# Core Component

The content of the Core component has the core skills mapped to where there are opportunities to develop them. The competencies and skills are not expected to be developed at every point where they are mapped, but using this guidance teachers will embed them into teaching to prepare students for the assessments in the Core component.

The core skills are assessed through the Employer Set Project. The core skills for this Core component are as follows:

1. Be able to reflectively evaluate
2. Communicate information clearly to a technical and non-technical audience
3. Work with others in a collaborative manner to allow for/encourage faster, better and more efficient achievement of goals
4. Develop software/Create an artefact
5. Apply a logical approach to solving problems to:
   * identify and fix defects
   * propose software solutions
6. Ensure software development activity mitigates risks to security.

## Content

### Core paper 1

|  |  |
| --- | --- |
| **Content area 1: Problem solving** | |
| Students will solve digital software development problems that form a complete solution or a sub-part of a solution.  Students will use problem-solving skills to analyse problems and to identify solutions that can be represented as systems, processes, relationships, organisations of data or code.  Where the term ‘code’ is used, this refers to Python 3.10 or later. | |
| **1.1 Computational thinking** | |
| 1.1.1 | Know the definition and understand the purpose of computational thinking. |
| 1.1.2 | Know when to use computational thinking. |
| 1.1.3 | Know and understand the benefits and drawbacks of using computational thinking. |
| 1.1.4 | Know the components of computational thinking:   * decomposition * pattern recognition * abstraction * algorithmic design. |
| 1.1.5 | Know and understand the benefits and drawbacks of using the components of computational thinking. |
| 1.1.6 | Know and understand the purpose of decomposition. |
| 1.1.7 | Know the tasks of decomposition:   * identify the main features of a problem * characterise each identified feature * break problems down into smaller, more manageable parts * break solutions down into smaller, more manageable parts. |
| 1.1.8 | Be able to use decomposition for problem solving. |
| 1.1.9 | Know and understand methods to represent decomposition:   * block diagrams * information flow diagrams * flowcharts * code * written descriptions. |
| 1.1.10 | Be able to use the methods to represent decomposition. |

|  |  |
| --- | --- |
| 1.1.11 | Know and understand the purpose of pattern recognition. |
| 1.1.12 | Be able to use pattern recognition for problem solving:   * find and interpret trends and similarities within and between problems and processes * find and interpret common features between a given problem and existing solutions * make predictions and assumptions based on identified patterns. |
| 1.1.13 | Know and understand the purpose of abstraction. |
| 1.1.14 | Know and understand the tasks of abstraction:   * identify information that is needed * filter out unnecessary details * hide details of internal workings. |
| 1.1.15 | Be able to use abstraction:   * what inputs are needed * what the expected outputs and outcomes are * things that will vary * things that will remain constant * key actions the solution must perform * repeated processes the solution will perform. |
| 1.1.16 | Be able to use abstraction in problem solving. |
| 1.1.17 | Understand the interrelationships between components of computational thinking and make judgements about the suitability of using the components in digital software development. |
| **1.2 Algorithmic design** | |
| 1.2.1 | Know the definition and understand the characteristics and purpose of algorithms. |
| 1.2.2 | Know and understand that algorithms can be expressed in:   * flowcharts:   + terminators   + processes   + sub-processes   + decisions   + inputs/outputs   + arrows   + labels * written descriptions using hierarchical markers to indicate sequence * code (commands in *Appendix 2*). |
| 1.2.3 | Know and understand the benefits and drawbacks of the ways of expressing algorithms in flowcharts. |
| 1.2.4 | Know and understand the benefits and drawbacks of the ways of expressing algorithms in written descriptions. |

|  |  |
| --- | --- |
| 1.2.5 | Know and understand the benefits and drawbacks of the ways of expressing algorithms in code. |
| 1.2.6 | Know and understand actions to control ordering of steps in algorithms:   * sequence * selection * iteration. |
| 1.2.7 | Be able to determine the purpose of an algorithm and how it works. |
| 1.2.8 | Be able to determine the output of an algorithm given an input. |
| 1.2.9 | Be able to identify errors in an algorithm. |
| 1.2.10 | Be able to correct errors in an algorithm. |
| 1.2.11 | Be able to translate between the different notations for algorithms. |
| 1.2.12 | Be able to design algorithms and solutions that use actions. |
| **1.3 Strategies** | |
| 1.3.1 | Know the different approaches to solving problems and understand their purpose and when they are used:   * top-down * bottom-up * modularisation. |
| 1.3.2 | Know the benefits and drawbacks of using the different approaches to solving problems. |
| 1.3.3 | Understand the purpose of root cause analysis and when it is used. |
| 1.3.4 | Know and understand approaches to root cause analysis:   * five whys * failure mode and effects analysis (FMEA) * event tree analysis (ETA) * actions to take after using root cause analysis:   + log   + close   + escalate to an appropriate manager, specialist or external third party. |
| 1.3.5 | Know and understand the process of the high-level problem-solving strategy:   * define the problem * gather information * analyse the information * make a plan of action * implement a solution * review the solution. |

