Fundamentals of Computer Science

Module description

Target Audience

- Those looking to explore the concepts that underpin modern computer science principles.
- Learners will begin to appreciate some of the core concepts that form the basis of modern computer systems, how they are designed and built and some of the challenges that they will face in developing their own systems.

Content Differentiation

- This course covers a variety of concepts, presenting them in a clear, accessible and reasoned way. Concepts are explained through examples and learning is reinforced through a variety of interactive activities.

Module goals and objectives

Upon successful completion of this module, you will be able to:

- Understand logical arguments and apply basic concepts of formal proof
- Analyse and predict the behaviour of an algorithm using mathematical techniques
- Understand the process of algorithmic thinking and a number of proof techniques and apply this knowledge to solve a range of computer science problems
- Understand and apply various concepts in automata theory such as deterministic automata, regular languages, and context-free grammar
- Understand the process of computation through Turing machines

Textbook and Readings

Specific essential readings for each week from the following list are included in the Readings page for each week:

1. Kenneth H. Rosen (2011). Discrete Mathematics and its Applications, 7th. McGraw-Hill
2. Michael Sipser (2012). Introduction to the theory of computation, 3rd. Cengage Learning
3. John Hopcroft et al. (2013). Introduction to Automata Theory, Languages and Computation, Pearson

4. Dexter Kozen (2007). Automata and Computability, 1st. Springer
5. Merlin Forbes (2012). A Theoretical Introduction to Turing Machine, 1st. Learning Press
6. Shi-Kui Chang (2003). Data Structures and Algorithms, 1st. World Scientific Publishing Co

Module outline

The module consists of ten topics that focus on key areas of the fundamentals of computer science.

Topic 1. Logic	**Key concepts:** Propositional logic, truth tables, tautology, consistency, contradiction, equivalences and first-order logic. **Learning outcomes:** • Understand logical arguments and apply basic concepts of formal proof.
Topic 2. Proof techniques	**Key concepts:** Deductive proof, proof by contradiction and description of inductive steps. **Learning outcomes:** • Follow correctly a sequence of justified steps to reach a conclusion statement. • Prove a conclusion statement by first assuming it is false. • Describe inductive steps. • Understand logical arguments and apply basic concepts of formal proof.
Topic 3. Basic Combinatorial principles	**Key concepts:** Counting, permutations, combinations, inclusions, exclusions and the pigeonhole principle. **Learning outcomes:**

	• Explore finite or countable discrete structures in the context of computer science. • Consider how different rules can be applied to appreciate the number of possible outcomes for an event • Explore relationships between sets. and elements within or across sets. • Consider how elements in a set can be counted.
Topic 4. Automata theory	**Key concepts:** Definitions, letters, strings, finite automata, language, deterministic and non-deterministic finite automatons. **Learning outcomes:** • Understand the basic terminologies of automata theory. • Describe finite automata and what it can represent. • Build dfa and nfa. • Understand and apply various concepts in automata theory such as deterministic automata, regular languages, and context-free grammar.
Topic 5. Regular languages	**Key concepts:** Regular expressions, finite automaton, pumping lemma **Learning outcomes:** • Describe formal languages in the context of regular expressions. • Identify examples of regular expressions and finite automaton. • Write regular expressions with and without

	the use of finite automaton.
Topic 6. Context-free languages	**Key concepts:** Grammar, language, regular expressions, Comsky Normal Form, context-free grammar. **Learning outcomes:** • Consider context-free grammar and its utility in computer science • Explore the language of grammar and designing a grammar • Be able to convert between regular expressions, context-free grammar • Conversion to normal form • Comsky normal form
Topic 7. Turing machines	**Key concepts:** Turing machines, non-context free language, language and power of Turing machines. **Learning outcomes:** • Understand the process of computation through Turing machines. • Consider the design and utility of Turing machines. • Explore the power and language of Turing machines. • Discuss none context-free languages.
Topic 8. Algorithms I	**Key concepts:** Algorithms, insertions, sorts, bubble sorts, representation, binary search, heap sorts. **Learning outcomes:** • Understand the process of algorithmic thinking and a number of proof techniques and apply this knowledge to solve a range of

	computer science problems. • Explore different techniques for sorting and searching using algorithms. • Identify the value of using different types of algorithmic techniques in different contexts
Topic 9. Algorithms II	**Key concepts:** Recursion, iteration, quick sorts, merging lists, merge sorts, Shapley-proofs, stable matching. **Learning outcomes:** • Explore more advanced algorithmic techniques, including recursion and sorting algorithms. • Consider merging lists, merge sorts and how they work. • Understand and apply algorithms in the context of Shapley-proofs and stable matching.
Topic 10. Complexity theory	**Key concepts:** Efficiency, bubble sorts, binary search, asymptotic complexity, big o notation, recursion complexity, master theorem, quick and merge sorts. **Learning outcomes:** • Explore the concepts of efficiency, average-best, worst • Explore the usage of bubble sorts and binary searches • Consider asymptotic complexity and Big O notation. • Explore recursion complexity and master theorem in the context of large problems.

	• Consider efficient methods for problem solving

Activities of this module

The module is comprised of the following elements:

- Lecture videos. In each week the concepts you need to know will be presented through a collection of short video lectures. You may stream these videos for playback within the browser by clicking on their titles or download the videos. You may also download the slides that go along with the videos.
- Readings. Each topic may include several suggested readings. These are a core part of your learning, and, together with the videos, will cover all of the concepts you need for this course.
- Practice Quizzes. Each week will include one or more practice quizzes, intended for you to assess your understanding of the topics. You will be allowed unlimited attempts at each practice quiz. Each attempt may present a different selection of questions to you. There is no time limit on how long you take to complete each attempt at the quiz. These quizzes do not contribute toward your final score in the module.
- Graded Quizzes. There are some graded quizzes. You will be allowed three attempts per every eight hours at each quiz. Each attempt may present a different selection of questions to you. Your highest score will be used when calculating your final score in the module.
- Discussion Prompt. Each week include one or more discussion prompts. You will see the discussion prompt alongside other items in the lesson. Each prompt provides a space for you to respond. After responding, you can see and comment on your peers' responses
- Interactive simulations. Some of the topics will include interactive software which help you understand certain concepts by interacting with simulations. We hope they will be fun, but they also aim to provide you with valuable practical learning.

How to pass this module

The module has two major assessments each worth 50% of your grade:

- Coursework: this consists of several activities that you do on the Coursera platform and which will be assessed half way through course (after week 11)
- Written examination: you will take this at an examination centre in your country.

The mark shown on the Coursera platform is your coursework mark and you should remember that the exam counts for another 50%.

The coursework consists of several activities. This is a detailed breakdown of all of the marks.

Activity	Required?	Deadline week	Estimated time per module	% of final grade
End of topic quizzes for topics 1-5	Yes	1-10	1-2 hours	25%
Written, staff graded coursework	Yes	11	Approximately 20 hours	25%
Written examination	Yes	22	2 hours 15 minutes	50%