Grade Descriptions for Cambridge International AS Level Computer Science 9618

What are Grade Descriptions?

Grade descriptions describe the level of performance typically demonstrated by candidates achieving the different grades awarded for a qualification. For Cambridge International AS Levels, they describe performance at three levels – grades 'e', 'c' and 'a'.

Grade descriptions sit alongside other key documents that illustrate examination standards, including:

- the syllabus, which presents what students should be taught over a course of study and explains how this is assessed
- the specimen assessment materials, which exemplify the structure of the assessment and the kinds of tasks that candidates complete
- grade thresholds, which show the total mark required to achieve a grade.

Grade descriptions are produced with a wide range of audiences in mind. For teachers, they support lesson planning and curriculum development, while students may gain useful insights into what is required to achieve a high grade and what candidate performance at lower grades typically looks like. For university admissions staff and employers, and those less familiar with Cambridge, they paint a picture of typical performance at different grades.

Cambridge publishes grade descriptions for a qualification once examinations have taken place for the first time and we review them when a qualification is substantially revised. They are developed by highly experienced examiners who understand performance standards in the subject area and have studied samples of candidate work.

How do I use this resource?

Grade descriptions are presented as a grid, with content areas at the start of each row and the different grades at the top of each column.

The content areas group together various aspects of the syllabus – they reflect topics, assessment objectives, key concepts, syllabus aims and/or components. The way they are organised is specific to each subject.

For each content area there is a descriptor for each grade. Reading across the row from left to right, the descriptors represent increasing levels of performance, with each grade descriptor building on, and including, the last.

Each column represents overall performance at a particular grade. Reading down the column from top to bottom, the descriptors capture the range of knowledge, understanding and skills that a candidate comfortably achieving the grade is likely to demonstrate.

Cambridge produces grade descriptions to support teaching and learning and the interpretation of candidate scores and grades. We do not use them to set grade thresholds. As such, they cannot be used to challenge the grade awarded to any individual candidate.

Version 1, published September 2023

Grade Descriptions

Area of knowledge, understanding and skills	Typical performance at grade E	Typical performance at grade C	Typical performance at grade A
Basic principles of computer science	Key areas of knowledge: • Digital storage and compression of data, including numerical, text, image and sound • Computer networks, including the transmission of data and the use of IP addresses • Computer hardware, including construction of logic circuits and use of sensors for monitoring and control • Computer architecture and the use of assembly language • System software, including operating systems, language translators and the use of an Integrated Development Environment (IDE)		
	• Students show basic knowledge of key areas, though there are some gaps in this knowledge. • They use binary, convert data to binary and use binary magnitudes. • They identify elements of a computer network and show awareness of IP addressing. • They identify the components of a computer system, including logic gates. • They identify registers and components used for data transfer and show awareness of assembly language programming. • They identify operating system tasks and types of translators and use an IDE.	• Students show good knowledge and some understanding of key areas. • They perform binary addition and subtraction and calculate file sizes. • They describe and know the roles of elements of a computer network and IP addressing. • They describe how components of a computer operate and construct basic logic expressions and circuits. • They describe the purpose of registers, the Fetch-Execute cycle, components used for data transfer, interrupts and assembly language instructions. • They explain operating system tasks and the use of different types of translators and features of an IDE.	• Students show comprehensive knowledge and understanding of key areas. • They show understanding of overflow. • They describe CSMA/CD, bit streaming and different types of IP address. • They explain the operation of monitoring and control systems. • They use register transfer notation and include the use of interrupts in the Fetch-Execute cycle. • They explain operating system tasks and the use of different types of translators and features of an IDE. • They use assembly language for small programs.

Area of knowledge, understanding and skills	Typical performance at grade E	Typical performance at grade C	Typical performance at grade A
Keeping data safe	Key areas of knowledge: • Data security threats and the methods used to protect computer systems • Methods of protecting data integrity, including validation and verification • How to act ethically as a computer professional • Copyright and software licencing • Artificial Intelligence (AI)		
	• Students show basic knowledge of key areas, though there are some gaps in this knowledge. • They identify threats and security methods. • They identify validation and verification methods. • They show awareness of ethical actions, copyright software licensing and AI.	• Students show good knowledge of key areas. • They match appropriate security methods to threats. • They use appropriate validation and verification methods. • They identify whether an act is ethical or unethical. • They describe different types of software licence. • They identify impacts of the use of AI.	• Students show comprehensive knowledge of key areas. • They explain the difference between data security, privacy and integrity. • They explain verification for data entry and transfer. • They explain why an act is ethical or unethical. • They choose an appropriate type of software licence. • They show understanding of the impact of AI.
Management of data	Key areas of knowledge: • Relational databases and their management systems • Data Definition Language (DDL) and Data Manipulation Language (DML)		
	• Students show basic knowledge of key areas, though there are some gaps in this knowledge. • They identify database terminology and complete an entity-relationship (E-R) diagram. • They identify features of a Database Management System (DBMS). • They identify simple Structured Query Language (SQL) commands.	• Students show good knowledge of key areas. • They can use relational database tools and techniques. • They compare using a database with using a file-based approach, draw an E-R diagram and identify the stages of normalisation. • They describe features of a DBMS. • They write simple SQL commands.	• Students show comprehensive knowledge of key areas. • They use relational database tools and techniques and justify why they have used them. • They can complete the stages of normalisation to Third Normal Form (3NF). • They explain when to use features of a DBMS. • They write simple SQL scripts.

Area of knowledge, understanding and skills	Typical performance at grade E	Typical performance at grade C	Typical performance at grade A
Computational thinking skills	Key areas of knowledge: • Abstraction, decomposition and construction of algorithms • Representation of algorithms using flowcharts and structure charts • Use of data types, storage, searching and sorting of data • Programming using pseudocode • The program development lifecycle and its stages		
	• Students show basic knowledge of constructing an algorithm to solve a problem. • They break simple problems down into sub-problems and write a basic algorithm in pseudocode. • They identify data types and the use of files. • They identify some Abstract Data Types (ADTs), including stacks and queues. • They identify the stages of the program development life cycle and different life cycles.	• Students show good knowledge of algorithms. • They can construct an algorithm to solve a problem. • They can break a problem down into sub-tasks using a structure chart. • They write algorithms in pseudocode for: ▪ searching and sorting data held in arrays ▪ handling text files. • They describe the use of stacks, queues and procedures. • They describe each stage of the program development life cycle and different life cycles.	• Students show comprehensive knowledge and understanding of algorithms. • They can construct an efficient algorithm to solve a problem. • They write and correct efficient algorithms in pseudocode, using procedures.