

What this helps you practise:

Understanding virtual memory and the conditions that cause performance degradation

How to use it well:

Include the role of page tables and swapping. Understanding thrashing demonstrates deeper knowledge that IB examiners reward.

Prompt 61: File management and storage

Copy this prompt into your AI tool:

Present an IB examination-style question about file management: ask me to describe how an operating system manages file storage using hierarchical directory structures, file allocation tables, and access control lists. Probe my understanding of sequential versus random file access methods, how disk fragmentation affects read performance, and

evaluate the benefits of journaling file systems for data integrity. For HL, extend your analysis to include distributed systems concepts and the CAP theorem implications.

What this helps you practise:

Explaining file system management and storage organisation

How to use it well:

Cover contiguous, linked, and indexed allocation. Paper 1 may test your understanding of how files are stored and retrieved.

Prompt 62: Multitasking and concurrency

Copy this prompt into your AI tool:

Quiz me on multitasking: ask me to explain how a single-processor system achieves the appearance of concurrent execution through rapid context switching between processes, storing and restoring process state in process control blocks. Evaluate the overhead costs of context switching, distinguish between pre-emptive and cooperative multitasking, and analyse how multi-core processors enable true parallel execution.

What this helps you practise:

Analysing how operating systems manage concurrent processes

How to use it well:

Consider time-slicing, context switching, and process states. This connects to HL topics on resource contention and deadlock.

Prompt 63: Deadlock

Copy this prompt into your AI tool:

Define deadlock in the context of operating system resource management, where two or more processes are permanently blocked waiting for resources held by each other. Describe the four necessary Coffman

conditions: mutual exclusion, hold and wait, no pre-emption, and circular wait. Evaluate three deadlock prevention strategies and analyse their trade-offs between system throughput and resource utilisation efficiency.

What this helps you practise:

Understanding deadlock conditions and prevention strategies

How to use it well:

Know the four Coffman conditions. HL Paper 1 may present a scenario and ask you to identify whether deadlock is possible.

Prompt 64: Device management and drivers

Copy this prompt into your AI tool:

Act as an IB examiner: ask me to explain the role of device drivers as software interfaces that translate generic operating system commands into device-specific instructions. Probe how the OS communicates with input and output peripherals using interrupt-driven I/O versus polling, and evaluate the security implications of installing third-party device drivers with kernel-level access privileges.

What this helps you practise:

Describing the role of device drivers in hardware-software communication

How to use it well:

Include the concept of hardware abstraction. This topic connects to system fundamentals and troubleshooting scenarios.

Prompt 65: Resource allocation in embedded systems

Copy this prompt into your AI tool:

Analyse how resource management in an embedded system, such as a microcontroller in a washing

machine, differs from a general-purpose desktop computer. Consider the constraints of limited RAM measured in kilobytes, fixed-function firmware, real-time scheduling requirements with deterministic response deadlines, and the absence of virtual memory and secondary storage in many embedded platforms.

What this helps you practise:

Comparing resource management in embedded versus general-purpose systems

How to use it well:

Embedded systems are relevant to control systems topics. Understanding their constraints demonstrates breadth of knowledge.

Prompt 66: Evaluating OS performance

Copy this prompt into your AI tool:

Evaluate the metrics used to assess an operating system's resource management performance, including CPU utilisation percentage, process throughput measured in tasks per second, average turnaround time, average waiting time, and response time for interactive processes. Discuss the trade-offs between maximising throughput and minimising response time, and explain how different workload types prioritise different metrics.

What this helps you practise:

Evaluating OS performance using appropriate metrics and trade-off analysis

How to use it well:

This evaluative question prepares you for higher-level Paper 1 questions. Consider real-world implications of prioritising different metrics.

Prompt 67: Virtualisation and containers

Copy this prompt into your AI tool:

Challenge me on virtualisation: ask me to explain the concepts of hardware virtualisation using hypervisors and containerisation using platforms like Docker. Probe how virtual machines provide complete operating system isolation while containers share the host kernel for lighter-weight deployment. Evaluate the advantages of each approach for server consolidation, application portability, development testing, and disaster recovery scenarios.

What this helps you practise:

Understanding modern resource management through virtualisation

How to use it well:

Link to cloud computing and server management.

This connects resource management to contemporary IT practices discussed in the IB syllabus.

Section 7

Control Systems and Modelling

Control systems use sensors, processors, and actuators to monitor and manage physical processes. Students must understand feedback loops, the role of sensors and actuators, and how computer models simulate real-world systems.

Key concepts include open-loop and closed-loop control, the role of feedback in maintaining system stability, and the use of computer modelling and simulation to predict system behaviour without physical experimentation.

These prompts develop from identifying components of control systems through to evaluating the reliability and limitations of computer models, reflecting the analytical progression required for IB Paper 1.

Prompt 68: Components of a control system

Copy this prompt into your AI tool:

Identify the key components of a computer-controlled system: sensors that convert physical quantities into electrical signals, processors that execute control algorithms, and actuators that produce physical outputs. Using the example of an automated greenhouse climate control system, explain how temperature sensors, humidity sensors, a microprocessor running threshold-based logic, and motorised ventilation actuators work together.

What this helps you practise:

Identifying and explaining the roles of control system components

How to use it well:

Use concrete examples to illustrate each component.

Paper 1 often presents a scenario and asks you to identify components and explain their function.

Prompt 69: Open-loop versus closed-loop control

Copy this prompt into your AI tool:

Distinguish between open-loop control systems, which execute predetermined actions without feedback such as a basic toaster timer, and closed-loop control systems, which continuously monitor output via sensors and adjust accordingly such as a thermostat-controlled heating system. Evaluate why closed-loop systems with negative feedback provide more accurate and reliable control in industrial applications.

What this helps you practise:

Comparing open-loop and closed-loop control with practical examples

How to use it well:

The key difference is feedback. Ensure you can explain how feedback enables a system to self-correct. This is a standard Paper 1 question.

Prompt 70: Feedback mechanisms

Copy this prompt into your AI tool:

Test me on feedback mechanisms: ask me to explain the role of negative feedback in a closed-loop control system where the measured output is compared against a set point to calculate an error signal. Using the example of a cruise control system that adjusts engine throttle based on speed sensor readings, probe my understanding of how proportional control reduces the error signal to maintain stable operation.

What this helps you practise:

Tracing feedback loops in control systems

How to use it well:

Draw a feedback loop diagram to support your explanation. Paper 1 may ask you to trace the operation of a specific control system.

Prompt 71: Sensors and data collection

Copy this prompt into your AI tool:

Quiz me on sensors: ask me to describe four types of sensors commonly used in control systems, specifically temperature sensors using thermistors, light sensors using photodiodes, pressure sensors using piezoelectric transducers, and motion sensors using passive infrared detectors. For each, explain the physical principle of operation, give a practical application, and describe the data output format.

What this helps you practise:

Identifying sensor types and factors affecting measurement accuracy

How to use it well:

Include temperature, pressure, light, and motion sensors. Paper 1 may ask you to select appropriate sensors for a given application.

Prompt 72: Analogue to digital conversion

Copy this prompt into your AI tool:

Construct a detailed examination question about analogue-to-digital conversion: ask me to explain why ADC is necessary in control systems because microprocessors can only process discrete binary values while real-world sensor inputs are continuous analogue signals. Probe my understanding of sampling rate, bit resolution, quantisation error, and how the Nyquist theorem determines the minimum sampling frequency.

What this helps you practise:

Understanding ADC processes and the trade-offs of sampling resolution

How to use it well:

Consider sampling rate and bit depth. This connects control systems to computer organisation topics on data representation.

Prompt 73: Computer modelling and simulation

Copy this prompt into your AI tool:

Ask me to explain the difference between a computer model, which represents a system using mathematical equations and parameters, and a simulation, which runs the model over time to predict behaviour. Evaluate why models are used in weather forecasting, climate science, and traffic flow analysis. Also consider the TOK implications of whether computer models can truly represent reality, given that all models involve simplification reflecting the modeller's assumptions. At HL, also address the ethical dimensions through an ACM Code of Ethics lens and consider autonomous systems.

