SPECIFICATION >

For exams January, May and November onwards
For teaching from September 2021 onwards

Learning Resource Network

INTERNATIONAL AS AND A-LEVEL COMPUTER SCIENCE (7932)

Contents

BACKGROUND TO LRN

Learning Resource Network (LRN) is a recognised Awarding Organisation that offers a range of qualifications to candidates, educational institutes, training providers, schools and employers.

LRN is recognised for its high quality qualifications that enable candidates to progress to other areas of study and employment in their designated fields.

In producing its qualifications, LRN uses the experience and expertise of academics, professionals working in the pertinent industries and assessment practitioners with a wealth of best practice and knowledge of validation, verification, delivery and assessment.

ACCOLADES

Queen’s Award

In April 2020, LRN received the Queen’s Award for Enterprise for International Trade. LRN is one of 220 organisations in the UK to be recognised with this prestigious accolade. This was in recognition of the expansion LRN brought to the overseas qualification market.

MANAGEMENT SYSTEMS

LRN has been awarded international accreditation as part of its quality controls, policies, systems and overall approach to its management systems. These awards are externally validated by the British Assessment Bureau. LRN has achieved accreditation in the form of ISO 9001: Quality Management Systems, ISO 14001: Environment Management Systems and ISO 27001: Information Security Management Systems.

CUSTOMER SERVICE EXCELLENCE

LRN has achieved the prestigious award of Customer Service Excellence. This is in recognition of its customer service practices, approach to managing and dealing with UK and Overseas customer needs, including the diverse needs of its centres.

LRN was the first UK Awarding Organisation to achieve Customer Service Excellence.
Following reaccreditation in 2019, LRN received an award for Customer Service Excellence: Compliance Plus, demonstrating that LRN went above and beyond the delivery of its customer service principles.

British Assessment Bureau UKAS MANAGEMENT SYSTEMS 8289 ISO 9001 QUALITY MANAGEMENT

British Assessment Bureau UKAS MANAGEMENT SYSTEMS 8289 ISO 27001 INFORMATION SECURITY MANAGEMENT

British Assessment Bureau UKAS MANAGEMENT SYSTEMS 8289 ISO 14001 ENVIRONMENTAL MANAGEMENT

CUSTOMER SERVICE EXCELLENCE® CSE

INTRODUCTION

This specification provides an overview to the LRN International AS & A Level Computer Science[1]. This document is suitable for various users, including candidates, centres, administrators, employers, parents/guardians, teachers (and other educational based staff) and examiners. The specification outlines the key features and administrative procedures required for this international qualification.

OBJECTIVE

The LRN International AS & A Level Computer Science is designed to enable international candidates to demonstrate their ability, in theoretical terms across a range of Data representations, internet technologies and hardware. The full range of subject content is shown below, and includes computational thinking, programming and software development.

MODE OF DELIVERY

This qualification has been constructed to be delivered within centres. Centres will need to demonstrate to LRN, through the centre recognition processes, that they have the resources, facilities and competence to deliver. However, centres must be able to demonstrate, in line with LRN's criteria, that they have the means, capability, capacity and resources (including suitably qualified centre staff) to deliver by the method chosen by the centre.

PROGRESSION

The LRN International AS & A Level Computer Science has been designed to reflect the wide variation in candidates' origins, levels of education and career aims. Progression opportunities may, therefore, take a variety of paths. Depending on the level of qualification achieved, it may be appropriate for the candidate to progress to:

1. Similar level 3 qualification in Computer Science;
2. LRN Level 3 Diploma in Pre U Foundation Studies;
3. Qualification (and/or membership) supported by the British Computer Society;
4. A higher level of any qualification – e.g., Diploma/ HNC/HND or Degree;
5. National or Vocationally Related Qualifications

[1] LRN International AS/A Level are globally recognised qualifications designed specifically for international candidates and are available outside the United Kingdom. Candidates based in England refer to the Ofqual register.

