KS4

Computer Science

Exam Details:

Exam Board: OCR

Exam consists of two papers:

Paper 1: Computer Systems
50%. 80 marks (non-calculator paper)

(1hr 30mins)

All questions are mandatory

Paper 2: Computational thinking, algorithms and programmingm50%. 80 marks

(1hr 30mins)

Paper consists of section A and B. Students must answers both sections. All questions are mandatory.

PAPER 1 – Computer Systems (J277/01)			
1.1 – Systems architecture			
1.1.1 Architecture of the CPU			
1.1.1 a The purpose of the CPU:			
1.1.1 a i The fetch-execute cycle			
1.1.1 b Common CPU components and their function:			
1.1.1 b i ALU (Arithmetic Logic Unit)			
1.1.1 b ii CU (Control Unit)			
1.1.1 b iii Cache			
1.1.1 b iv Registers			
1.1.1 c Von Neumann architecture:			
1.1.1 c i MAR (Memory Address Register)			
1.1.1 c ii MDR (Memory Data Register)			
1.1.1 c iii Program Counter			
1.1.1 c iv Accumulator			
1.1.2 CPU performance			
1.1.2 a How common characteristics of CPUs affect their performance:			
1.1.2 a i Clock speed			
1.1.2 a ii Cache size			
1.1.2 a iii Number of cores			
1.1.3 Embedded systems			
1.1.3 a The purpose and characteristics of embedded systems			
1.1.3 b Examples of embedded systems			
1.2 – Memory and storage			
1.2.1 Primary storage (Memory)			
1.2.1 a The need for primary storage			
1.2.1 b The difference between RAM and ROM			
1.2.1 c The purpose of ROM in a computer system			
1.2.1 d The purpose of RAM in a computer system			
1.2.1 e i Why virtual memory may be needed in a system			
1.2.1 e ii How virtual memory works			
1.2.2 Secondary storage			
1.2.2 a The need for secondary storage			
1.2.2 b Common types of storage:			
1.2.2 b i Optical			
1.2.2 b ii Magnetic			
1.2.2 b iii Solid state			
1.2.2 c Suitable storage devices and storage media for a given application			
1.2.2 d The advantages and disadvantages of different storage devices and storage media relating to these characteristics:			
1.2.2 d i Capacity			
1.2.2 d ii Speed			

1.2.2 d iii Portability			
1.2.2 d iv Durability			
1.2.2 d v Reliability			
1.2.2 d vi Cost			
1.2.3 Units			
1.2.3 a The units of data storage:			
1.2.3 a i Bit			
1.2.3 a ii Nibble (4 bits)			
1.2.3 a iii Byte (8 bits)			
1.2.3 a iv Kilobyte (1,000 bytes or 1 KB)			
1.2.3 a v Megabyte (1,000 KB)			
1.2.3 a vi Gigabyte (1,000 MB)			
1.2.3 a vii Terabyte (1,000 GB)			
1.2.3 a viii Petabyte (1,000 TB)			
1.2.3 b Familiarity with data units and moving between each			
1.2.3 c Why data must be stored in binary format			
1.2.3 d How data needs to be converted into a binary format to be processed by a computer			
1.2.3 e Data capacity and calculation of data capacity requirements			
1.2.3 f Calculate capacity of devices			
1.2.3 g Calculate required capacity for a given set of files			
1.2.4 Data storage			
Numbers			
1.2.4 a How to convert positive denary whole numbers to binary numbers (up to and including 8 bits) and vice versa			
1.2.4 b How to add two binary integers together (up to and including 8 bits) and explain overflow errors which may occur			
1.2.4 c How to convert positive denary whole numbers into 2-digit hexadecimal numbers and vice versa			
1.2.4 d How to convert binary integers to their hexadecimal equivalents and vice versa			
1.2.4 e Binary shifts			
1.2.4 f Understanding of the terms most significant bit, and least significant bit			
1.2.4 g Characters			
1.2.4 g i The use of binary codes to represent characters			
1.2.4 g ii The term ‘character set’			
1.2.4 g iii The relationship between the number of bits per character in a character set, and the number of characters which can be represented, e.g.:			
1.2.4 g iv ASCII (8 bit)			
1.2.4 g v Unicode			
1.2.4 h text file size = bits per character x number of characters			
1.2.4. j Images			
1.2.4 j i How an image is represented as a series of pixels, represented in binary for a specific code			
1.2.4 j ii Metadata			
The effect of colour depth and resolution on:			

