****

**[www.ocr.org.uk/computerscience](http://www.ocr.org.uk/computerscience)**

**Student revision checklist**

Version 1

GCSE (9-1)

***COMPUTER SCIENCE***

**J276**

For first teach in 2016



# Student revision checklist

## Revision checklists

The tables below can be used as a revision checklist.

For more information please see the [OCR GCSE Computer Science specification.](https://www.ocr.org.uk/Images/225975-specification-accredited-gcse-computer-science-j276.pdf)

The table headings are explained below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Assessable learning outcomes**  You will be required to: | **R** | **A** | **G** | **Comments** |
| Here is a list of the learning outcomes for this qualification and the content you need to cover and work on. | You can use the tick boxes to show when you have revised an item and how confident you feel about it.  R = **RED** means you are unsure and lack confidence; you might want to focus your revision here and ask your teacher for help.  A = **AMBER** means you are fairly confident but need some extra practice.  G = **GREEN** means you are very confident.  Concentrate on the **RED** and **AMBER** items and turn them into **GREEN** items.  You might find it helpful to highlight each topic in red, orange or green to help you prioritise. | | | You can use the comments column to:   * add more information about the details for each point * add formulae or notes * include a reference to a useful resource * highlight areas of difficulty or things that you need to talk to your teacher about or look up in a textbook. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Component 1 Computer systems** | | | | |
| * 1. **Systems architecture** | | | | |
| **Learning outcomes**  You will be required to know: | **R** | **A** | **G** | **Comments** |
| 1.1 the purpose of the CPU |  |  |  |  |
| 1.1 Von Neumann architecture:   * MAR (Memory Address Register * MDR (Memory Data Register) * Program Counter * Accumulator. |  |  |  |  |
| 1.1 common CPU components and their function:   * ALU (Arithmetic Logic Unit * CU (Control Unit) * Cache. |  |  |  |  |
| 1.1 the function of the CPU as fetch and execute instructions stored in memory |  |  |  |  |
| 1.1 how common characteristics of CPUs affect their performance:   * clock speed * cache size * number of cores. |  |  |  |  |
| 1.1 embedded systems:   * purpose of embedded systems * examples of embedded systems. |  |  |  |  |
| **Component 1 Computer systems** | | | | |
| * 1. **Memory** | | | | |
| **Learning outcomes**  You will be required to know: | **R** | **A** | **G** | **Comments** |
| 1.2 the difference between RAM and ROM |  |  |  |  |
| 1.2 the purpose of ROM in a computer system |  |  |  |  |
| 1.2 the purpose of RAM in a computer system |  |  |  |  |
| 1.2 the need for virtual memory |  |  |  |  |
| 1.2 flash memory |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Component 1 Computer systems** | | | | |
| * 1. **Storage** | | | | |
| **Learning outcomes**  You will be required to know: | **R** | **A** | **G** | **Comments** |
| 1.3 the need for secondary storage |  |  |  |  |
| 1.3 data capacity and calculation of data capacity requirements |  |  |  |  |
| 1.3 common types of storage:   * optical * magnetic * solid state. |  |  |  |  |
| 1.3 suitable storage devices and storage media for a given application, and the advantages and disadvantages of these, using characteristics:   * capacity * speed * portability * durability * reliability * cost. |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Component 1 Computer systems** | | | | |
| * 1. **Wired and wireless networks** | | | | |
| **Learning outcomes**  You will be required to know: | **R** | **A** | **G** | **Comments** |
| 1.4 types of networks:   * LAN (Local Area Network) * WAN (Wide Area Network). |  |  |  |  |
| 1.4 factors that affect the performance of networks |  |  |  |  |
| 1.4 the different roles of computers in a client- server and a peer-to-peer network |  |  |  |  |
| 1.4 the hardware needed to connect stand-alone computers into a Local Area Network:   * wireless access points * routers/switches * NIC (Network Interface Controller/Card) * Transmission media. |  |  |  |  |
| 1.4 the internet as a worldwide collection of computer networks:   * DNS (Domain Name Server) * hosting * the cloud. |  |  |  |  |
| 1.4 the concept of virtual networks |  |  |  |  |
| **Component 1 Computer systems** | | | | |
| * 1. **Network topologies, protocols and layers** | | | | |
| **Learning outcomes**  You will be required to know: | **R** | **A** | **G** | **Comments** |
| 1.5 star and mesh topologies |  |  |  |  |
| 1.5 Wifi:   * frequency and channels * encryption. |  |  |  |  |
| 1.5 ethernet |  |  |  |  |
| 1.5 the uses of IP addressing, MAC addressing, and protocols including:   * TCP/IP (Transmission Control Protocol/Internet Protocol) * HTTP (Hyper Text Transfer Protocol) * HTTPS (Hyper Text Transfer Protocol Secure) * FTP (File Transfer Protocol) * POP (Post Office Protocol) * IMAP (Internet Message Access Protocol) * SMTP (Simple Mail Transfer Protocol). |  |  |  |  |
| 1.5 the concept of layers |  |  |  |  |
| 1.5 packet switching |  |  |  |  |
| **Component 1 Computer systems** | | | | |
| * 1. **System security** | | | | |
| **Learning outcomes**  You will be required to know: | **R** | **A** | **G** | **Comments** |
| 1.6 forms of attack |  |  |  |  |
| 1.6 threats posed to networks:   * malware * phishing * people as the ‘weak point’ in secure systems (social engineering) * brute force attacks * denial of service attacks * data interception and theft * the concept of SQL injection * poor network policy. |  |  |  |  |
| 1.6 identifying and preventing vulnerabilities:   * penetration testing * network forensics * network policies * anti-malware software * firewalls * user access levels * passwords * encryption. |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Component 1 Computer systems** | | | | |
| * 1. **Systems software** | | | | |
| **Learning outcomes**  You will be required to know: | **R** | **A** | **G** | **Comments** |
| 1.7 the purpose and functionality of systems software |  |  |  |  |
| 1.7 operating systems:   * user interface * memory management/multitasking * peripheral management and drivers * user management * file management. |  |  |  |  |
| 1.7 utility system software:   * encryption software * defragmentation * data compression * the role and methods of backup:   + full   + incremental. |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Component 1 Computer systems** | | | | |
| * 1. **Ethical, legal, cultural and environmental concerns** | | | | |
| **Learning outcomes**  You will be required to know: | **R** | **A** | **G** | **Comments** |
| 1.8 how to investigate and discuss Computer Science technologies while considering:   * ethical issues * legal issues * cultural issues * environmental issues * privacy issues. |  |  |  |  |
