CM3106 Revision Guide

Prof David Marshall

dave.marshall@cs.cardiff.ac.uk

and

Dr Kirill Sidorov

K.Sidorov@cs.cf.ac.uk

www.facebook.com/kirill.sidorov

School of Computer Science & Informatics
Cardiff University, UK

CM3106 Multimedia Revision Guide

We will discuss a list of aspects of the course that are **not** going to be examined and parts of the course that **are** likely to be examined.

The best revision aid:

Past exam papers and indicative solutions for some most relevant exam papers are available on the Multimedia website:

`http://www.cs.cf.ac.uk/Dave/Multimedia/EXAMS/`

The best way to practice for the exam is to use these to test yourself and also for the style of exam questions and what the **indicative** solutions expect.

Past Papers

This year's exam paper follows the same patten as almost all past papers:

- There are **4 questions** in total, each worth 25 marks.
- You need to answer **3 out of 4** (max 75 marks).
- Expect the following types of questions:
  1. Basic bookwork definitions.
  2. Bookwork description of theory or algorithms.
  3. Application of known algorithms.
  4. Unseen extended reasoning problems: apply your knowledge of multimedia to a novel problem.

Please note the amount of marks available per parts of questions — as an indicator to the amount of attention you should devote to your solution.

What you DO NOT have to learn

- Basic historical facts as mentioned in the lecture notes. *E.g.* dates of algorithm development, MPEG video history, dates of sythesisers/synthesis methods.
  - **Note**: You are **expected** to note differences between different algorithms that may have been developed from previous versions. *E.g.* differences between LZW and LZ, MPEG-1 (and 2) video and H.261.
  - Other related historical "trivia" related to certain algorithms, synthesis methods are **not required**. *E.g.* When/where sythesis methods first used in hardware/software implementations, development of MPEG video and audio, etc.

What you DO NOT have to learn

- MATLAB.
  - There will be **no** specific exam question on MATLAB programming, graphics or GUI design in the exam.
  - **Note**: Exam questions **may ask** for a suitable algorithmic description. In many cases giving MATLAB code (or fragments) might be the best solution and maps most closely to what has been described in the lecture notes.
    IN SUCH CASES giving MATLAB code as part of your solution is perfectly valid.
    ALTERNATIVELY you may use pseudo code to describe the algorithm as long as it is understandable.

What you DO NOT have to learn

- MATLAB.
  - We would also advise looking at the MATLAB code examples, running them and trying out variations as suggested in lab classes as a way of understanding more deeply almost all aspects of the course.
  - If you missed labs — **do them now**! (Downloadable from the website.)

What you DO NOT have to learn

- Basic calculus based derivations of some examples discussed in lecture notes. *E.g.* FFT of sinc function, convolution integral example will **not** be examined.
- Detailed facts relating to certain areas of the notes will not be required to be memorised:
  - *E.g* exact MIDI commands, General MIDI instruments numbers.
    Note: some idea of how MIDI works **is required** and has been examined in the past.
  - *E.g* exact numbers on formats such PAL/NTSC resolution, number of bits used in various MPEG compression schemes.
  - *E.g* exact JPEG quantisation tables.

What you DO NOT have to learn

- The following topics largely covered in CM0268 will **not** be examined **by themselves**:
  - Signal diagrams.
  - Z-transforms.
  - Filter design (Finite Impulse Response/Infinite Impulse Response).
- However, you **should have** a sufficient understanding of these when they are applied to *e.g.* digital audio synthesis.

What you **DO** have to learn

- All basic definitions, including mathematical definitions. *E.g* Fourier or Discrete Cosine Transform
- Basic descriptions of algorithms for all described in lectures. This may be via pseudo-code and/or diagrams and/or MATLAB code fragments.
- Broad differences between similar algorithms. *E.g.* JPEG/MPEG I-frames, H.261/MPEG video etc.
- How to apply algorithms to encode example data streams — **especially** basic compression algorithms — Huffman coding, arithmetic coding, LZW etc.
- See exam papers for examples of questions.

Calculator

- Please **bring a calculator** to the exam as there might exist some calculations that you can benefit from having a calculator.
- See university regulations of "Use of Calculators".
- The calculators which students may use in examinations must be noiseless, battery or solar-powered, scientific calculators with numeric displays only. Programmable calculators, or calculators with an alphabetic keyboard and/or the ability to store and retrieve text are not permitted in any examination.