Department for Education

Computer science

Proposed GCSE subject content

May 2024

Contents

Introduction

1. The GCSE subject content sets out the knowledge, understanding and skills common to all GCSE specifications in a given subject. Together with the assessment objectives it provides the framework within which the awarding organisations create the detail of their specifications, so ensuring progression from key stage 3 national curriculum requirements and the possibilities for development into A level.

2. GCSE study of computer science provides the foundational knowledge and understanding of the subject, and experience of applying appropriate knowledge through the discipline of programming. Specifications should be designed to develop both theoretical knowledge and practical understanding, in familiar and unfamiliar contexts. Students should develop an understanding of the role that digital technologies play in transforming how we live, work and interact, the future challenges within the field, their relevance to students' personal use of technology, and the implications of computer science on society.

Subject aims and learning outcomes

3. GCSE specifications in computer science must build on the knowledge, understanding and skills established through the computer science elements of the programme of study for computing at key stage 3, satisfy the computer science elements of computing at key stage 4 and enable students to progress into further learning and/or employment.

4. GCSE specifications in computer science should enable students to:

   a) understand and apply the fundamental principles and concepts of computer science, including abstraction, decomposition, logic, algorithms, and data representation.

   b) analyse problems in computational terms through practical experience of solving such problems, including designing, writing, testing and refining programs.

   c) think computationally, creatively, innovatively, analytically, logically and critically.

   d) understand the components that make up digital systems, and how they communicate with one another and with other systems.

   e) understand the impacts of digital technology to the individual and to wider society.

   f) apply mathematical skills relevant to computer science.

Subject Content

Knowledge and Understanding

5. GCSE specifications in computer science must require students to develop the knowledge and understanding of the fundamentals of computer science as set out below and the ability to apply this knowledge and understanding.

Algorithms, data and programming

- Following, understanding, and designing algorithms to solve problems, including algorithms for searching and sorting.
- Key programming concepts, including:
    - describing programs in terms of inputs, processes and outputs
    - variables and assignment
    - sequence
    - selection (conditionals)
    - iteration, both with finite (count controlled) and with conditional termination
    - the use of subroutines[1] including ones with parameters and how they support modularity and abstraction
- Key data type concepts in programming, including:
    - the concept of data type, including integer, Boolean, real, string, and data structures including arrays
    - Boolean logic using AND, OR and NOT, combinations of these, and the application of logical operators in appropriate truth tables to solve problems
    - representation of numbers in binary and hexadecimal; conversion between these and decimal; binary addition and shifts
    - the digital representation of text, sound, and graphics
- Methods used to test programs to evaluate how well the program meets its requirements, including its effectiveness, robustness, maintainability and the user experience.
- Characteristics and purpose of different levels of programming language, including low-level language.

1 The term “subroutine” includes functions, procedures, and class methods, depending on the particular programming language.

Systems architecture and hardware components

- The characteristics of systems architectures, including:
    - hardware components of typical computer systems
    - CPU architecture
    - the role of CPU and memory in the fetch/decode/execute cycle
    - the purpose and functionality of systems software, including the operating system and utility software
    - data storage methods, including internal memory, physical devices and remote storage, and their strengths and weaknesses in terms of size, capacity, cost, access time, and volatility
    - the calculation of data capacity requirements
    - understanding that embedded systems are computer systems with specialised functions that can be embedded within larger systems, and that they are typically not reprogrammable.

Networking

- Networks and the importance of:
    - connectivity, both wired and wireless
    - how data is transmitted in packets across a network to its destination[2]
    - network security
    - the concept of networking protocols as a set of rules governing how data is communicated between different devices, the purpose of protocols and why they are grouped into layers.

Broader impacts of computing

- Cyber security, including:
    - forms of attack (based on technical weaknesses and behaviour)
    - methods of identifying vulnerabilities
    - ways to protect software systems (during design, creation, testing and use)

2 Understanding of the internal structure of packets or the details of packet routing is not required.

- The broader impacts that digital technology (including artificial intelligence) can have on individuals, wider society, the economy and the environment, including issues of ethics, legality, bias.

Skills

6. GCSE specifications must require students to develop the following skills, applying appropriate knowledge and understanding from (5):

- Use decomposition and abstraction effectively.
    a. to model selected aspects of the external world in a program
    b. to structure programs appropriately into modular parts with clear, well-documented interfaces
- Systematically design[3], write and test and refine programs, in a high-level programming language**[4]**.
- Use logical reasoning and test data to critically evaluate how well a program meets its requirements, including its efficiency, robustness, and the user experience.
- In relation to a given example of a digital technology (real or hypothetical), be able to provide a reasoned evaluation of what impacts and issues it may be expected to have on individuals, wider society, the economy, and the environment.
- Apply computing-related mathematics.

3 A 'systematic design' does not necessarily involve a formal notation; e.g. block diagrams, flow charts, or other graphical notations, natural language.
4 A high level language is one that represents the key programming constructs set out above through an interface that may be visual or text based.

Department
for Education

© Crown copyright 2024

This publication is licensed under the terms of the Open Government Licence v3.0, except where otherwise stated. To view this licence, visit nationalarchives.gov.uk/doc/open-government-licence/version/3.

Where we have identified any third-party copyright information, you will need to obtain permission from the copyright holders concerned.

About this publication:

enquiries https://www.gov.uk/contact-dfe
download www.gov.uk/government/publications

Follow us on Twitter: @educationgovuk
Connect with us on Facebook: facebook.com/educationgovuk