| 1.8 how key stakeholders are affected by technologies |  |  |  |  |
| 1.8 environmental impact of Computer Science |  |  |  |  |
| 1.8 cultural implications of Computer Science |  |  |  |  |
| 1.8 open source vs proprietary software |  |  |  |  |
| 1.8 legislation relevant to Computer Science:   * + The Data Protection Act 1998   + Computer Misuse Act 1990   + Copyright Designs and Patents Act 1988   + Creative Commons Licensing   + Freedom of Information Act 2000. |  |  |  |  |
| **Component 2 Computational thinking, algorithms and programming** | | | | |
| * 1. **Algorithms** | | | | |
| **Learning outcomes**  You will be required to know: | **R** | **A** | **G** | **Comments** |
| 2.1 computational thinking:   * abstraction * decomposition * algorithmic thinking. |  |  |  |  |
| 2.1 standard searching algorithms:   * binary search * linear search. |  |  |  |  |
| 2.1 standard sorting algorithms:   * bubble sort * merge sort * insertion sort. |  |  |  |  |
| 2.1 how to produce algorithms using:   * pseudocode * using flow diagrams. |  |  |  |  |
| 2.1 interpret, correct or complete algorithms |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Component 2 Computational thinking, algorithms and programming** | | | | |
| * 1. **Programming techniques** | | | | |
| **Learning outcomes**  You will be required to know: | **R** | **A** | **G** | **Comments** |
| 2.2 the use of variables, constants, operators, inputs, outputs and assignments |  |  |  |  |
| 2.2 use of the three basic programming constructs used to control the flow of a program:   * sequence * selection * iteration (count and condition controlled loops. |  |  |  |  |
| 2.2 the use of basic string manipulation |  |  |  |  |
| 2.2 use of basic file handling operations:   * open * read * write * close. |  |  |  |  |
| 2.2 the use of records to store data |  |  |  |  |
| 2.2 the use of SQL to search for data |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Component 2 Computational thinking, algorithms and programming** | | | | |
| **2.2 Programming techniques** | | | | |
| **Learning outcomes**  You will be required to know: | **R** | **A** | **G** | **Comments** |
| 2.2 the use of arrays (or equivalent) when solving problems, including both one and two dimensional arrays |  |  |  |  |
| 2.2 how to use sub programs (functions and procedures) to produce structured code |  |  |  |  |
| 2.2 the use of data types:   * integer * real * Boolean * character and string * casting. |  |  |  |  |
| 2.2 the common arithmetic operators |  |  |  |  |
| 2.2 the common Boolean operators |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Component 2 Computational thinking, algorithms and programming** | | | | |
| * 1. **Producing robust programs** | | | | |
| **Learning outcomes**  You will be required to know: | **R** | **A** | **G** | **Comments** |
| 2.3 defensive design considerations:   * input sanitisation/validation * planning for contingencies * anticipating misuse * authentication. |  |  |  |  |
| 2.3 maintainability:   * comments * indentation. |  |  |  |  |
| 2.3 the purpose of testing |  |  |  |  |
| 2.3 types of testing:   * iterative * final/terminal. |  |  |  |  |
| 2.3 how to identify syntax and logic errors |  |  |  |  |
| 2.3 selecting and using suitable test data |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Component 2 Computational thinking, algorithms and programming** | | | | |
| * 1. **Computational logic** | | | | |
| **Learning outcomes**  You will be required to know: | **R** | **A** | **G** | **Comments** |
| 2.4 why data is represented in computer systems in binary form |  |  |  |  |
| 2.4 simple logic diagrams using the operations AND, OR and NOT |  |  |  |  |
| 2.4 truth tables |  |  |  |  |
| 2.4 combining Boolean operators using AND, OR and NOT to two levels |  |  |  |  |
| 2.4 applying logical operators in appropriate truth tables to solve problems |  |  |  |  |
| 2.4 applying computing-related mathematics:   * + * – * / * \* * Exponentiation (^) * MOD * DIV. |  |  |  |  |
| **Component 2 Computational thinking, algorithms and programming** | | | | |
| * 1. **Translators and facilities of languages** | | | | |
| **Learning outcomes**  You will be required to know: | **R** | **A** | **G** | **Comments** |
| 2.5 characteristics and purpose of different levels of programming language, including low level languages |  |  |  |  |
| 2.5 the purpose of translators |  |  |  |  |
| 2.5 the characteristics of an assembler, a compiler and an interpreter |  |  |  |  |
| 2.5 common tools and facilities available in an integrated development environment (IDE):   * editors * error diagnostics * run-time environment * translators. |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Component 2 Computational thinking, algorithms and programming** | | | | |
| * 1. **Data representation** | | | | |
| **Learning outcomes**  You will be required to know: | **R** | **A** | **G** | **Comments** |
| 2.6 **Units**   * bit, nibble, byte, kilobyte, megabyte, gigabyte, terabyte, petabyte * how data needs to be converted into a binary format to be processed by a computer. |  |  |  |  |
| 2.6  **Numbers**   * how to convert positive denary whole numbers (0–255) into 8 bit binary numbers and vice versa * how to add two 8 bit binary integers and explain overflow errors which may occur * binary shifts * how to convert positive denary whole numbers (0–255) into 2 digit hexadecimal numbers and vice versa * how to convert from binary to hexadecimal equivalents and vice versa * check digits. |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Component 2 Computational thinking, algorithms and programming** | | | | |
| **2.6 Data representation** | | | | |
| 2.6 **Characters**   * the use of binary codes to represent characters * the term ‘character-set’ * the relationship between the number of bits per character in a character set and the number of characters which can be represented (for example ASCII, extended ASCII and Unicode.) |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Component 2 Computational thinking, algorithms and programming** | | | | |
| * 1. **Data representation** | | | | |
| **Learning outcomes**  You will be required to know: | **R** | **A** | **G** | **Comments** |
| 2.7 **Images**   * how an image is represented as a series of pixels represented in binary * metadata included in the file * the effect of colour depth and resolution on the size of an image file. |  |  |  |  |
| 2.7 **Sound**   * how sound can be sampled and stored in digital form * how sampling intervals and other factors affect the size of a sound file and the quality of its playback: * sample size * bit rate * sampling frequency. |  |  |  |  |
| 2.7 **Compression**   * need for compression * types of compression: * lossy * lossless. |  |  |  |  |