QUALIFICATION OVERVIEW

Number	Subject Content	LRN International AS Level	LRN International A Level	AO	Exam
1	Information and Data Representations	√	√	1, 2 and 3	Combination of written exam papers (externally set and marked) and a practical demonstration of skills. **AS Level** **Paper 1:** Short answers and structured questions Duration: 1 hour 30 minutes Weighting: 50% **Paper 2:** Short answers and structured questions Duration: 2 hours Weighting: 50% **A Level** **Paper 1:** Short answers and structured questions Duration: 1 hour 30 minutes Weighting: 25% **Paper 2:** Short answers and structured questions Duration: 2 hours Weighting: 25%
2	Communication and Internet Technologies	√	√	1, 2 and 3	
3	Hardware and Virtual Machines	√	√	1, 2 and 3	
4	Processor Fundamentals	√	√	1, 2 and 3	
5	System Software	√	√	1, 2 and 3	
6	Security, Privacy and Data Integrity	√	√	1, 2 and 3	
7	Ethics and Ownership	√	√	1, 2 and 3	
8	Databases	√	√	1, 2 and 3	
9	Computational thinking, Algorithm Design and Problem-Solving	-	√	1, 2 and 3	
10	Data types and structures	√	√	1, 2 and 3	
11	Programming	√	√	1, 2 and 3	
12	Software Development	-	√	1, 2 and 3	
13	Artificial Intelligence (AI)	-	√	1, 2 and 3	

					Paper 3: Short answers and structured questions Duration: 1 hour 30 minutes Weighting: 25% **Paper 4 (Practical Skills):** Short answers and structured questions Duration: 2 hours Weighting: 25%

BREAKDOWN OF ASSESSMENT OBJECTIVES

AO1: Demonstrate knowledge and understanding of the principles and concepts of computer science, including abstraction, logic, algorithms and data representation.

AO2: Apply knowledge and understanding of the principles and concepts of computer science, including to analyse problems in computational terms.

AO3: Design, program, and evaluate computer systems to solve problems, making reasoned judgements about these.

ASSESSMENT

The assessment for this qualification consists of (i) written exam papers, and (ii) practical demonstration of skills, set and marked by the LRN.

Assessment objectives (AOs)	Weighting			
	Paper 1	Paper 2	Paper 3	Paper 4
AO1	40%	35%	40%	20%
AO2	35%	40%	30%	40%
AO3	25%	25%	30%	40%

GUIDED LEARNING HOURS (GLH)

The LRN International AS Level guided learning hours (GLH) are 180 and 360 guided learning hours for LRN International A Level. Please note the hours stated are indicative.

ENTRIES CODES

One entry per qualification is sufficient and will cover all the question papers including certification.

PRIVATE CANDIDATES

Centres are advised that private candidates are only to be enrolled with prior agreement and confirmation from LRN.

GRADING

The LRN International A Level will be graded on a six-point scale: A*, A, B, C, D and E and LRN International AS Level will be graded on a five-point scale: A, B, C, D and E Candidates who fail to reach the minimum standard for grade E will be recorded as U (unclassified) and will not receive a qualification certificate.

RESULTS

Exam series are in:

- January (results released in March)
- June (results released in August)
- November (results released in January)

RE-TAKES

Whereas candidates can re-take each paper as often as they wish, within the shelf-life of the specification.

CUSTOMER SERVICE STATEMENT

Learning Resource Network (LRN) is committed to ensuring all customers are dealt with promptly and in a professional and helpful manner. In order to guarantee this, we commit to ensuring the following in our day to day interactions with students, assessment centres and our stakeholder network:

- All customers will be treated equally and with respect;
- All customer information will only be used in a way which has been agreed in advance, unless we are informed of something that places them or others at risk of harm;
- All customers will be treated by staff in a professional manner.

LRN has arrangements in place to provide a telephone and e-mail helpdesk which will be staffed from 09:00 to 17:00 from Monday to Friday. Furthermore, it will respond to each e-mail, letter or telephone message it receives regarding feedback on its qualifications, centre approvals process or other matters relating to its products and/or services. The timetable for responding is as follows:

- E-mail: 5 working days
- Letter: 5 working days
- Telephone message: 5 working days

DIVERSITY AND EQUALITY

Learning Resource Network (LRN) is committed to ensuring fair and equal access to its qualifications, examinations and support materials. Our Diversity and Equality policy seeks to eliminate unjustifiable discrimination, harassment and/or victimisation and to advance equality of opportunity, thereby ensuring all candidates are treated fairly, in accordance with the protected characteristics of the Equality Act 2010. Specifically, we comply fully with the requirements laid out in the Equality Act 2010. In addition, and within the constraints of this policy, LRN will have due regard for the General data Protection Regulations (GDPR) in the retention of information which is unnecessary.

