

## 2023 Year 10 Computer Science Mock Exam Revision Timetable

**Student Name:** You MUST make revision notes. Remember revision must mean you make your own notes so choose your preferred technique well. Do not just read your notes, this will only help you a remember for a short time and you will not remember everything. It is always best to do something with your notes, rewriting them or testing yourself will help much more.

### Exam

You will have 2 exam papers to do. Each exam will be 45 minutes in length and will take place in the classroom during your normal Computer Science lessons.

Unit 1 – Computer Systems

Unit 2 – Computational thinking, algorithms and programming

### Resources

- Showbie has all the work you have done in class for this unit with lots of links and information.
- [www.erevision.uk](http://www.erevision.uk) – this has many exercises that you can do to check your understanding.

What to revise – a suggested guide	Revised & Tested?	Any Problems?
<b>Week 1 wb 06/03</b> <b>1.1.1 Architecture of CPU</b> <ul style="list-style-type: none"> <li>• The purpose of the CPU</li> <li>• Common CPU components and their function</li> <li>• Von Neumann architecture</li> </ul> <b>1.1.2 CPU Performance</b> <ul style="list-style-type: none"> <li>• How common characteristics of CPUs affect their performance</li> </ul> <b>1.1.3 Embedded Systems</b> <ul style="list-style-type: none"> <li>• The purpose and characteristics of embedded systems</li> <li>• Examples of embedded systems</li> </ul>		
<b>Week 2 wb 13/03</b> <b>1.2.1 Primary Storage Memory</b> <ul style="list-style-type: none"> <li>• The need for primary storage</li> <li>• The difference between RAM and ROM</li> <li>• The purpose of ROM in a computer system</li> <li>• The purpose of RAM in a computer system</li> <li>• Virtual memory</li> </ul> <b>2.2.1 Programming fundamentals</b> <ul style="list-style-type: none"> <li>• Variables, constants, operators, inputs, outputs and assignments.</li> </ul>		
<b>Week 3 wb 20/03</b> <b>1.2.2 Secondary Memory</b> <ul style="list-style-type: none"> <li>• The need for secondary storage</li> <li>• Common types of storage: Optical, Magnetic and Solid state</li> <li>• Suitable storage devices and storage media for a given application</li> <li>• The advantages and disadvantages of different storage devices and storage media relating to these characteristics.</li> </ul> <b>2.2.1 Programming fundamentals</b> <ul style="list-style-type: none"> <li>• The use of the three basic programming constructs used to control the flow of a program: Sequence, Selection and Iteration.</li> </ul>		

<p><b>Week 4 wb 27/03</b></p> <p><b>1.2.3 Units</b></p> <ul style="list-style-type: none"> <li>• The units of data storage</li> <li>• How data needs to be converted into a binary format to be processed by a computer</li> <li>• Data capacity and calculation of data capacity requirements</li> </ul> <p><b>1.2.4 Data Storage Numbers</b></p> <ul style="list-style-type: none"> <li>• How to convert positive denary whole numbers to binary numbers (up to and including 8 bits) and vice versa</li> <li>• How to add two binary integers together (up to and including 8 bits) and explain overflow errors which may occur</li> <li>• How to convert positive denary whole numbers into 2-digit hexadecimal numbers and vice versa</li> <li>• How to convert binary integers to their hexadecimal equivalents and vice versa</li> <li>• Binary shifts</li> </ul> <p><b>2.1.1 Computation Thinking</b></p> <ul style="list-style-type: none"> <li>• Principles of computational thinking: Abstraction, Decomposition and Algorithmic thinking</li> </ul>		
<p><b>Week 5 wb 03/04 – Easter Break</b></p> <p><b>1.2.4 Data Characters</b></p> <ul style="list-style-type: none"> <li>• The use of binary codes to represent characters</li> <li>• The term ‘character set’</li> <li>• The relationship between the number of bits per character in a character set, and the number of characters which can be represented.</li> </ul> <p><b>1.2.4 Data Images</b></p> <ul style="list-style-type: none"> <li>• How an image is represented as a series of pixels, represented in binary • Metadata</li> <li>• The effect of colour depth and resolution.</li> </ul> <p><b>1.2.4 Data Sound</b></p> <ul style="list-style-type: none"> <li>• How sound can be sampled and stored in digital form</li> <li>• The effect of sample rate, duration and bit depth.</li> </ul> <p><b>1.2.5 Compression</b></p> <ul style="list-style-type: none"> <li>• The need for compression</li> <li>• Types of compression – Lossy and lossless</li> </ul> <p><b>2.2.2 Data types</b></p> <ul style="list-style-type: none"> <li>• The use of data types: Integer, Real, Boolean, Character and string.</li> </ul>		
<p><b>Week 6 wb 10/04</b></p> <p><b>2.1.2 Designing, Creating and refining Algorithms.</b></p> <ul style="list-style-type: none"> <li>• Identify the inputs, processes, and outputs for a problem</li> <li>• Structure diagrams</li> <li>• Create, interpret, correct, complete, and refine algorithms using:</li> <li>• Identify common errors</li> </ul>		