What this helps you practise:

Distinguishing between models and simulations and explaining their value

How to use it well:

Consider cost, safety, speed, and repeatability. Paper 1 questions on modelling require you to explain both advantages and limitations.

Prompt 74: Limitations of computer models

Copy this prompt into your AI tool:

Evaluate the limitations of computer models in predicting real-world outcomes, considering factors such as incomplete input data from sensors with limited precision, oversimplified mathematical assumptions that ignore non-linear interactions, computational constraints that limit model resolution, and the inherent unpredictability of

chaotic systems. Analyse how model validation against historical data and sensitivity analysis help assess reliability.

What this helps you practise:

Critically evaluating the reliability and limitations of computer models

How to use it well:

Consider data quality, assumptions, and complexity.

Evaluating limitations demonstrates the critical thinking IB values at higher markbands.

Prompt 75: Control systems in everyday life

Copy this prompt into your AI tool:

Identify three examples of control systems used in everyday life outside of industrial settings, such as a domestic air conditioning thermostat, anti-lock braking systems in vehicles, and automatic focus mechanisms in smartphone cameras. For each system, describe the sensor input, processing algorithm, actuator output, and feedback loop, explaining how the control system improves performance compared to manual operation.

What this helps you practise:

Applying control system concepts to familiar real-world contexts

How to use it well:

Consider home appliances, vehicles, and personal devices. Applying theory to familiar contexts demonstrates understanding beyond textbook definitions.

Prompt 76: Autonomous systems and AI control

Copy this prompt into your AI tool:

Discuss the role of artificial intelligence and machine learning algorithms in modern control systems, such as self-driving vehicles using computer vision for object detection and path

planning, or robotic manufacturing arms using reinforcement learning for adaptive motion control.

Reflecting on TOK questions about artificial intelligence and knowledge, evaluate whether AI-controlled systems can demonstrate genuine understanding or merely pattern matching.

What this helps you practise:

Analysing the impact and ethics of AI-driven control systems

How to use it well:

Connect to social and ethical issues in the IB syllabus. Consider safety, accountability, and job displacement alongside technical capabilities.

Prompt 77: Designing a control system solution

Copy this prompt into your AI tool:

Design a control system for an automated irrigation system that monitors soil moisture levels using capacitive sensors and weather forecast data from an API. Specify the sensor types, microprocessor requirements, actuator mechanisms for controlling water valves and pump motors, the control algorithm including threshold values and timing schedules, and the feedback mechanism that prevents overwatering during rainfall.

What this helps you practise:

Designing a complete control system solution for a given problem

How to use it well:

This type of design question integrates multiple concepts. Practise structuring your answer around inputs, processing, outputs, and feedback.

Prompt 78: Safety-critical control systems

Copy this prompt into your AI tool:

Analyse the additional requirements for control systems used in safety-critical applications such as

medical infusion pumps, aircraft autopilot systems, or nuclear reactor monitoring. Evaluate how redundancy through duplicate sensor arrays, fail-safe default states, watchdog timers, formal verification of control software, and rigorous testing standards such as DO-178C ensure system reliability when failure could endanger human life.

What this helps you practise:

Evaluating the design requirements for safety-critical control systems

How to use it well:

Consider redundancy, fail-safes, testing standards, and regulatory requirements. This shows depth of understanding valued in top-markband responses.

Section 8

Software Development: SDLC, Testing, and OOP

Software development covers the processes, methodologies, and programming paradigms used to create reliable software. Students study the software development life cycle (SDLC), testing strategies, and object-oriented programming (OOP) concepts central to Paper 2.

Paper 2 focuses on object-oriented programming and is based on a pre-released case study. Students must understand classes, objects, inheritance, polymorphism, and encapsulation, and be able to apply these concepts to the case study scenario.

These prompts build from fundamental OOP concepts through to designing and evaluating software solutions, preparing students for both Paper 2 and the IA development process.

