

Concepts of Computer Science I

Introduction

The Module



Lecturers:

Dr. Mike Edwards, CoFo 217, michael.edwards@swansea.ac.uk

Dr. Jingjing Deng, CoFo 223, <u>i.deng@swansea.ac.uk</u>

Lectures: 10 teaching weeks of 3, 1-hour lectures (labs in Weeks 6, 7, 8)

Auditorium, Great Hall on Monday, 1 hour, 13:00-14:00

GH043, Great Hall on Friday, 2 hour, 12:00-14:00

Assessment:

3 Assignments (10% each)
 Data Representation (Released: 8th Oct., Deadline: 25th Oct.)
 Assembly Programming (Released: 29th Oct., Deadline: 15th Nov.)
 Logic Gates (Released 19th Nov., Deadline: 4th Dec.)

Exam (70%)

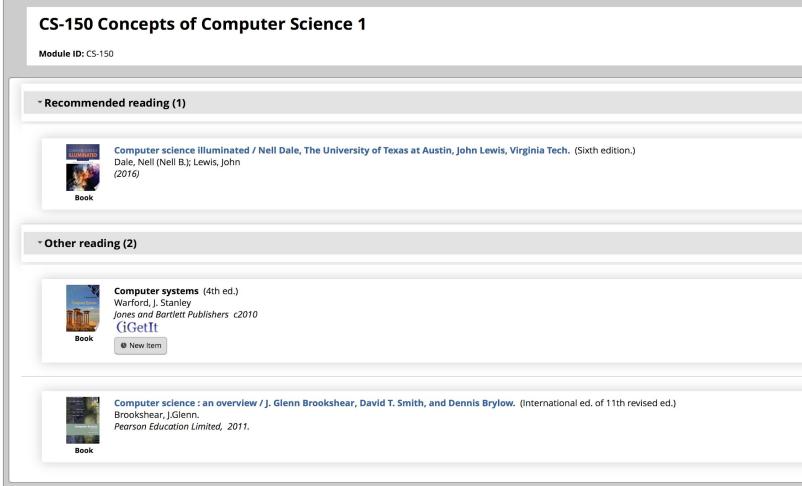
My Expectations



- You will likely have covered some of the content of this module before, there will be differences and challenges for all – do not assume you know everything already!
- When things are new and challenging, deciding that its "too hard for me" or "why do I need to know this anyway" is the wrong attitude!
- When you don't understand something, please ask.
- Have fun!







Blackboard



On the Blackboard site you will find:

- Module Information
- Announcements
- Module Content
 - PowerPoint Slides as used in Lectures
 - Video Tutorials
- Assessment and Feedback
- Reading List

Topics Within the Module



- Brief history of computers and software
- Number systems
- Data representation
- Computing components
- Assembly language programming
- Programming languages and translation
- Logic, gates and circuits

Brief History of Computers and Software



- Nature of computer systems
- Levels of abstraction
- History of hardware
- History of software
- Computing vs Computer Science vs ICT
- Types of computer user

Number Systems



- Categories of numbers
 Natural, negative, integer and rational numbers
- Positional notation
- Differing number bases
 Base 10, 2, 8 and 16
- Conversion between bases
 - Simple algorithms
- Important bases for Computer Science

Data Representation



- Analogue vs. digital information
- Binary formats for negative and floating-point values
 - Sign-magnitude, 2s complement and floating-point
- Characteristics of the ASCII and Unicode character sets
- Data compression and compression ratios
- Perform various types of text compression
 - Key-word encoding, Run-length encoding, Huffman encoding
- The nature of sound and its representation
- How RGB values define a colour
- Distinguish between raster and vector graphics

Computing Components



- Components and their function in von Neumann machine
 - ALU, Control unit, CPU, Memory
- Fetch-decode-execute cycle of von Neumann machine
- Organization and access of memory
- Auxiliary storage devices
 - Seek time, Latency, Access time
- Parallel computer configurations and speeding up processing
 - Pipelining
 - SISD, MIMD, SIMD, MISD
 - RISC architectures

Low Level Programming



- What operations can a computer can perform
 - Machine language
- The Pep/8 virtual machine
- Immediate and direct addressing modes
- Writing simple machine-language programs
- Machine language and assembly language
- Creating and running an assembly-language program
- Writing simple programs using the Pep/8 virtual machine
- Concept of assembler instructions

Pseudocode and Language Translation



- Distinguish between following an algorithm and developing one
- Describe the pseudocode constructs used in expressing an algorithm
- Use pseudocode to express an algorithm
- Design and implement a test plan for a simple assembly-language program
- Translating pseudocode into assembly language
- Translating assembly language into machine code
- Translating high-level languages into executable form
- Compilers, Interpreters and Compiler-Compilers
- Recursion vs Iteration
 - A question of efficiency
- Evolution of programming languages

Gates and Circuits



- Identify the basic gates and describe the behavior of each
 - NOT, AND, OR, XOR, NAND and NOR
- Describe how gates are implemented using transistors
- Combine basic gates into circuits
 - Combinational circuits
 - Sequential circuits
- Describe the behavior of a gate or circuit using Boolean expressions, truth tables, and logic diagrams
 - Circuit equivalence
- Compare and contrast a half adder and a full adder
- Describe how a multiplexer works
- Explain how an S-R latch operates
- Describe the characteristics of the four generations of integrated circuits