



Lacon Childe School

More than just a school

OCR

Oxford Cambridge and RSA

GCSE COMPUTER SCIENCE

[9 – 1]

MY PERSONAL LEARNING TRACKER

Name:

PAPER 1 – COMPUTER SYSTEMS (50%) - (80 MARKS)

1.1 Systems Architecture

1.1.1 Architecture of the CPU	Autumn	Feb
☐ The purpose of the CPU		
● The fetch-execute cycle		
☐ Von Neumann architecture:		
● MAR (Memory Address Register)		
● MDR (Memory Data Register)		
● Program Counter		
● Accumulator		
☐ Common CPU components and their function:		
● ALU (Arithmetic Logic Unit)		
● CU (Control Unit)		
● Cache		
☐ Registers		
1.1.2 CPU performance	Autumn	Feb
☐ How common characteristics of CPUs affect their performance:		
● Clock speed		
● Cache size		
● Number of cores		
1.1.3 Embedded systems:	Autumn	Feb
☐ The purpose and characteristics of embedded systems		
☐ Examples of embedded systems.		

1.2 Memory and Storage

1.2.1 Primary Storage (Memory)	Autumn	Feb
☐ The need for primary storage		
☐ The difference between RAM and ROM		
☐ The purpose of ROM in a computer system		
☐ The purpose of RAM in a computer system		
☐ The need for virtual memory		
1.2.2 Secondary Storage	Autumn	Feb
☐ The need for secondary storage		
☐ Common types of storage:		
● Optical		
● Magnetic		
● Solid state		
☐ Suitable storage devices and storage media for a given application		
☐ The advantages and disadvantages of different storage devices and storage media relating to these characteristics:		
● Capacity		
● Speed		
● Portability		
● Durability		
● Reliability		
● Cost		
1.2.3 Units	Autumn	Feb
☐ The units of data storage:		

● Bit		
● Nibble (4 bits)		
● Byte (8 bits)		
● Kilobyte (1,000 bytes or 1KB)		
● Megabyte (1,000 KB)		
● Gigabyte (1,000 MB)		
● Terabyte (1,000 GB)		
● Petabyte (1,000 TB)		
☐ How data is converted into binary format for processing		
☐ Data capacity and calculations of data capacity requirements		

1.2.4 Data Storage	Autumn	Feb
---------------------------	---------------	------------

Numbers		
----------------	--	--

☐ How to convert positive denary whole numbers to binary numbers (up to and including 8 bits) and vice versa		
☐ How to add two binary integers together (up to and including 8 bits) and explain overflow errors which may occur		
☐ How to convert positive denary whole numbers into 2-digit hexadecimal numbers and vice versa		
☐ How to convert binary integers to their hexadecimal equivalents and vice versa		
☐ Binary shifts		

Characters		
-------------------	--	--

☐ The use of binary codes to represent characters		
☐ The term 'character set'		
☐ Relationship between the number of bits per character in a character set, and the number of characters which can be represented, e.g.		
● ASCII		
● Unicode		

Images		
---------------	--	--

☐ How an image is represented as a series of pixels, represented in binary		
☐ Metadata		
☐ The effect of colour depth and resolution on:		
● The quality of the image		
● The size of the image		

Sound		
--------------	--	--

☐ How sound can be sampled and stored in digital form		
☐ The effect of sample rate, duration and bit depth on:		
● The playback quality		
● The size of a sound file		

1.2.5 Compression	Autumn	Feb
--------------------------	---------------	------------

☐ The need for compression		
☐ Types of compression:		
● Lossy		
● Lossless		

My notes:

1.3 Computer Networks, Connections and Protocols

1.3.1 Networks and topologies	Autumn	Feb
☐ Types of network:		
● LAN (Local Area Network)		
● WAN (Local Area Network)		
☐ Factors that affect the performance of networks		
☐ The different roles of computers in a client-server and a peer-to-peer network		
☐ The hardware needed to connect stand-alone computers into a Local Area Network:		
● Wireless access points		
● Routers		
● Switches		
● NIC (Network Interface Controller/Card)		
● Transmission media		
☐ The internet as a worldwide collection of computer networks:		
● DNS (Domain Name Server)		
● Hosting		
● The Cloud		
● Web servers and clients		
☐ Star and Mesh network topologies		
1.3.2 Wired and wireless networks, protocols and layers	Autumn	Feb
☐ Modes of connection		
● Wired		
● Ethernet		
● Wireless		
● Wi-Fi		
● Bluetooth		
☐ Encryption		
☐ IP addressing and MAC addressing		
☐ Standards		
☐ Common protocols including:		
● TCP/IP (Transmission Control Protocol/Internet Protocol)		
● HTTP (Hypertext Transfer Protocol)		
● HTTPS (Hypertext Transfer Protocol Secure)		
● FTP (File Transfer Protocol)		
● POP (Post Office Protocol)		
● IMAP (Internet Message Access Protocol)		
● MPT (Simple Mail Transfer Protocol)		
☐ The concept of layers		