Prompt 79: Software development methodologies

Copy this prompt into your AI tool:

Compare the waterfall model, which follows sequential phases from requirements through deployment, and agile development methodology, which uses iterative sprints with continuous client feedback and adaptive planning. Evaluate the advantages and disadvantages of each, and analyse for what types of projects, such as safety-critical medical software versus a startup web application, each methodology is most appropriate.

What this helps you practise:

Comparing development methodologies and their suitability for different projects

How to use it well:

Understand the key differences: sequential versus iterative, documentation versus flexibility. Paper 1 may ask you to recommend a methodology.

Prompt 80: OOP fundamentals: classes and objects

Copy this prompt into your AI tool:

Test me on object-oriented programming fundamentals: ask me to explain the relationship between a class as a blueprint defining attributes and methods, and an object as a specific instance of that class with its own state. Using the example of a BankAccount class with attributes for accountNumber, balance, and ownerName, challenge me to define appropriate methods including deposit, withdraw with validation, and getBalance.

What this helps you practise:

Defining classes and creating objects with appropriate attributes and methods

How to use it well:

Paper 2 requires you to work with classes and objects from the case study. Practise defining your own classes to build fluency.

Prompt 81: Encapsulation and access modifiers

Copy this prompt into your AI tool:

Quiz me on encapsulation: ask me to explain the principle of encapsulation in object-oriented programming, why instance attributes should be declared as private and accessed only through public getter and setter methods. Demonstrate with pseudocode for a Student class how data hiding prevents external code from setting invalid values, such as a negative age, by implementing validation within the setter method.

What this helps you practise:

Understanding encapsulation and implementing data hiding

How to use it well:

Encapsulation is a fundamental OOP principle tested in Paper 2. Understand why it matters for data integrity, not just how to implement it.

Prompt 82: Inheritance and class hierarchies

Copy this prompt into your AI tool:

Act as an IB examiner: ask me to describe inheritance in object-oriented programming, where a subclass extends a superclass to inherit its attributes and methods while adding specialised functionality. Explain its benefits for code reuse and hierarchical modelling, and design a class hierarchy for a school system with a base Person class, and derived Student, Teacher, and Administrator subclasses with appropriate attributes. SL students should cover core database concepts, while HL students should extend to normalisation forms and relational algebra.

What this helps you practise:

Designing class hierarchies using inheritance

How to use it well:

Inheritance is heavily tested in Paper 2. Practise identifying is-a relationships and designing appropriate class hierarchies.

Prompt 83: Polymorphism

Copy this prompt into your AI tool:

Challenge me on polymorphism: ask me to explain polymorphism with a practical example where a Shape superclass defines a calculateArea method that is overridden differently in Circle, Rectangle, and Triangle subclasses. Demonstrate how polymorphism allows different object types to

respond to the same method call with behaviour specific to their class, and evaluate why this principle supports extensible software design.

What this helps you practise:

Understanding and applying polymorphism in object-oriented design

How to use it well:

Use method overriding to illustrate polymorphism. Paper 2 may ask you to explain how polymorphism is used in the case study.

Prompt 84: Testing strategies

Copy this prompt into your AI tool:

Formulate an examination-level question about testing strategies: ask me to describe the differences between unit testing of individual methods, integration testing of module interactions, and system testing of the complete application. Include user acceptance testing where the client verifies requirements, and challenge me to explain how test-driven development writes test cases before implementation code to improve software reliability.

What this helps you practise:

Distinguishing testing levels and their purposes

How to use it well:

Testing is assessed in both Paper 1 and the IA. Your IA must include evidence of systematic testing at multiple levels.

Prompt 85: Test data and boundary conditions

Copy this prompt into your AI tool:

Design a comprehensive set of test data for a function that validates user registration ages, which must be between thirteen and one hundred twenty inclusive. Include normal test data within the valid range, boundary values at twelve, thirteen, one hundred twenty, and one hundred twenty-one, and

erroneous data such as negative numbers, zero, extremely large values, and non-integer inputs, explaining the expected output for each case.

What this helps you practise:

Designing comprehensive test data including boundary and erroneous cases

How to use it well:

Boundary testing is a key skill. Paper 1 may ask you to design test data, and your IA should demonstrate thorough testing with varied inputs.