1.2.4 j iii The quality of the image			
1.2.4 j iv The size of an image file			
1.2.4 j v colour depth x image height (px) x image width (px)			
1.2.4 k Sound			
1.2.4 k i How sound can be sampled and stored in digital form			
1.2.4 k ii Sample rate – measured in Hertz (Hz)			
1.2.4 k iii Bit depth – number of bits available to store each sample (e.g. 16-bit)			
The effect of sample rate, duration and bit depth on:			
1.2.4 k iv The playback quality			
1.2.4 k v The size of a sound file = sample rate x duration (s) x bit depth			
1.2.5 Compression			
1.2.5 a The need for compression			
1.2.5 b Lossy Compression			
1.2.5 c Lossless Compression			
1.2.5 d Common scenarios where compression may be needed			
1.2.5 e Advantages and disadvantages of each type of compression			
1.2.5 f Effects on the file for each type of compression			
1.3 – Computer networks, connections and protocols			
1.3.1 Networks and topologies			
1.3.1 a Types of network, characteristics and examples:			
1.3.1 a i LAN (Local Area Network)			
1.3.1 a ii WAN (Wide Area Network)			
1.3.1 b Factors that affect the performance of networks (number of devices and bandwidth)			
1.3.1 c The different roles of computers in a client-server and a peer-to- peer network			
1.3.1 d The hardware needed to connect stand-alone computers into a Local Area Network:			
1.3.1 d i Wireless access points			
1.3.1 d ii Routers			
1.3.1 d iii Switches			
1.3.1 d iv NIC (Network Interface Controller/Card)			
1.3.1 d v Transmission media			
1.3.1 d vi File servers			
1.3.1 e The Internet as a worldwide collection of computer networks:			
1.3.1 e i DNS (Domain Name Server)			
1.3.1 e ii Hosting			
1.3.1 f The Cloud (storage, software, processing)			
1.3.1 f i Advantages and disadvantages of the Cloud			
1.3.1 g Web servers and clients			
1.3.1 h Star and Mesh network topologies			
1.3.1 h i Advantages and disadvantages of the Star and Mesh topologies			
1.3.2 Wired and wireless networks, protocols and layers			
1.3.2 a Modes of connection:			
1.3.2 a i Wired Ethernet			

1.3.2 a ii Wireless Wi-Fi			
1.3.2 a iii Wireless Bluetooth			
1.3.2 b Compare benefits and drawbacks of wired versus wireless connection			
1.3.2 c Recommend one or more connections for a given scenario			
1.3.2 d Encryption			
1.3.2 e IP addressing (IPv4 and IPv6)			
1.3.2 f MAC addressing			
1.3.2 g Standards to provide rules for areas of computing			
1.3.2 h Standards allows hardware/software to interact across different manufacturers/producers			
1.3.2 j Common protocols including:			
1.3.2 j i TCP/IP (Transmission Control Protocol/Internet Protocol)			
1.3.2 j ii HTTP (Hyper Text Transfer Protocol)			
1.3.2 j iii HTTPS (Hyper Text Transfer Protocol Secure)			
1.3.2 j iv FTP (File Transfer Protocol)			
1.3.2 j v POP (Post Office Protocol)			
1.3.2 j vi IMAP (Internet Message Access Protocol)			
1.3.2 j vii SMTP (Simple Mail Transfer Protocol)			
1.3.2 k The concept of layers used in protocols, and the benefits of using layers (4-layer TCP/IP model)			
1.4 – Network security			
1.4.1 Threats to computer systems and networks			
1.4.1 a Forms of attack:			
1.4.1 a i Malware			
1.4.1 a ii Social engineering, e.g. phishing, people as the 'weak point'			
1.4.1 a iii Brute-force attacks			
1.4.1 a iv Denial of service attacks			
1.4.1 a v Data interception and theft			
1.4.1 a vi The concept of SQL injection			
1.4.2 Identifying and preventing vulnerabilities			
1.4.2 a Common prevention methods:			
1.4.2 a i Penetration testing			
1.4.2 a ii Anti-malware software			
1.4.2 a iii Firewalls			
1.4.2 a iv User access levels			
1.4.2 a v Passwords			
1.4.2 a vi Encryption			
1.4.2 a vii Physical security			
1.5 – Systems software			
1.5.1 Operating systems			
1.5.1 a The purpose and functionality of operating systems:			
1.5.1 a i User interface			
1.5.1 a ii Memory management and multitasking			
1.5.1 a iii Peripheral management and drivers			