1	**Information and Data Representations**		
Aim The aim of this subject content is to enable learners to demonstrate a theoretical and practical understanding of how data is gathered and then converted from one base to another			
Learning Outcomes - The learner will:		**Assessment Criteria - The learner can:**	
1	Understand data representation in the context of binary and character sets.	1.1	**Convert** positive integers between binary hexadecimal and denary.
		1.2	**Analyse** how character sets are used in computer systems
		1.3	**Analyse** how binary data is used in computer systems.
2	Understand ways in which multimedia is represented through graphics and sound.	2.1	**Assess** how a bitmap image is represented and stored on a computer.
		2.2	**Explain** how a vector graphic is represented and stored on a computer.
		2.3	**Analyse** whether a bitmap image or vector graphic is more appropriate for a given task.
		2.4	**Explain** how an analogue sound wave is digitised.
		2.5	**Evaluate** the effect of changing the sample rate and resolution on a sound wave.
3	Understand the principles of data compression.	3.1	**Assess** the purpose of data compression.
		3.2	**Differentiate** between lossy and lossless data compression.
4	Be able to demonstrate a practical application of information and data representations.	4.1	**Investigate** methods for converting a number from one base to another.
		4.2	**Perform** calculations with binary additional and subtraction
		4.3	**Apply** ASCII, extended ASCII and Unicode to represent textual data.
		4.4	**Investigate** lossy and lossless data compression
		4.5	**Justify** the use of a method in a number of given situations.
		4.6	**Survey** an appropriate method of file organisation and file access for a given problem.

		4.7	**Select and design** an appropriate user-defined data type for a given problem.
		4.8	**Convert** binary floating-point read numbers into denary and vice versa.
		4.9	**Normalise** floating-point numbers.
		4.10	**Investigate** how a sound/image/text can be compressed using run-length encoding.
5	Understand the concepts of user-defined data types	5.1	**Examine** why user defined data types are necessary.
		5.2	**Define and use** composite and non-composite data types.
6	Understand the principles of file organisation and access	6.1	**Determine** the different methods of file organisation and file access
		6.2	**Describe and use** hashing algorithms.
7	Understand floating-point numbers, representation, and manipulation	7.1	**Describe** the format of binary floating-point real numbers.
		7.2	**Point out** the consequences of a binary representation only being an approximation to the real number it represents (in certain cases).
		7.2	**Justify** that binary representations can give rise to rounding errors.

2	**Communication and Internet Technologies**		
Aim The aim of this subject content is to enable learners to demonstrate a theoretical understanding of communication and networks including the internet.			
Learning Outcomes - The learner will:		**Assessment Criteria - The learner can:**	
1	Understand networks including the internet (introduction to types of network, hardware, and data transmission)	1.1	**Examine** the purpose and benefits of networking devices.
		1.2	**Discover** the characteristics of a LAN and a WAN.
		1.3	**Judge** whether a given network is a LAN or a WAN
		1.4	**Describe** the use, benefits and drawbacks of cloud computing.
		1.5	**Analyse** the characteristics of a client-server and peer-to-peer network.
		1.6	**Summarise** the benefits and drawbacks of a client-server and peer-to-peer network.
		1.7	**Justify** the use of a client-server or peer-to-peer network in a given scenario.
		1.8	**Analyse** the characteristics, benefits and drawbacks, of different network topologies.
		1.9	**Differentiate** between wired and wireless networks.
		1.10	**Classify** the benefits and drawbacks of both wired and wireless connections.
		1.11	**Assess** the purpose of hardware components that can support a LAN.
		1.12	**Propose** the appropriate components to create a LAN.
		1.13	**Explain** the role and function of a router in a network.
		1.14	**Define** collisions in data transmission and **determine** how Ethernet detects and avoids collisions.
		1.15	**Compare** the internet and the WWW.
		1.16	**Point out** the hardware required to communicate over the internet.

		1.17	**Appraise** the use of IP addresses in the transmission of data over the internet.
		1.18	**Outline** the benefits of a URL over an IP.
		1.19	**Analyse** the role of a DNS in converting a URL to IP.
2	Understand different communication protocols and their purposes.	2.1	**Determine** why a protocol is essential for communication between computers.
		2.2	**Examine** protocol implements as a stack, with each layer having its own functionality
		2.3	**Illustrate** the TCP/IP protocol suite.
		2.4	**Outline** the purposes of these protocols: HTTP, FTP, POP3, IMAP, SMTP, BitTorrent.
3	Understand the principles of circuit and packet switching	3.1	**Investigate** the purpose, benefits and drawbacks of circuit switching and packet switching.
		3.2	**Judge** the use of packet and/or circuit switching in a scenario.

