Fundamentals of Computing

Michael Zakharyaschev

Department of Computer Science and Information Systems

Birkbeck, University of London

- room 265, Main Building, Malet Street
- email: michael@dcs.bbk.ac.uk, m.zakharyaschev@bbk.ac.uk
- homepage: http://www.dcs.bbk.ac.uk/~michael

- FoC resources (lecture slides, videos, tutorials, exercises, reading list, useful links) are available on Moodle

Module Structure

Duration: 10 weeks, with extra revision sessions in week 11 on demand

Lectures: written lecture notes and pre-recorded videos on Moodle
for studying at home before live tutorials (≈ 1.5 hours per week)

Live tutorials: 1.5 hours per week on campus at BBK or via MS Teams
to give you some hands-on experience in basic computer science
(tutorial sheets and model answers are available on Moodle)

Asynchronous Q & A sessions: send your questions to the module tutors

Homework (not assessed): extra exercises
with model answers published a week later

Assessment: **coursework** (to be issued on 10 November, deadline on 21 November)

+

examination in January (online)

Final mark $=$ **CW** (20%) $+$ **exam** (80%)

Key contacts

The module's teaching staff are happy to help you with any questions about the module material

Lecturer: Prof Michael Zakharyaschev michael@dcs.bbk.ac.uk

Assistants: Dr Dimitrios Airantzis dimitris.airantzis@bbk.ac.uk

Mr Patrick Greaves p.greaves@bbk.ac.uk

Dr Yury Savateev estlladon@gmail.com

However, the lecturer and TAs have no means to influence procedural issues.

In particular, if you have any problems with

- your module registration on Moodle or MS Teams,
- navigating around Moodle or MS Teams,

please contact the MSc administrators at cs-pg@bbk.ac.uk

Recommended weekly FoC workflow

- **Before** each tutorial:
  - Study the week's lecture material (both the slides and the videos)
  - Try to solve the exercises on the tutorial sheet without using the given model answers but consulting the lecture material again if needed
  - Compare your solutions with the model answers. (Bear in mind that maths exercises may have many different correct solutions.)
  - Send us your questions, if any, by email or via Moodle forum
- Attend the weekly live tutorial, where some of the problems will be discussed
- **After** each tutorial:
  - Check again your understanding of the exercises.
  - Try to solve the additional practice work exercises. Your answers will not be marked (so do not send them to us). But detailed model answers will be provided (with a week delay).
- If you have questions, use the **FoC discussion forum** on Moodle or **email**. Detailed answers to your questions will be published within a week.

Learning the material

- You are not supposed to understand **everything** just by 'going through'. The lecture material is prepared to guide you through the chosen topics.
- The **only** way to master maths material is by **doing it yourself**
- It is very important to 'make your hands dirty' and to work through the provided exercises **yourself**. The exercises are carefully chosen to help you understand the introduced concepts and techniques. If you cannot solve a particular exercise, consult the lecture material again, but do not check the model solutions yet. Only check out the model solutions when you have something to compare them with. If after reading the model solutions you still have questions, please use the online forums or send us an email.
- More exercises (with solutions) will be provided on the module site. If you think you need even more, please consult the recommended textbooks. And finally: please **use the Web** (there is an extensive amount of online material, including textbooks and videos, on the covered topics).

What is this module for?

- This module aims to (i) provide a common **mathematical background** for your other modules, and to (ii) bring the **mathematical skills** of the class to the same level.
- The module also aims to develop your problem solving skills and your ability to express yourself in a more precise manner.
- If you had Mathematics A-level, you may find that some parts of the module cover material you already know to some extent (and you might be bored:-)

  But watch out: there will be something new any time
- If you did not have Mathematics A-level: **don't panic!**

  You are not expected to know topics that you have never seen before, everything will be developed from scratch.

  And much of Maths A-level is not relevant to this module anyway (for example, we won't use differentiation or trigonometry).

What is used?

- **integer numbers**

$$\overbrace{\ldots,\ -5,\ -4,\ -3,\ -2,\ -1}^{\text{negative}},\ \underbrace{0,\ \overbrace{1,\ 2,\ 3,\ 4,\ 5,\quad \ldots}^{\text{positive}}}_{\text{natural numbers}}$$

  - odd numbers, even numbers, divisibility
  - operations on integers: $+,\ -,\ \times,\ /$
  - basics of exponentiation: $5^6,\ (-2)^0,\ 6^3,\ \ldots$ and logarithms: $\log_2 16$
  - comparing integers using binary relations $=,\ >,\ <,\ \geq,\ \leq$
  - denoting integers with letters $(n, m, x, y, \ldots)$

    $\rightsquigarrow$ understanding statements like "$2^n > n^2$ *for every integer* n". — is it the case?

- a bit of **algebra:** $(a + b) \times c = (a \times c) + (b \times c),\quad \ldots$
- **working with fractions**
- **common sense**

Please refresh your knowledge of these !

Syllabus

Part I: Computer arithmetic and logic

- How are numbers represented in computers?
- How does hardware really add, subtract, multiply, or divide numbers?

Part II: Basic mathematical concepts and structures

- Sets and operations on sets, relations, functions, graphs, trees.

Part III: Models of computation

- What is a computation or an algorithm?
- What can and what cannot be computed?
- What can be computed with limited memory?

Part IV: Algorithms, Complexity, and Data structures

- How to design an algorithm?
- Is my algorithm efficient?
- How to represent data in computers?

Reading list (see Moodle for more)

✓ **The provided materials** (lecture notes, videos, exercises, models answers)

- **K.H. Rosen.** *Discrete Mathematics and its Applications.* McGraw-Hill, 2019.
- **M.M. Mano and C.R. Kime.** *Logic and Computer Design Fundamentals.* Fourth edition. Pearson, 2008.
- **D.A. Patterson and J.L. Hennessy.** *Computer Organization and Design: The Hardware/Software Interface.* Fourth Edition. Morgan Kaufmann, 2008.
- **E. Kinber and C. Smith.** *Theory of Computing.* A gentle introduction, Prentice Hall, 2001
- **T. Cormen et al.** *Introduction to Algorithms.* Fourth edition. MIT Press, 2022.

✓ **Web resources** such as Google for more; lots of videos on YouTube

`http://en.wikipedia.org/wiki/Logic_gate`
`http://en.wikipedia.org/wiki/Naive_set_theory`
`http://en.wikipedia.org/wiki/Automata_theory`
`https://en.wikipedia.org/wiki/Data_structure` ...