Prompt 86: Debugging techniques

Copy this prompt into your AI tool:

Identify and explain three debugging techniques a programmer might use to locate and fix errors in their code: using breakpoints and stepping through execution line by line in an integrated development environment, inserting print statements to trace variable values at key checkpoints, and using a watch window to monitor how specific variables change during runtime. Give a practical scenario for each technique.

What this helps you practise:

Applying appropriate debugging strategies to different types of errors

How to use it well:

Consider trace tables, breakpoints, and print statements. Debugging skills are essential for the IA development process.

Prompt 87: UML diagrams and design documentation

Copy this prompt into your AI tool:

Test me on UML diagrams: ask me to explain the purpose of UML class diagrams in documenting object-oriented software design, including how classes are represented with three compartments for

class name, attributes with data types, and methods with parameters. Then challenge me to draw a UML class diagram for a library management system showing Book, Member, and Loan classes with associations and multiplicity.

What this helps you practise:

Creating UML diagrams to represent object-oriented designs

How to use it well:

UML diagrams feature in Paper 2 case studies. Practise reading and creating them until you can quickly interpret class relationships.

Prompt 88: User interface design principles

Copy this prompt into your AI tool:

Discuss four principles of effective user interface design: consistency in layout and navigation patterns, visibility of system status through feedback messages and progress indicators, error prevention through input validation and confirmation dialogs, and accessibility compliance for users with visual or motor impairments. Analyse how these principles improve usability and evaluate their application in designing a student portal interface.

What this helps you practise:

Applying UI design principles to evaluate software usability

How to use it well:

Consider consistency, feedback, simplicity, and accessibility. UI design is relevant to both Paper 1 questions and IA development.

Prompt 89: Evaluating software quality

Copy this prompt into your AI tool:

Evaluate the criteria used to assess the quality of a software product, including functional correctness, reliability measured by mean time between failures,

usability assessed through user testing, efficiency in resource usage, and maintainability determined by code documentation quality. Consider the TOK implications of whether quantitative software metrics can truly capture quality, or whether qualitative judgement by experienced developers constitutes a different but equally valid way of knowing.

What this helps you practise:

Evaluating software quality using multiple criteria

How to use it well:

This evaluative perspective prepares you for Paper 1 questions and strengthens the evaluation section of your IA.

Section 9

Databases and the Internal Assessment

This section covers relational database concepts and the Internal Assessment (IA), which requires students to develop a computational solution for a real client. Database knowledge supports both Paper 1 questions and many IA projects that involve data management.

Key database concepts include tables, records, fields, primary and foreign keys, relationships, normalisation, and SQL queries. Students must understand how databases organise, store, and retrieve data efficiently.

The IA prompts support each stage of the project, from client identification and requirements gathering through development, testing, and evaluation, aligned with the IB IA assessment criteria.

Prompt 90: Relational database concepts

Copy this prompt into your AI tool:

Define the following fundamental terms in the context of relational databases: table as a structured collection of related records, record as a single row representing one entity instance, field as a column storing a specific attribute, primary key as a unique identifier for each record, foreign key as a reference establishing relationships between tables, and referential integrity as the constraint ensuring foreign keys reference valid primary keys.

What this helps you practise:

Defining core relational database terminology with practical examples

How to use it well:

Know these definitions precisely. Paper 1 may ask you to define terms or identify keys and relationships in a given schema.

Prompt 91: Database design and normalisation

Copy this prompt into your AI tool:

Quiz me on normalisation: ask me to explain the purpose of normalisation in eliminating data redundancy and update anomalies in database design. Present me with a flat-file table containing student enrolment data with repeating groups, and challenge me to normalise it through first normal form removing repeating groups, second normal form eliminating partial dependencies, and third normal form removing transitive dependencies.

What this helps you practise:

Applying normalisation to eliminate data redundancy

How to use it well:

Understand at least first and second normal form. Paper 1 may present an unnormalised table and ask you to improve the design.