1.4 Network security

1.4.1 Threats to computer systems and networks	Autumn	Feb
☐ Forms of attack:		
● Malware		
● Social engineering, e.g. phishing, people as the 'weak point'		
● Brute force attacks		
● Denial of service attacks (DOS)		
● Data interception and theft		
● The concept of SQL injection		

1.4.2 Identifying and preventing vulnerabilities

☐ Common prevention methods:		
● Penetration testing		
● Anti-malware software		
● Firewalls		
● User access levels		
● Passwords		
● Encryption.		
● Physical security		

1.5 Systems Software

1.5.1 Operating systems:	Autumn	Feb
☐ The purpose and functionality of operating systems:		
● User interface		
● Memory management/ multitasking		
● Peripheral management and drivers		
● User management		
● File management		

1.5.2 Utility software

☐ The purpose and functionality of utility software		
☐ Utility system software:		
● Encryption software		
● Defragmentation		
● Data compression		

1.6 Ethical, legal, cultural and environmental impacts of digital technology

1.6.1 Ethical, legal, cultural and environmental impact	Autumn	Feb
☐ Impacts of digital technology on wider society including:		
● Ethical issues		
● Legal issues		
● Cultural issues		
● Environmental issues		
● Privacy issues.		
☐ Legislation relevant to Computer Science:		
● The Data Protection Act 2018 (1998)		
● Computer Misuse Act 1990		
● Copyright Designs and Patents Act 1988		
● Software licences (i.e. open source and proprietary)		

My notes:

PAPER 2 – COMPUTATIONAL THINKING, ALGORITHMS & PROGRAMMING (80 MARKS (50%))

2.1 – Algorithms

2.1.1 Computational thinking	Autumn	Feb
☐ Principles of computational thinking:		
● Abstraction		
● Decomposition		
● Algorithmic thinking		
2.1.2 Designing, creating and refining algorithms	Autumn	Feb
☐ Identify the inputs, processes, and outputs for a problem		
☐ Structure diagrams		
● Flowcharts		
● Reference language/high-level programming language		
☐ Identify common errors		
☐ Trace tables		
2.1.3 Searching and sorting algorithm	Autumn	Feb
☐ Standard searching algorithms:		
● Binary search		
● Linear sort		
☐ Standard sorting algorithms:		
● Bubble sort		
● Merge sort		
● Insertion sort		

2.2 – Programming fundamentals

2.2.1 Programming fundamentals		
☐ The use of variables, constants, operators, inputs, outputs and assignments		
☐ The use of the three basic programming constructs used to control the flow of a program:		
● Sequence		
● Selection		
● Iteration (count- and condition-controlled loops)		
☐ The common arithmetic operators		
☐ The common Boolean operators AND, OR and NOT		
2.2.2 Data types		
☐ The use of data types:		
● Integers		
● Real		
● Boolean		
● Character and string		
● Casting		
2.2.3 Additional programming techniques		
☐ Additional programming techniques		
☐ The use of basic file handling operations:		
● Open		
● Read		
● Write		
● Close		
☐ The use of records to store data		

☐ The use of SQL to search for data		
☐ The use of arrays (or equivalent) when solving problems, including both:		
● one-dimensional (1D)		
● two-dimensional (2D) arrays		
☐ How to use sub programs (functions and procedures) to produce structured code		
☐ Random number generation		

2.3 – Producing robust programs

2.3.1 Defensive design

☐ Defensive design considerations:		
● Anticipating misuse		
● Authentication		
☐ Input validation		
☐ Maintainability:		
● Use of sub programs		
● Naming conventions		
● Indentation		
● Commenting		

2.3.2 Testing

☐ The purpose of testing		
☐ Types of testing:		
● Iterative		
● Final/terminal		
☐ Identify syntax and logic errors		
☐ Selecting and using suitable test data:		
● Normal		
● Boundary		
● Invalid/Erroneous		
☐ Refining algorithms		

2.4 – Boolean logic

2.4.1 Boolean logic

☐ Simple logic diagrams using the operators AND, OR and NOT		
☐ Truth tables		
☐ Combining Boolean operators using AND, OR and NOT		
☐ Applying logical operators in truth tables to solve problems		

2.5 – Programming languages and Integrated Development Environments

2.5.1 Languages

☐ Characteristics and purpose of different levels of programming language:		
● High-level languages		
● Low-level languages		

2.5.2 The Integrated Development Environment (IDE)

☐ Common tools and facilities available in an Integrated Development Environment (IDE):		
● Editors		
● Error diagnostics		
● Run-time environment		
● Translators		

My notes: