Department for Education

Computer science

GCSE subject content

July 2014

Contents

The content for computer science GCSEs

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.

Subject aims and learning outcomes

2. All specifications in computer science must build on the knowledge, understanding and skills established through the computer science elements of the computing programme of study 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.

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

- understand and apply the fundamental principles and concepts of computer science, including abstraction, decomposition, logic, algorithms, and data representation
- analyse problems in computational terms through practical experience of solving such problems, including designing, writing and debugging programs to do so
- think creatively, innovatively, analytically, logically and critically
- understand the components that make up digital systems, and how they communicate with one another and with other systems
- understand the impacts of digital technology to the individual and to wider society
- apply mathematical skills relevant to computer science

Subject content

Knowledge and understanding

6. GCSE specifications must require learners to develop a knowledge and understanding of the fundamentals of computer science and programming including:

- standard algorithms, including standard searches and sorts
- following and writing algorithms to solve problems including

  - sequence, selection and iteration
  - each of input, processing and output
- the concept of data type, including integer, Boolean, real, character and string, and data structures, including records and one- and two-dimensional arrays
- binary and hexadecimal data representation, addition and how to convert between binary, hexadecimal and denary
- Boolean logic using AND, OR and NOT, combinations of these, and the application of logical operators in appropriate truth tables
- the purpose and functionality of systems software, including the operating system and utility software
- characteristics of systems architectures, including
  - CPU architecture, including Von Neumann and the role of the components of the CPU in the fetch-execute cycle
  - main and secondary storage and ways of storing data on devices including magnetic, optical and solid state
  - data capacity and calculation of data capacity requirements
  - input and output devices
  - hardware components and embedded systems
- characteristics of networks and the importance of
  - types of network, including LAN and WAN
  - connectivity, both wired and wireless
  - common network topologies
  - network security
  - the concept of networking protocols, including Ethernet, Wi-Fi, TCP/IP, HTTP, HTTPS, FTP and email protocols
- the ethical, legal and environmental impacts of digital technology on wider society, including issues of privacy and cybersecurity
- characteristics and purpose of different levels of programming language, including low-level language

Skills

7. GCSE specifications must require learners to develop the following skills:

- to take a systematic approach to problem solving including the use of decomposition and abstraction, and make use of conventions including pseudo code and flowcharts
- to design, write, test and refine programs, using one or more high-level textual programming language(s), either to a specification or to solve a problem
- to explain how particular programs work and evaluate their fitness for purpose in meeting requirements and their efficiency using logical reasoning and test data
- to use abstraction effectively
  - to model selected aspects of the external world in a program
  - to appropriately structure programs into modular parts with clear, well-documented interfaces
- be able to apply computing-related mathematics

Department
for Education

Any enquiries regarding this publication should be sent to us at www.education.gov.uk/contactus.

This document is available to download at www.gov.uk/government/publications.

Follow us on Twitter: @educationgovuk

Like us on Facebook: www.facebook.com/educationgovuk

Reference: DFE-00487-2014