|  |  |
| --- | --- |
| 1.3.6 | Understand the interrelationships between problems and problem- solving strategies and make judgements about the suitability of strategies for solving the problems in digital software development. |

|  |  |
| --- | --- |
| **Content area 2: Introduction to programming** | |
| Students will analyse digital software development problems that may involve software, people, processes and data.  Students will use a variety of tools and techniques when developing a complete solution or a sub-part of a solution.  Where the term ‘code’ is used, this refers to Python 3.10 or later. | |
| **2.1 Standard data types** | |
| 2.1.1 | Know the definition of common data types and understand their purpose and when each is used:   * integer * float * string * Boolean. |
| **2.2 Variables and constants** | |
| 2.2.1 | Know the definition of variables, understand their purpose and when they are used. |
| 2.2.2 | Know the definition of constants, understand their purpose and when they are used. |
| 2.2.3 | Understand the purpose of data type conversion functions and why they are used.  See *Appendix 2* for functions. |
| 2.2.4 | Know a definition of scope and understand the role of scope and when it is used. |
| 2.2.5 | Know and understand how variables are managed by scope:   * global variables * local variables. |
| 2.2.6 | Be able to use scope. |
| 2.2.7 | Be able to declare variables and constants using standard data types. |
| 2.2.8 | Be able to use variables and constants. |
| 2.2.9 | Be able to use data type conversion functions. |
| **2.3 Data structures** | |
| 2.3.1 | Know the standard data structures and understand their purpose and when each is used:   * list * array * dictionary. |
| 2.3.2 | Be able to interpret code using data structures. |

|  |  |
| --- | --- |
| 2.3.3 | Be able to develop code using data structures. |
| 2.3.4 | Be able to debug code using data structures. |
| **2.4 Operators** | |
| 2.4.1 | Understand the purpose of arithmetic operators:   * add * subtract * divide * multiply * exponentiation * integer division * modulus.   See *Appendix 2* for operators. |
| 2.4.2 | Understand the purpose of relational operators:   * equivalence * less than * greater than * not equal * less than or equal to * greater than or equal to. See *Appendix 2* for operators. |
| 2.4.3 | Understand the purpose of Boolean operators:   * not * and * or.   See *Appendix 2* for operators. |
| 2.4.4 | Be able to use operators:   * arithmetic * relational * Boolean. |
| 2.4.5 | Be able to interpret code using operators. |
| 2.4.6 | Be able to create code using operators. |
| 2.4.7 | Be able to debug code using operators. |
| **2.5 Input and output** | |
| 2.5.1 | Understand how to implement input and output:   * keyboard * screen * text file. |

|  |  |
| --- | --- |
| 2.5.2 | Know and understand the use of text files for input and output:   * open a file for reading * open a file for writing * write lines to the file * close the file. |
| 2.5.3 | Be able to interpret code using input and output. |
| 2.5.4 | Be able to create code using input and output. |
| 2.5.5 | Be able to debug code using input and output. |
| **2.6 Actions** | |
| 2.6.1 | Know and understand sequence, its purpose and when it is used. |
| 2.6.2 | Know and understand selection, its purpose and when it is used:   * if * else if * else * match/case.   See *Appendix 2* for key words. |
| 2.6.3 | Know and understand loops, their purpose and when they are used:   * count-controlled loop * condition-controlled loop. |
| 2.6.4 | Know and understand iteration:   * count-controlled ‘for’ loops * condition-controlled ‘while do’ loops. |
| 2.6.5 | Know and understand the benefits and drawbacks of loops. |
| 2.6.6 | Be able to interpret code using actions:   * sequence * selection * iteration. |
| 2.6.7 | Be able to develop code using actions:   * sequence * selection * iteration. |
| 2.6.8 | Be able to debug code using actions:   * sequence * selection * iteration. |