1.5.1 b User management			
1.5.1 b i Allocation of an account			
1.5.1 b ii Access rights			
1.5.1 b iii Security, etc.			
1.5.1 c File management			
1.5.1 c i Naming			
1.5.1 c ii Allocating to folders			
1.5.1 c iii Moving files			
1.5.1 c iv Saving, etc.			
1.5.1 d Processes need to be managed and what this entails (e.g. the use of buffers when transferring data to a printer)			
1.5.2 Utility software			
1.5.2 a The purpose and functionality of utility software			
1.5.2 a i Encryption software			
1.5.2 a ii Defragmentation			
1.5.2 a iii Data compression			
1.6 – Ethical, legal, cultural and environmental impacts of digital technology			
1.6.1 Ethical, legal, cultural and environmental impact			
1.6.1 a Impacts of digital technology on wider society including:			
1.6.1 a i Ethical issues			
1.6.1 a ii Legal issues			
1.6.1 a iii Cultural issues			
1.6.1 a iv Environmental issues			
1.6.1 a v Privacy issues			
1.6.1 b Knowledge of a variety of examples of digital technology and how this impacts on society			
1.6.1 c Legislation relevant to Computer Science:			
1.6.1 c i The Data Protection Act 2018			
1.6.1 c ii Computer Misuse Act 1990			
1.6.1 c iii Copyright Designs and Patents Act 1988			
1.6.1 c iv Software licences (i.e. open source and proprietary)			
1.6.1 d The need to license software and the purpose of a software licence			
1.6.1 e Features of open source (providing access to the source code and the ability to change the software)			
1.6.1 f Features of proprietary (no access to the source code, purchased commonly as off-the-shelf)			
1.6.1 g Recommend a type of licence for a given scenario including benefits and drawbacks			

PAPER 2 – Computational Thinking, algorithms and programming (J277/02)			
2.1 – Algorithms			
2.1.1 Computational thinking			
2.1.1 a Principles of computational thinking:			
2.1.1 a i Abstraction			
2.1.1 a ii Decomposition			
2.1.1 a iii Algorithmic thinking			
2.1.2 Designing, creating and refining algorithms			
2.1.2 a Identify the inputs, processes, and outputs for a problem			
2.1.2 b Structure diagrams			
2.1.2 c Create, interpret, correct, complete, and refine algorithms using:			
2.1.2 c i Pseudocode			
2.1.2 c ii Flowcharts			
2.1.2 c iii Reference language/high-level programming language			
2.1.2 d Identify syntax errors and suggest fixes			
2.1.2 e Identify logic errors and suggest fixes			
2.1.2 f Create and use trace tables to follow an algorithm			
2.1.2 g Refining algorithms			
2.1.3 Searching and sorting algorithms			
2.1.3 a Standard searching algorithms:			
2.1.3 a i Binary search			
2.1.3 a ii Linear search			
2.1.3 b Standard sorting algorithms:			
2.1.3 b i Bubble sort			
2.1.3 b ii Merge sort			
2.1.3 b iii Insertion sort			
2.2 – Programming fundamentals			
2.2.1 Programming fundamentals			
2.2.1 a The use of variables			
2.2.1 b The use of constants			
2.2.1 c The use of operators			
2.2.1 d The use of inputs			
2.2.1 e The use of outputs			
2.2.1 f The use of assignments			
2.2.1 g The use of the three basic programming constructs used to control the flow of a program:			
2.2.1 g i Sequence			
2.2.1 g ii Selection			
2.2.1 g iii Iteration			
2.2.1 g iv count controlled i.e. for loop			
2.2.1 g v condition controlled i.e. while loop, repeat until			
2.2.1 h The common arithmetic operators			
2.2.1 j The common Boolean operators AND, OR and NOT			
2.2.1 k Comparison operators Arithmetic operators			

2.2.1 k i == Equal to + Addition			
2.2.1 k ii != Not equal to – Subtraction			
2.2.1 k iii < Less than * Multiplication			
2.2.1 k iv <= Less than or equal to / Division			
2.2.1 k v > Greater than MOD Modulus			
2.2.1 k vi >= Greater than or equal to DIV Quotient			
2.2.1 k vii ^ Exponentiation (to the power)			
2.2.2 Data types			
2.2.2 a The use of data types:			
2.2.2 a i Integer			
2.2.2 a ii Real			
2.2.2 a iii Boolean			
2.2.2 a iv Character and string			
2.2.2 a v Casting			
2.2.3 Additional programming techniques			
2.2.3 a The use of basic string manipulation			
2.2.3 a i Concatenating			
2.2.3 a ii Slicing			
2.2.3 b The use of basic file handling operations:			
2.2.3 b i Open			
2.2.3 b ii Read			
2.2.3 b iii Write			
2.2.3 b iv Close			
2.2.3 c The use of records to store data			
2.2.3 d The use of SQL to search for data			
2.2.3 e The use of arrays as fixed length static structures when solving problems			
2.2.3 f The use of 2D arrays as fixed length static structures when solving problems			
2.2.3 g How to use sub programs (procedures) to produce structured code			
2.2.3 h How to use sub programs (functions) to produce structured code			
2.2.3 j Random number generation			
2.2.3 k SQL commands:			
2.2.3 k i SELECT			
2.2.3 k ii FROM			
2.2.3 k iii WHERE			
2.3 – Producing robust programs			
2.3.1 Defensive design			
2.3.1 a Defensive design considerations:			
2.3.1 a i Anticipating misuse and invalid data			
2.3.1 a ii Authentication to confirm the identity of a user			
2.3.1 b Input validation			
2.3.1 b i Length check			

2.3.1 b ii Range check			
2.3.1 b iii Presence check			
2.3.1 c Practical experience of designing input validation and simple authentication (e.g. username and password)			
2.3.1 d Maintainability:			
2.3.1 d i Use of sub programs			
2.3.1 d ii Naming conventions			
2.3.1 d iii Indentation			
2.3.1 d iv Commenting			
2.3.2 Testing			
2.3.2 a The purpose of testing			
2.3.2 b Types of testing:			
2.3.2 b i Iterative (module/unit tests)			
2.3.2 b ii Final/terminal			
2.3.2 c Syntax errors as errors which break the grammatical rules of the programming language and stop it from being run/translated			
2.3.2 d Logic errors as errors which produce unexpected output			
2.3.2 e Selecting and using suitable test data:			
2.3.2 e i Normal test data as data which should be accepted by a program without causing errors			
2.3.2 e ii Boundary test data as data of the correct type which is on the very edge of being valid			
2.3.2 e iii Invalid test data as data of the correct type but outside accepted validation limit			
2.3.2 e iv Erroneous test data as data of the incorrect type which should be rejected by a computer system			
2.3.2 f Ability to create/complete a test plan			
2.4 – Boolean logic			
2.4.1 Boolean logic			
2.4.1 a Simple logic diagrams using the operators AND (conjunction)			
2.4.1 b Simple logic diagrams using the operators OR (disjuntction)			
2.4.1 c Simple logic diagrams using the operators NOT (negation)			
2.4.1 d Truth tables			
2.4.1 e Combining Boolean operators using AND, OR and NOT			
2.4.1 f Applying logical operators in truth tables to solve problems			
2.5 – Programming languages and Integrated Development Environments			
2.5.1 Languages			
2.5.1 a Characteristics and purpose of different levels of programming language:			
2.5.1 a i High-level languages			
2.5.1 a ii Low-level languages			
2.5.1 b The purpose of translators			
2.5.1 b i The characteristics of a compiler			
2.5.1 b ii The characteristics of an interpreter			
2.5.2 The Integrated Development Environment (IDE)			

2.5.2 a Common tools and facilities available in an Integrated Development Environment (IDE):			
2.5.2 a i Editors			
2.5.2 a ii Error diagnostics			
2.5.2 a iii Run-time environment			