

## Paper 1: Digital Analysis, Legislation and Emerging Issues

### Content area 1: Problem solving

Students must be able to apply problem-solving skills to analyse problems and to identify solutions that can be developed into computer programs. Students will be expected to solve realistic problems that may form a complete solution or a sub-part of a larger program. Students will be expected to use the flowchart symbols and pseudocode listed in Appendix 1.

What students need to learn		
1.1 Computational thinking		
1.1.1	Be able to use top-down, bottom-up and modularisation approaches to solve problems.	<b>E1</b> <b>M7</b>
1.1.2	Be able to decompose problems by: <ul style="list-style-type: none"><li>identifying and describing the main features of a problem or process</li><li>breaking a problem down into smaller, more manageable parts.</li></ul>	<b>E1 E2</b> <b>M7</b>
1.1.3	Be able to use pattern recognition to: <ul style="list-style-type: none"><li>identify and describe trends and similarities within and between problems and processes</li><li>identify and describe common features between a given problem and existing solutions</li><li>make predictions and assumptions based on identified patterns.</li></ul>	<b>M2 M4</b> <b>M8</b> <b>D4</b>
1.1.4	Be able to use abstraction to: <ul style="list-style-type: none"><li>identify information that is needed to solve an identified problem</li><li>filter out unnecessary details at different stages of a problem</li><li>create a layer of abstraction appropriate to the stage in the problem-solving process, including:<ul style="list-style-type: none"><li>what inputs are needed</li><li>what the expected outputs are</li><li>things that will vary</li><li>things that will remain constant</li><li>key actions the program must perform</li><li>repeated processes the program will perform.</li></ul></li></ul>	<b>E1 E5</b> <b>M10</b>

## What students need to learn

### 1.2 Algorithms

1.2.1	Understand what algorithms are and how they are expressed (flowcharts, written descriptions, pseudocode, program code).	
1.2.2	Be able to express an algorithm using flowcharts and pseudocode, and understand how to use these when planning a digital solution.	<b>M7 M10 D3 D6</b>
1.2.3	Be able to write algorithms that make use of programming constructs (sequence, selection, iteration).	<b>M7 M10 D3 D6</b>
1.2.4	Understand the purpose of a given algorithm (flowcharts, written descriptions, pseudocode, program code) and how the algorithm works.	<b>M4 M7</b>
1.2.5	Be able to determine the correct output of an algorithm.	<b>M2 M4 M5 D4</b>
1.2.6	Be able to identify and correct errors in an algorithm (flowcharts, written descriptions, pseudocode, program code).	<b>M2 M4 M5 M7 D4</b>

## Content area 2: Introduction to programming

Students should be able to apply an understanding of computer programming to solve problems. Students should be able to design, read, write and debug program code. Students will be expected to solve realistic problems that may form a complete solution or a sub-part of a larger program.

When designing a program, students will be expected to use the flowchart symbols listed in Appendix 1. Students will be expected to write, interpret and debug code in the programming language Python 3.

Students will be expected to create functions and procedures to structure and carry out programming requirements.

Students will be expected to use code development tools, including Integrated Development Environments (IDE).

When writing, interpreting and debugging code, students will be expected to understand and use the libraries, functions and methods listed in Appendix 1.

What students need to learn		
2.1 Program data		
2.1.1	Understand the use of, and need for, data types: <ul style="list-style-type: none"><li>• string</li><li>• character</li><li>• integer</li><li>• real/float</li><li>• Boolean.</li></ul>	<b>M5 D6</b>
2.1.2	Be able to declare and use constants and variables that use appropriate data types.	<b>M4 M5 D4 D6</b>
2.1.3	Understand the use of, and need for, data structures: <ul style="list-style-type: none"><li>• list</li><li>• array</li><li>• dictionary.</li></ul>	<b>M4 M5 M6 D4 D6</b>
2.1.4	Understand how to manage variables within a program, including: <ul style="list-style-type: none"><li>• use of local and global variables</li><li>• when local and global variables should be used, and why</li><li>• variable naming conventions:<ul style="list-style-type: none"><li>○ meaningful names</li><li>○ case (e.g. camelCase, UPPER CASE)</li><li>○ underscores (e.g. address_line_1).</li></ul></li></ul>	<b>M4 M5 M6 D4 D6</b>