|  |  |
| --- | --- |
| **2.7 Functions and procedures** | |
| 2.7.1 | Know characteristics of functions, understand their purpose and when they are used:   * may or may not take parameters * must return a result. |
| 2.7.2 | Know characteristics of procedures, understand their purpose and when they are used:   * may or may not take parameters * must not return a result. |
| 2.7.3 | Know and understand where functions and procedures come from:   * user-written * pre-written and built-in as part of the programming language * pre-written and supplied in libraries as part of the programming language * pre-written and supplied in libraries from third parties. |
| 2.7.4 | Know and understand the benefits and drawbacks of using pre-written code. |
| 2.7.5 | Be able to interpret code using user-written and pre-written code:   * functions * procedures. |
| 2.7.6 | Be able to develop code using user-written and pre-written code:   * functions * procedures. |
| 2.7.7 | Be able to debug code using user-written and pre-written code:   * functions * procedures. |
| **2.8 Validation** | |
| 2.8.1 | Know the definition of validation and understand the purpose and when validation checks are used:   * presence check * length check * range check * type check * format check * check digit. |
| 2.8.2 | Be able to interpret code using validation. |
| 2.8.3 | Be able to develop code using validation. |
| 2.8.4 | Be able to debug code using validation. |

|  |  |
| --- | --- |
| **2.9 Design considerations and programming practices** | |
| 2.9.1 | Determine the logical order for actions within processes. |
| 2.9.2 | Determine the order of operations in calculations and processes:   * to ensure outputs are accurate * to ensure errors are avoided. |
| 2.9.3 | Determine the selection of data structures:   * to ensure efficiency of execution time * to ensure efficient use of memory. |
| 2.9.4 | Determine the order of actions:   * to ensure efficiency of execution time * to ensure efficient use of memory. |
| 2.9.5 | Know and understand naming conventions and their purpose:   * meaningful names * camelCase * snake\_case. |
| 2.9.6 | Understand the impact on readability of code style conventions:   * naming conventions * use of white space * maximum line length. |
| 2.9.7 | Make judgements about the suitability of an algorithm in meeting requirements, efficiency in use of storage and execution time, appropriateness in the choice of data structures, data types, variables, constants, presentation and maintainability. |
| **2.10 Robust code** | |
| 2.10.1 | Know and understand the characteristics of robust code:   * handles unexpected inputs * handles unexpected terminations * produces specific and meaningful error messages. |
| 2.10.2 | Know and understand the process of debugging and when it is used:   * locating errors in code * correcting errors in code. |
| 2.10.3 | Understand the role of debugging in producing solutions that are robust. |
| 2.10.4 | Be able to locate errors in code. |
| 2.10.5 | Be able to correct errors in code. |
| **2.11 Common algorithms** | |
| 2.11.1 | Know and understand the algorithms for searching, how they work and when they are used:   * linear search * binary search. |

|  |  |
| --- | --- |
| 2.11.2 | Know and understand the algorithms for sorting, how they work and when they are used:   * bubble sort * insertion sort * merge sort. |
| 2.11.3 | Know and understand the benefits and drawbacks of using:   * linear search * binary search * bubble sort * insertion sort * merge sort. |
| 2.11.4 | Know and understand metrics to compare algorithms:   * use of memory space * execution time * number of comparisons. |
| 2.11.5 | Know and understand best case, worst case and average case for common algorithms, using logical reasoning (Big O not required). |
| 2.11.6 | Make judgements about the suitability of using different algorithms for searching. |
| 2.11.7 | Make judgements about the suitability of using different algorithms for sorting. |
| **2.12 Testing** | |
| 2.12.1 | **Testing components** |
| 2.12.1.1 | Understand the reasons for testing individual components of a solution before putting them together in the final solution:   * software * hardware * data * interfaces * resulting service (final product). |
| 2.12.2 | **Testing methods** |
| 2.12.2.1 | Know a definition of testing methods and understand their purpose, benefits and drawbacks, and when they are used:   * concept * unit * boundary * integration * performance * system * acceptance * usability |