Prompt 92: SQL queries: SELECT and filtering

Copy this prompt into your AI tool:

Write SQL queries to retrieve the following from a school database containing Students, Courses, and Enrolments tables: all students in Year twelve ordered alphabetically, students with a GPA above six point five, the total number of students enrolled in each course using GROUP BY, and students who have not yet enrolled in any course using a LEFT JOIN with a NULL check on the enrolment foreign key.

What this helps you practise:

Writing SQL SELECT queries with filtering conditions

How to use it well:

SQL is tested in Paper 1. Practise writing queries from plain-language descriptions until the syntax is automatic.

Prompt 93: SQL: JOINS and aggregation

Copy this prompt into your AI tool:

Write an SQL query that joins a Students table and a Courses table through an Enrolments linking table to display each student's name alongside the courses they are enrolled in. Then write an aggregate query using COUNT and GROUP BY to show how many students are in each course, filter results using HAVING to show only courses with more than fifteen students, and explain the difference between WHERE and HAVING clauses.

What this helps you practise:

Writing SQL queries involving JOINS and aggregate functions

How to use it well:

JOINS and aggregation are higher-level SQL skills. Paper 1 may require you to query data across related tables.

Prompt 94: Data integrity and validation

Copy this prompt into your AI tool:

Present questions on data integrity: ask me to explain entity integrity ensuring every record has a unique non-null primary key, referential integrity maintaining valid foreign key relationships, and domain integrity restricting field values through data type constraints and validation rules. Challenge me to design validation checks for a customer orders database including range checks, format checks, and presence checks on mandatory fields.

What this helps you practise:

Understanding data integrity mechanisms in database systems

How to use it well:

Consider validation rules, data types, referential integrity, and input masks. These concepts are relevant to both Paper 1 and IA development.

Prompt 95: IA: Identifying a client and problem

Copy this prompt into your AI tool:

Ask me to walk through the process of identifying a suitable client and problem for my Computer Science Internal Assessment solution development. Probe what makes a problem appropriate in terms of computational complexity, whether the client is genuinely accessible for iterative consultation, how I would conduct an initial interview to establish requirements, and how I distinguish between problems suited to a database solution versus an algorithmic solution.

What this helps you practise:

Selecting and scoping an appropriate IA project with a real client

How to use it well:

Choose a client with a genuine problem that requires data processing or management. Avoid overly ambitious projects that cannot be completed within the IA scope.

Prompt 96: IA: Requirements and success criteria

Copy this prompt into your AI tool:

Draft a set of measurable success criteria for your Internal Assessment project based on your client's functional and non-functional requirements. Explain how you ensure these criteria are specific, testable, and directly linked to client needs using the SMART

framework. Analyse how your success criteria will guide the solution development process and form the basis for systematic evaluation during the testing phase of your IA.

What this helps you practise:

Defining measurable success criteria aligned with client requirements

How to use it well:

Vague criteria lead to weak evaluations. Make each criterion testable with a clear pass/fail condition. The IA assessment rewards well-defined criteria.

Prompt 97: IA: Design and planning

Copy this prompt into your AI tool:

Outline the design stage of your Internal Assessment, including the data structures selected such as hash tables or binary search trees, the algorithms planned with pseudocode for key functions, the user interface wireframes showing navigation flow, and the database schema with entity-relationship diagrams. Explain how your design decisions address the specific requirements identified during client consultation and how each choice supports efficient solution development.

What this helps you practise:

Creating a coherent design that connects to defined requirements

How to use it well:

Include flowcharts, class diagrams, or pseudocode as appropriate. The design must clearly lead to your implementation and testing plans.

Prompt 98: IA: Testing and evaluation

Copy this prompt into your AI tool:

Design a comprehensive testing plan for your Internal Assessment that includes unit tests verifying individual functions with boundary and

erroneous data, integration tests confirming modules interact correctly through defined interfaces, and user acceptance testing where your client validates the solution against the agreed success criteria. Explain how you will document test results in a structured table showing input, expected output, actual output, and pass/fail status.

What this helps you practise:

Planning systematic testing and client-based evaluation for the IA

How to use it well:

Testing must be thorough and documented. Include screenshots of test results and client feedback. Weak testing is a common reason for lost IA marks.

Prompt 99: IA: Evaluation and extensibility

Copy this prompt into your AI tool:

Write a critical evaluation of your Internal Assessment project addressing: the extent to which each success criterion was met with supporting evidence, specific feedback obtained from your client through a structured evaluation interview, limitations of your solution including identified edge cases, and recommendations for future extensibility such as adding features like data export, multi-user access, or a web-based interface.

What this helps you practise:

Writing a reflective and honest IA evaluation with client input

How to use it well:

Be honest about limitations. The IB rewards self-aware evaluation over claims of perfection. Include genuine client feedback and realistic improvement suggestions.

Prompt 100: IA: Academic integrity and documentation

Copy this prompt into your AI tool:

Act as an IB examiner: ask me to explain the IB's expectations regarding academic integrity in the Computer Science Internal Assessment, including proper acknowledgement of all external code libraries and online resources, the boundary between acceptable research and plagiarism, how code comments demonstrate understanding of borrowed algorithms, and the requirement to maintain a development log documenting the iterative solution development process.

What this helps you practise:

Understanding and applying academic integrity standards in IA development

How to use it well:

Document all external code with clear attribution. The IB takes academic honesty very seriously, and undocumented use of others' code can result in severe penalties.

Final Closing Note

You have now worked through 100 prompts designed to help you think more clearly, revise more effectively, and prepare more confidently for your GCSE.

Remember: the goal was never to rely on AI for answers. The goal was to use it as a tool to test, challenge, and strengthen your own understanding.

The strongest students are not those who avoid difficulty, but those who engage with it deliberately. Each mistake you identified, each explanation you improved, and each gap you filled has strengthened your thinking.

As you continue your studies, aim to depend less on prompts and more on your own judgement. AI can support you — but your reasoning, clarity, and persistence are what earn marks.

Approach your exams calmly. Think carefully. Write clearly.

You are more prepared than you think.

Using AI Beyond This Book

The prompts in this book are starting points, not final forms.

As you grow more confident, begin modifying them:

- Add constraints (for example, “limit to three key points”).
- Increase difficulty gradually.
- Ask the AI to challenge your reasoning.
- Request alternative explanations.
- Ask it to critique your thinking rather than provide answers.

The most powerful use of AI is not asking it to tell you things — it is asking it to test and refine your thinking.

In the future, those who understand how to use tools intelligently will have an advantage. Treat AI as a tutor, not a shortcut. The skill of asking better questions will continue to matter long after your exams are over.

About the Author

James R. Martin holds an MSci in Physics from the University of Bristol and a PGCE with a Physics focus from the University of Oxford. He has over a decade of experience teaching and tutoring students aged 11-18 across a range of subjects, including Physics, Biology, Chemistry, Mathematics, Economics, and Electronics.

He has worked with multiple syllabi, including GCSE, A-Level, KS3, and the International Baccalaureate Diploma Programme (IBDP), supporting students of varying abilities to develop clarity, confidence, and exam success.

His work focuses on effective revision strategies, independent thinking, and the responsible use of artificial intelligence as a tool to strengthen — not replace — understanding.

Other Titles in This Series

The *100 AI Prompts for Smarter Revision* series supports students across GCSE, A-Level, and IB DP subjects.

GCSE

- English Language
- English Literature
- Mathematics
- Physics
- Biology
- Chemistry
- Geography
- History
- Computer Science
- Economics
- Business Studies
- Religious Studies
- Psychology
- French
- Spanish
- German

A-Level

- Mathematics
- Further Mathematics
- Physics
- Chemistry
- Biology
- Economics
- History
- Geography
- English Literature
- Psychology
- Computer Science

- Politics
- Business

IBDP

- Mathematics: Analysis & Approaches
- Mathematics: Applications & Interpretation
- Physics
- Chemistry
- Biology
- Economics
- Geography
- History
- English A: Literature
- English A: Language & Literature
- Psychology
- Business Management
- Computer Science