What students need to learn		
<b>2.2 Operators</b>		
2.2.1	Understand the purpose of, and how to use, mathematical operators in program code and algorithms (add, subtract, divide, multiply, integer division, modulus).	<b>M3 M4</b>
2.2.2	Understand the purpose of, and how to use, relational operators (==, <, >, <>, <=, >=).	<b>M3 M4</b>
2.2.3	Understand the purpose of, and how to use, Boolean operators (NOT, AND, OR).	<b>M4 D6</b>
<b>2.3 File handling</b>		
2.3.1	Understand how to use text files for input and output of data.	<b>M5 D1 D4 D6</b>
<b>2.4 Program structure</b>		
2.4.1	Understand how sequence, selection (branching) and iteration are used within programs and algorithms.	
2.4.2	Be able to write, interpret and debug code that makes use of sequence: <ul style="list-style-type: none"> <li>Determine the most efficient and logical order for actions within a process.</li> <li>Understand the correct order of operations in calculations and processes, to ensure outputs are accurate and errors are avoided.</li> </ul>	<b>M2 D6</b>
2.4.3	Be able to write, interpret and debug code that makes use of selection (branching): <ul style="list-style-type: none"> <li>IF, THEN, ELSE, ELSEIF (ELIF)</li> <li>CASE.</li> </ul>	<b>M2 M4 M10 D6</b>
2.4.4	Be able to write, interpret and debug code that makes use of iteration: <ul style="list-style-type: none"> <li>Understand how 'For' loops are used to iterate code a set number of times.</li> <li>Understand how 'While' loops are used to iterate code while a set criterion is met.</li> <li>Understand how loops are used to iterate code until a set criterion is met.</li> </ul>	<b>M4 M10 D6</b>
2.4.5	Be able to declare and call functions and procedures.	<b>M4 M10 D6</b>
2.4.6	Understand how standard searching and sorting algorithms work, and the benefits and drawbacks of each: <ul style="list-style-type: none"> <li>linear and binary search</li> <li>bubble sort, insertion sort, merge sort.</li> </ul>	<b>E1 M4 M5 D4 D6</b>

What students need to learn		
2.5 Built-in functions		
2.5.1	Understand the benefits and drawbacks of using pre-written code.	<b>E4 D6</b>
2.5.2	Be able to select and justify the use of pre-written code provided by the Python programming language (e.g. built-in functions, standard libraries).	<b>E4 E5 D6</b>
2.5.3	Be able to write code that makes use of user-written and pre-written code (e.g. built-in functions, standard libraries).	<b>E4 E5 D6</b>
2.6 Validation and error handling		
2.6.1	Understand the need for different types of input validation and be able to write, interpret and debug code that makes use of these validation techniques: <ul style="list-style-type: none"> <li>• presence check</li> <li>• length check</li> <li>• type check</li> <li>• format check</li> <li>• range check</li> <li>• check digit.</li> </ul>	<b>M2 M4 M5 M6 D6</b>
2.6.2	Understand the need to develop reliable and robust code.	<b>M6 D6</b>
2.7 Maintainable code		
2.7.1	Understand the accepted style conventions (such as Python's PEP 8) and how these are implemented to create readable and maintainable code.	<b>M4 D3 D6</b>
2.8 Testing		
2.8.1	Understand the fundamental importance of testing for all components: <ul style="list-style-type: none"> <li>• software</li> <li>• hardware</li> <li>• data</li> <li>• interfaces</li> <li>• resulting service (final product).</li> </ul>	<b>D1 D6</b>

## What students need to learn

### 2.8 Testing *continued*

2.8.2	<p>Understand the use of testing and quality assurance methodologies to seek out problems and issues:</p> <ul style="list-style-type: none"> <li>• concept testing</li> <li>• unit testing</li> <li>• boundary testing</li> <li>• integration testing</li> <li>• performance testing</li> <li>• system testing</li> <li>• acceptance and usability testing</li> <li>• regression testing</li> <li>• load/stress testing.</li> </ul>	<b>E5</b> <b>M6</b> <b>D1 D6</b>
2.8.3	<p>Understand how automated and functional testing tools can be applied to test digital systems and code.</p>	<b>E5</b> <b>D1 D6</b>
2.8.4	<p>Understand how to apply root cause analysis to solve problems:</p> <ul style="list-style-type: none"> <li>• what it is (the five whys)</li> <li>• when to use it</li> <li>• how to use it</li> <li>• what next.</li> </ul>	<b>E5</b> <b>D3</b>
2.8.5	<p>Understand how to construct an effective test plan, including:</p> <ul style="list-style-type: none"> <li>• identifying tests to be carried out</li> <li>• describing the purpose of the identified test</li> <li>• identifying test data to be used (valid, valid extreme, invalid, invalid extreme, erroneous)</li> <li>• describing the expected results.</li> </ul>	<b>E1 E5</b> <b>M2</b> <b>M10</b> <b>D2 D6</b>

### Content area 3: Emerging issues and impact of digital

Students should be able to apply an understanding of ethical and moral issues in the digital sector in a range of business contexts. They should explore how developments in technology impact on organisations, individuals and society as a whole.

Students should be aware of the ever-developing nature of digital technologies, and keep up to date with knowledge of important and innovative developments in the sector.