|  |  |
| --- | --- |
|  | * regression * load/stress * closed box * open box. |
| 2.12.2.2 | Be able to use testing methods. |
| 2.12.3 | **Automation** |
| 2.12.3.1 | Know and understand the purpose of automation methods and when they are used:   * macros * scripts * functional testing tools. |
| 2.12.4 | **Test data and test plan** |
| 2.12.4.1 | Know a definition for types of test data and understand the purpose of test data and when it is used:   * valid * invalid * boundary * erroneous. |
| 2.12.4.2 | Be able to create test data. |
| 2.12.4.3 | Know and understand the steps and structure of a test plan and when it is used:   * identifying tests to be carried out * describing the purpose of the identified test * identifying test data to be used * describing the expected results * recording actual results. |

|  |  |
| --- | --- |
| **Content area 3: Emerging issues** | |
| **3.1 Impact of digital technologies** | |
| 3.1.1 | Understand how the increased reliance on digital systems impacts:   * organisational culture:   + changes in communication methods (face to face, email, video calls)   + increased productivity and availability expectations   + increase in staff monitoring   + new working practices (remote/hybrid/in-office working)   + automation of services including the use of artificial intelligence (AI) * society:   + loss of jobs   + shift in skill requirements   + reduction in human decision making and loss of empathy |

|  |  |
| --- | --- |
|  | * privacy (digital footprint, surveillance) * changing behaviours (loss of social skills, digital identity) * access to wider social networks (personal and professional) * access to online services (government, commercial and entertainment) * potential isolation (lack of skill, equipment, connectivity, resistance to change) * improved access to information (professional and personal) * increased use of AI including generative AI (textual, graphical, video and audio) * globalisation:   – access to global media sources. |
| 3.1.2 | Understand the importance of digital inclusion:   * ensuring fair access to digital services:   + suitable technologies (hardware and software)   + connectivity   + checking for bias within datasets   + conforming to codes of best practice   + public sector bodies’ website and mobile applications accessibility regulations:   – key features and purpose. |
| 3.1.3 | Understand how end user characteristics affect the use of and inclusivity of digital systems:   * age * skills:   + digital   + literacy * internal/external audience * cultural issues, including bias in digital systems * additional needs:   + accessibility issues. |
| 3.1.4 | Know and understand the benefits of professional development:   * increased industry and sector competence * increased employability potential and employment security * achieving access to knowledge of and adherence to industry standards. |
| **3.2 Emerging technologies** | |
| 3.2.1 | Understand how developments in technologies impact organisations, individuals and society:   * storage media:   + increased demand for storage * processing technologies:   + quantum computing |

|  |  |
| --- | --- |
|  | * Internet of Things (IoT):   + edge computing   + use within different contexts (industrial, smart city, domestic) * artificial intelligence:   + generative AI   + machine learning * extended reality:   + augmented reality   + virtual reality * open source software * blockchain * environmental:   + consumption of rare metals   + energy to produce electronic systems   + environmental impact of disposal * autonomous machines:   + self-driving cars   + robotic assembly lines. |
| 3.2.2 | Understand the interrelationships between digital and emerging technologies and make judgements about their impacts on organisations, society and individuals in digital software development. |

|  |  |
| --- | --- |
| **Content area 4: Legislation and regulatory requirements** | |
| **4.1 Legislation** | |
| 4.1.1 | Understand the key points and implications to employers of the relevant health and safety legislation:   * Health and Safety at Work Act   + key points:     - provide a safe working environment     - ensure staff are properly trained     - adequate welfare provision     - provide relevant information, instruction and supervision * display screen equipment.   + implications for employers:     - conduct a display screen equipment workstation assessment     - reduce risks including making sure workers take breaks from display screen equipment work     - provide an eye test if an employee asks for one     - provide training and information for employees. |

|  |  |
| --- | --- |
| 4.1.2 | Understand the health and safety risks and preventative measures of working with digital systems:   * possible risks:   + using display screen equipment   + health and safety requirements * methods of mitigating risk   + adequate training   + safe working environment   + safe working practices. |
| 4.1.3 | Understand data security and protection legislation including the effect on organisations and individuals:   * Data Protection Act/General Data Protection Regulation:   + purpose of legislation   + eight principles. |
| 4.1.4 | Understand computer misuse legislation:   * the principles of the Computer Misuse Act (CMA) 1990 * consequences for company and employee * employee awareness * types of crime covered by legislation. |
| 4.1.5 | Understand equality legislation:   * the nine protected characteristics * types of discrimination:   + direct   + indirect   + harassment   + victimisation * where individuals are protected * when to take action against discrimination:   + time limits for claims. |
| 4.1.6 | Understand intellectual property legislation:   * unregistered designs * registered designs * patents. |
| 4.1.7 | Understand the interrelationships between digital software development and digital legislation, and make judgements about the impact on organisations, society and individuals. |
| 4.1.8 | Know that international law applies to some offences:   * international law in cyberspace * international law and surveillance. |