Whether you already offer OCR qualifications, are new to OCR, or are considering switching from your current provider/awarding organisation, you can request more information by completing the Expression of Interest form which can be found here: [www.ocr.org.uk/expression-of-interest](http://www.ocr.org.uk/expression-of-interest)

Looking for a resource? There is now a quick and easy search tool to help find free resources for your qualification:   
[www.ocr.org.uk/i-want-to/find-resources/](http://www.ocr.org.uk/i-want-to/find-resources/)

**OCR Resources**: *the small print*OCR’s resources are provided to support the delivery of OCR qualifications, but in no way constitute an endorsed teaching method that is required by the Board, and the decision to use them lies with the individual teacher. Whilst every effort is made to ensure the accuracy of the content, OCR cannot be held responsible for any errors or omissions within these resources.

Our documents are updated over time. Whilst every effort is made to check all documents, there may be contradictions between published support and the specification, therefore please use the information on the latest specification at all times. Where changes are made to specifications these will be indicated within the document, there will be a new version number indicated, and a summary of the changes. If you do notice a discrepancy between the specification and a resource please contact us at:

[resources.feedback@ocr.org.uk](mailto:resources.feedback@ocr.org.uk).

© OCR 2020 - This resource may be freely copied and distributed, as long as the OCR logo and this message remain intact and OCR is acknowledged as the originator of this work. OCR acknowledges the use of the following content: n/a

Please get in touch if you want to discuss the accessibility of resources we offer to support delivery of our qualifications: [resources.feedback@ocr.org.uk](mailto:resources.feedback@ocr.org.uk)