What students need to learn		
3.1 Moral and ethical issues		
3.1.1	<p>Understand the ethical and moral issues that an increasing reliance on technology raises, and how organisations and individuals can respond to these challenges:</p> <ul style="list-style-type: none"> <li>• acceptable use</li> <li>• autonomous operation</li> <li>• changes in societal norms and the behaviour of individuals</li> <li>• changes in the culture within an organisation</li> <li>• environmental issues</li> <li>• globalisation</li> <li>• inclusion and diversity</li> <li>• monitoring of employees</li> <li>• open source and Creative Commons</li> <li>• the collection and use of data</li> <li>• unequal access to technology and/or digital services.</li> </ul>	<b>E2 E4 E5 D5</b>
3.1.2	<p>Understand how organisations and individuals respond to ethical and moral issues when designing and developing digital systems, including:</p> <ul style="list-style-type: none"> <li>• use of guidelines from professional organisations</li> <li>• strategic planning and decisions</li> <li>• the content of internal policy documents</li> <li>• company culture and how this is established, communicated and sustained</li> <li>• whistleblowing.</li> </ul>	<b>E2 E4 E5 D5</b>
3.1.3	<p>Understand how individuals use a range of observational techniques to inform situational awareness:</p> <ul style="list-style-type: none"> <li>• observing normal behaviour</li> <li>• awareness of co-workers</li> <li>• recognising changing or abnormal behaviour.</li> </ul>	<b>E2 E4 E5 D5</b>
3.2 Emerging trends and technologies		
3.2.1	<p>Understand how developments in digital technologies impact on organisations, individuals and society, including:</p> <ul style="list-style-type: none"> <li>• Internet of Things (IoT)</li> <li>• Artificial Intelligence (AI), machine learning and deep learning</li> <li>• Augmented Reality (AR) and Virtual Reality (VR).</li> </ul>	<b>E2 E4 E5 D1</b>

## Content area 4: Legislation and regulatory requirements

Students should be able to apply an understanding of legal issues in the digital sector in a range of business contexts. Students should explore how compliance with legislation impacts on the way in which organisations and their stakeholders use and interact with digital technologies.

Students should be aware of the ever developing nature of digital technologies and keep up to date with changes in legislation in response to technological developments.

What students need to learn		
4.1 Legislation		
Understand the role of current legislation and its impact on the design, development and use of digital in relation to:		
4.1.1	Health and safety when working with computers: <ul style="list-style-type: none"><li>• display screen regulations</li><li>• general working environment</li><li>• possible risks and prevention.</li></ul>	<b>E1 E2 E4 E5 D1</b>
4.1.2	Data Protection Act: <ul style="list-style-type: none"><li>• the principles of the act</li><li>• General Data Protection Regulations (GDPR)</li><li>• marketing consent</li><li>• the rights of the data subject</li><li>• enforcement.</li></ul>	<b>E1 E2 E4 E5 M6 D1 D5</b>
4.1.3	Computer Misuse Act: <ul style="list-style-type: none"><li>• the principles of the act</li><li>• consequences (company and employee)</li><li>• employee awareness.</li></ul>	<b>E1, E2, E4, E5 D1, D5</b>
4.1.4	Equality Act: <ul style="list-style-type: none"><li>• types of discrimination (protected characteristics)</li><li>• where individuals are protected</li><li>• when to take action against discrimination.</li><li>• how individuals can be discriminated against (direct, indirect, harassment and victimisation).</li></ul>	<b>E1 E2 E4 E5 D1 D5</b>
4.1.5	Intellectual Property Act: <ul style="list-style-type: none"><li>• unregistered designs</li><li>• registered designs</li><li>• patents.</li></ul>	<b>E1 E2 E4 E5 D1 D5</b>



What students need to learn		
4.1 Legislation <i>continued</i>		
4.1.6	Understand the use of digital technologies for monitoring the workplace: <ul style="list-style-type: none"> <li>• monitoring electronic communications</li> <li>• use of secret monitoring</li> <li>• employers' monitoring policies</li> <li>• monitoring systems.</li> </ul>	<b>E1 E2 E4 E5 D1 D5</b>
4.1.7	Understand the role of legislation relating to international law and its importance when designing, developing and using digital systems.	<b>E1 E2 E4 E5 D1 D5</b>
4.2 Guidelines and codes of conduct		
4.2.1	Understand the purpose and role of codes of conduct produced by professional bodies for the use of digital: <ul style="list-style-type: none"> <li>• British Computer Society (BCS) Code of Conduct</li> <li>• The Institution of Analysts and Programmers Code of Conduct.</li> </ul>	<b>E1 E2 E4 E5 D1 D5</b>
4.2.2	Understand the guidelines provided in professional codes of practice in terms of: <ul style="list-style-type: none"> <li>• professional responsibilities (quality of work, meeting deadlines, communication, confidentiality, trust)</li> <li>• contribution to society</li> <li>• safety</li> <li>• security and privacy</li> <li>• innovation.</li> </ul>	<b>E5 D5</b>
4.2.3	Understand the impact that implementing guidelines from professional codes of practice has on organisations and their stakeholders.	<b>E4 E5 D5</b>
4.2.4	Understand how guidelines and agreed standards ensure the accessibility and quality of IT systems, including: <ul style="list-style-type: none"> <li>• ISO (international Standards Organisation) standards</li> <li>• Web Content Accessibility Guidelines (WCAG) 1.0 and 2.0</li> <li>• World Wide Web Consortium (W3C®)</li> <li>• Internet Engineering Task Force (IETF).</li> </ul>	<b>E4 E5 D1 D5</b>
4.2.5	Understand the role and implications of acceptable use policies within an organisation.	<b>E4 E5 D5</b>