3	**Hardware and Virtual Machines**		
Aim The aim of this subject content is to enable learners to demonstrate a theoretical and practical analysis of hardware, virtual machines, and their applications.			
Learning Outcomes - The learner will:		**Assessment Criteria - The learner can:**	
1	Understand the purpose of computers and their components.	1.1	**Distinguish** between primary and secondary storage.
		1.2	**Summarise** the items that are stored in secondary storage.
		1.3	**Differentiate** between RAM and ROM.
		1.4	**Compare** SRAM with DRAM.
		1.5	**Compile** the difference(s) between PROM, EPROM, and EEPROM.
		1.6	**Analyse** the principal operations of a range of hardware devices.
		1.7	**Discover** the purpose and use of buffers in a range of devices.
		1.8	**Survey** the uses of sensors and identify appropriate sensors for a scenario.
		1.9	**Distinguish** between a monitoring and control system.
		1.10	**Examine** the use and function of a monitoring and control system in a given situation.
		1.11	**Discover and define** the functions of: NOT, AND, OR, NAND, NOR, and XOR (EOR) truth table.
		1.12	**Analyse** Reduced Instruction Set Computers (RISC) and Complex Instruction Set Computers (CISC) processors
		1.13	**Summarise** the importance and use of pipelining and registers in RISC processors.
		1.14	**Investigate** the four basic computer architectures (SISD, SIMD, MISD, and MIMD).
		1.15	**Outline** the characteristics of massively parallel computers.

		1.16	**Compile** the concept, benefits and limitations of a virtual machine.
2	Be able to demonstrate a practical application of hardware and virtual machines**.**	2.1	**Use** the NOT, AND, OR, NAND, NOR and XOR logic gate symbols to **create** the truth table for each of the logic gates
		2.2	**Construct** a logic circuit and logic expression
		2.3	**Create** truth tables for logic circuits including half adders and full adders.
		2.4	**Describe** the function and **design** a truth table for a flip-flop (SR, JK).
		2.5	**Use** Boolean algebra to manipulate Boolean expressions.
		2.6	**Predict** the use of, and **use** a Karnaugh map (K-map)

4	**Processor Fundamental**		
Aim The aim of this subject content is to enable learners to demonstrate a theoretical and practical analysis of CPU architecture, assembly language, and bit manipulation.			
Learning Outcomes - The learner will:		**Assessment Criteria - The learner can:**	
1	Understand processor fundamentals.	1.1	**Describe** the Von Neumann model for a computer system.
		1.2	**Analyse** the purpose and role of each register in the Von Neumann model.
		1.3	**Evaluate** the purpose of and role of the components within the processor.
		1.4	**Infer** how the different ports allow connection to peripherals.
		1.5	**Discuss** the stages of the Fetch-Execute cycle.
		1.6	**Describe** the purpose of interrupts.
		1.7	**Illustrate** how interrupts are handled in the F-E cycle.
		1.8	**Examine** the relationship between assembly language and machine code.
		1.9	**Describe** the stages of the assembly process for a two-pass assembler.
		1.10	**Categorise** assembly language instructions.
		1.11	**Summarize** the different modes of addressing.
		1.12	**Discuss** the impact of a shift on a binary number
2	Be able to demonstrate a practical application of processor fundamentals.	2.1	**Use** assembly language instructions to dry run a program.
		2.2	**Perform** shifts on a binary number.
		2.3	**Apply** bit manipulation to check values in registers

5	**System Software**		
Aim The aim of this subject content is to enable learners to demonstrate a theoretical understanding and practical applications of operating system and language translators.			
Learning Outcomes - The learner will:		**Assessment Criteria - The learner can:**	
1	Understanding the fundamentals of system software.	1.1	**Justify** why a computer system requires an Operating System.
		1.2	**Describe** the key management tasks carried out by the Operating System.
		1.3	**Defend** the need for utility software.
		1.4	**Discuss** the purpose and function of typical utility software.
		1.5	**Describe** the purpose of program libraries and the benefits of using a library (including DLL)
		1.6	**Organize** the purpose of an assembler, compiler and interpreter.
		1.7	**Discuss** the benefits of using a compiler and/or interpreter in a given situation.
		1.8	**Describe** the features found in an IDE.
		1.9	Explain how an OS can maximise the use of resources.
		1.10	**Show** the ways in which the user interface hides the complexities of the hardware from the user.
		1.11	**Summarize** how processes are managed by the OS.
		1.12	**Describe** the use of virtual memory, paging and segmentation for memory management.
		1.13	**Examine** how an interpreter can execute programs without producing a translated version.

		1.14	**Investigate** all various stages in the compilation of a program.
2	Be able to demonstrate a practical application of system software.	2.1	**Use** Backus-Naur Form (BNF) and syntax diagrams to express the grammar of a language.
		2.2	**Use** Reverse Polish Notation (RPN) to carry out the evaluation of expressions

6	**Security, Privacy, and Data Integrity**		
Aim The aim of this subject content is to enable learners to demonstrate a theoretical understanding and applications of data security and data integrity.			
Learning Outcomes - The learner will:		**Assessment Criteria - The learner can:**	
1	Understand the fundamentals of security, privacy, and data integrity.	1.1	**Differentiate** between security, integrity and privacy of data.
		1.2	**Describe** the threats to data and computer systems.
		1.3	**Examine** how threats can be prevented or restricted.
		1.4	**Analyse** the methods to secure data.
		1.5	**Summarise** different validation routines.
		1.6	**Describe** how verification can be used to make sure data is the same as the original.
		1.7	**Justify** how data can be verified during data entry and transfer.
		1.8	**Discuss** the key terms associated with encryption.
		1.9	**Examine** the use of encryption, symmetric and asymmetric encryption.
		1.10	**Describe** the purpose and use of SSL and TLS.
		1.11	**Describe** how digital certificates are used.

7	**Ethics and Ownership**		
Aim The aim of this subject content is to enable learners to demonstrate a theoretical understanding and applications of copyright and artificial intelligence			
Learning Outcomes - The learner will:		**Assessment Criteria - The learner can:**	
1	Understand the applications of ethnics and ownership.	1.1	**Describe** the need for ethics and to act ethically.
		1.2	**Examine** the impact of acting ethically and unethically.
		1.3	**Point out** ways a person can act ethically and/or unethically in a given situation.
		1.4	**Describe** the key features of a range of software licences.
		1.5	**Determine** the need for Artificial Intelligence (AI).
		1.6	**Summarise** the benefits and drawbacks of AI.

<table>
<tr><td><b>8</b></td><td colspan="3"><b>Databases</b></td></tr>
<tr><td colspan="4"><b>Aim</b>
The aim of this subject content is to enable learners to demonstrate a theoretical understanding and practical experience with database concepts, DBMS and SQL</td></tr>
<tr><td colspan="2"><b>Learning Outcomes - The learner will:</b></td><td colspan="2"><b>Assessment Criteria - The learner can:</b></td></tr>
<tr><td rowspan="5">1</td><td rowspan="5">Understand database concepts and database management system.</td><td>1.1</td><td><b>Identify</b> the limitations of a file-based approach.</td></tr>
<tr><td>1.2</td><td><b>Summarize</b> the features of a relational database that addresses the limitations of a file-based approach.</td></tr>
<tr><td>1.3</td><td><b>Examine</b> the normalisation process of a database.</td></tr>
<tr><td>1.4</td><td><b>Describe</b> how a DBMS addresses the limitations of a file-based approach.</td></tr>
<tr><td>1.5</td><td><b>Investigate</b> the features and software tools of a DBMS</td></tr>
<tr><td rowspan="4">2</td><td rowspan="4">Be able to demonstrate a practical application of databases.</td><td>2.1</td><td><b>Design</b> entity-relationship (E-R) diagrams to document a database design.</td></tr>
<tr><td>2.2</td><td><b>Reconstruct</b> a normalised database design for a given database description.</td></tr>
<tr><td>2.3</td><td><b>Support</b> DDL and DML commands written in SQL.</td></tr>
<tr><td>2.4</td><td><b>Create</b> SQL scripts to perform DDL and DML tasks.</td></tr>
</table>

9	**Computational Thinking, Algorithm Design, and Problem Solving**		
Aim The aim of this subject content is to enable candidates to demonstrate a theoretical understanding and practical knowledge of computational thinking skills and algorithms.			
Learning Outcomes - The learner will:		**Assessment Criteria - The learner can:**	
1	Understand theoretical concepts of computational thinking, algorithm design and problem solving.	1.1	**Discuss** the purpose of and need for abstraction.
		1.2	**Examine** the purpose of and need for decomposition.
		1.3	**Choose** appropriate identifier names.
		1.4	**Describe** how stepwise refinement can be used to express an algorithm to a level of detail from which the task may be programming.
		1.5	**Analyse** a linear and binary search.
		1.6	**Analyse** an insertion sort and a bubble sort.
		1.7	**Discuss** linked lists, stacks, queues and binary trees.
		1.8	**Describe** the use of Big O notation to specify time and space complexity.
		1.9	**Examine** algorithms on criteria such a time taken and memory used.
		1.10	**Detect** the essential features of recursion.
		1.11	**Compare** the use of recursion to iteration.
		1.12	**Appraise** what a compiler has to do to translate recursive programming code.
2	Be able to demonstrate a practical application of computational thinking algorithm design and problem solving.	2.1	**Develop** an abstract model of a system.
		2.2	**Breakdown** a problem into its sub-problems.
		2.3	**Write** programs in pseudocode using input, process, and output.

		2.4	**Create** pseudocode using assignment, sequence, selection and repetition (including logic statements).
		2.5	**Formulate** pseudocode from a structured English description and from flowchart.
		2.6	**Compose** algorithms to implement a binary and linear search.
		2.7	**Write** algorithms to implement an insertion and bubble sort.
		2.8	**Write** algorithms to find items in a linked list and a binary tree.
		2.9	**Write** algorithms to insert items into a stack, a queue, a linked list and a binary tree.
		2.10	**Create** algorithms to delete an item from a stack, a queue and a linked list.
		2.11	**Analyse** how an ADT can be implemented using a built-in data type and another ADT, and **write** algorithms to implement this
		2.12	**Create** and trace recursive algorithms.

10	**Data Types and Structures**		
Aim The aim of this subject content is to enable learners to demonstrate a theoretical understanding and practical knowledge of data types, records, arrays, files, and ADT.			
Learning Outcomes - The learner will:		**Assessment Criteria - The learner can:**	
1	Understand the concepts of data types, records, arrays, files, and abstract data types.	1.1	**Select** and use appropriate data types for a problem solution.
		1.2	**Choose** a suitable data structure (1D or 2D array) to use for a given task.
		1.3	**Judge** why files are needed.
		1.4	**Illustrate** how a queue, stack and linked list can be implemented using arrays.
		1.5	**Assess** how a stack, queue and linked list are examples of ADTs.
		1.6	**Investigate** that an ADT is a collection of data and a set of operations on those data.
2	Be able to demonstrate a practical knowledge of data types and structures.	2.1	**Use** a record structure to hold a set of different data types under one identifier.
		2.2	**Use** the technical terms associated with arrays.
		2.3	**Create** pseudocode for 1D and 2D arrays.
		2.4	**Create** pseudocode to process array data.
		2.5	**Write** pseudocode to handle text files that consist of one or more lines.
		2.6	**Use** a stack, queue and linked list to store data.

11	**Programming**		
Aim The aim of this subject content is to enable learners to demonstrate a theoretical understanding and practical knowledge of programming and structured programming.			
Learning Outcomes - The learner will:		**Assessment Criteria - The learner can:**	
1	Understand the concepts of programming.	1.1	**Justify** the purpose of the one loop structure when solving problems.
		1.2	**Analyse** the terminologies associated with procedures and functions.
		1.3	**Discuss** what is meant by a programming paradigm.
		1.4	**Examine** the terminology associated with OOP such as attributes, objects, methods.
		1.5	**Analyse** the importance of exception handling.
		1.6	**Explain** when to consider the constructor of an algorithm in terms of its appropriateness
2	Be able to demonstrate a practical application of programming.	2.1	**Use** a section of code that is repeated multiple times to
		2.2	**Write** pseudocode from a given design presented as either a program flowchart or structured English.
		2.3	**Develop** pseudocode statements for: ▪ the declaration of variables and constants ▪ the assignment of values to variables and constants ▪ expressions involving any of the arithmetic or logical operators input from the keyboard and output to the console.

		2.4	**Use** pseudocode to **produce**: ▪ an IF structure including ELSE and nested IF statements ▪ a CASE statement ▪ a count-controlled loop ▪ a post-condition loop ▪ a pre-condition loop
		2.5	**Use** parameters in a procedure and a function
		2.6	**Write** efficient pseudocode.
		2.7	**Develop** low-level code that uses various addressing modes.
		2.8	**Generate** imperative programming code that uses constructs, procedures and functions.
		2.9	**Write** low-level code that uses various addressing modes.
		2.10	**Write** imperative programming code that uses constructs, procedures and functions.
		2.11	**Create** program code to solve problems by designing appropriate classes and making use of OOP techniques.
		2.12	**Revise** and **construct** program code to solve problems by writing appropriate facts and rules.
		2.13	**Write** code to perform file-processing operations.
		2.14	**Write** program code to use exception handling.

12	**Software Development**		
Aim The aim of this subject content is to enable learners to demonstrate a theoretical understanding and practical experience with software development lifecycle, program design, testing and maintenance.			
Learning Outcomes - The learner will:		**Assessment Criteria - The learner can:**	
1	Understand the program development lifecycle.	1.1	**Analyse** the purpose of a development life cycle.
		1.2	**Appraise** the need for different development life cycles depending on the program being developed.
		1.3	**Critique** the principles, benefits and drawbacks of each type of life cycle.
		1.4	**Describe** the analysis, design, coding, testing and maintenance stages in the program development life cycle.
		1.5	**Explain** how faults in programs can be exposed and avoided.
		1.6	**Justify** the need for continuing maintenance of a system and the differences between each type of maintenance.
		1.7	**Analyse** an existing program and make amendments to enhance functionality.
2	Be able to demonstrate a practical application of software development.	2.1	**Use** a structure chart to decompose a problem into sub-tasks and express the parameters passed between the various modules, procedures or functions which are part of the algorithm design.
		2.2	**Formulate** a state-transition diagram to document an algorithm
		2.3	**Use** a state-transition diagram to document an algorithm
		2.4	**Detect** the different types of errors.
		2.5	**Rewrite** identified errors.
		2.6	**Use** different methods of testing and appropriate data for each method.

		2.7	**Choose** appropriate data for a test plan.
		2.8	**Investigate** the need for a test strategy and test plan, and their likely contents.

<table>
<tr><th>13</th><th colspan="3">Artificial Intelligence</th></tr>
<tr><td colspan="4"><b>Aim</b>
The aim of this subject content is to enable learners to demonstrate a theoretical understanding and practical experience with artificial intelligence graphs and applications.</td></tr>
<tr><td colspan="2"><b>Learning Outcomes - The learner will:</b></td><td colspan="2"><b>Assessment Criteria - The learner can:</b></td></tr>
<tr><td rowspan="5">1</td><td rowspan="5">Understand artificial intelligence graphs and applications.</td><td>1.1</td><td><b>Analyse</b> how graphs can be used to aid Artificial Intelligence.</td></tr>
<tr><td>1.2</td><td><b>Assess</b> how artificial neural networks help with machine learning</td></tr>
<tr><td>1.3</td><td><b>Evaluate</b> the use of Deep Learning, Machine Learning and Reinforcement Learning and the reasons for using these methods.</td></tr>
<tr><td>1.4</td><td><b>Justify</b> the reasons for using Deep Learning, Machine Learning and Reinforcement Learning.</td></tr>
<tr><td>1.5</td><td><b>Appraise</b> back propagation and regression methods in machine learning.</td></tr>
<tr><td rowspan="2">2</td><td rowspan="2">Be able to demonstrate a practical application of Artificial Intelligence.</td><td>2.1</td><td><b>Use</b> A* and Dijkstra’s algorithms to perform searches on a graph.</td></tr>
<tr><td>2.2</td><td><b>Create</b> a game using sequence/selection/loops-using variables/Constants/maths symbols/input/output.</td></tr>
</table>

APPENDIX

Use of calculator

Calculators are not allowed on any paper.

Programming Languages

Programming languages which LRN will accept:

- Python
- C family of languages (for example C# C+ etc.)
- Java
- Visual Basic
- PHP
- Delphi