|  |  |
| --- | --- |
| **4.2 Guidelines** | |
| 4.2.1 | Know the sources of codes of conduct:   * organisational * professional:   + British Computer Society (BCS)   + The Institution of Analysts and Programmers (IAP)   + Chartered Institute of Information Security (CIISec) * governmental. |
| 4.2.2 | Understand how guidelines in codes of conduct influence professional behaviour:   * ensuring individuals follow policies, procedures and legislation * ensuring quality of work:   + minimising risk to the public   + acting with competence and integrity * meeting deadlines * effective communication * maintaining confidentiality and trust. |
| 4.2.3 | Know the sources of digital industry standards:   * International Organization for Standardization (ISO) * Web Content Accessibility Guidelines (WCAG) * World Wide Web Consortium (W3C®) * Internet Engineering Task Force (IETF) * British Standard (BS) * Institute of Electrical and Electronics Engineers (IEEE) * Payment Card Industry Security Standards Council (PCI SSC). |
| 4.2.4 | Understand the purpose of acceptable use policies (AUP):   * purpose of AUP * typical content:   + permitted activities   + prohibited activities   + working practices including confidentiality   + communication etiquette including projecting correct organisation image   + sanctions/penalties. |
| 4.2.5 | Understand the importance of whistleblowing procedures. |
| 4.2.6 | Understand the interrelationships between digital software development and guidelines, and make judgements about the impact on organisations, society and individuals. |

### Core paper 2

|  |  |
| --- | --- |
| **Content area 5: Business context** | |
| **5.1 Business environment** | |
| 5.1.1 | Know the purpose and sectors of different types of organisations:   * purpose of the organisation:   + providing a service   + providing a product * private sector:   + small or medium-sized enterprise (SME)   + large enterprise   + non-governmental organisation (NGO) * public sector * voluntary/charity:   + not for profit. |
| 5.1.2 | Know the names and definitions of different business models:   * Business to Customer (B2C) * Business to Business (B2B) * Business to Many (B2M). |
| 5.1.3 | Know the different types of stakeholders:   * internal stakeholders:   + owners   + directors   + employees * external stakeholders:   + customers/clients   + suppliers   + shareholders   + outsourced services   + investors/funders   + government. |

|  |  |
| --- | --- |
| **5.2 Digital value to organisations** | |
| 5.2.1 | Understand how digital systems are used to support key organisation areas:   * sales and Marketing:   + better market research   + better brand promotion, including social media   + online selling   + contextualising customer behaviour to personalise services offered   + better customer retention   + brand differentiation and values   + use of analytic tools, including search and social media analytics * research, design and development:   + provision of unique products and services * Human Resources:   + staff records   + performance management   + training records * operations:   + enhanced internal communication   + automation of internal processes   + automated manufacturing   + remote working   + intranet/shared workspace   + document sharing and online shared storage * management:   + real-time monitoring of key performance indicators:     - sales     - customers served     - units manufactured     - real-time location of assets * logistics:   + automated stock control * finance:   + reduced costs   + increased revenue   + better financial reporting via up-to-date information. |

|  |  |
| --- | --- |
| 5.2.2 | Understand how digital systems are used to meet user needs and ensure quality of product/service:   * appropriate and effective functionality:   + allows users to do all required tasks * reduction of pain points:   + response time (communication of expected response time, notification of change in response time)   + complexity of task * appropriate accessibility provision * compatibility:   + with internal legacy systems   + with proposed future systems   + with external services * availability of service:   + minimise downtime   + future proofing for upgrades * effective end user support:   + provision of digital support * ease of installation:   + provision of installation package. |
| **5.3 Risk to organisations of using digital systems** | |
| 5.3.1 | Understand the potential risks to organisations when using digital systems:   * security breaches:   + compromised confidentiality   + loss of integrity   + reduced availability * privacy breaches:   + personal information   + business information * regulatory and legal non-compliance * audience exclusion:   + biases   + poor user experience * emerging rival technologies * technical issues:   + reliance and system failure   + system not fit for purpose. |

|  |  |
| --- | --- |
| 5.3.2 | Understand the potential impact of risks to organisations when using digital systems:   * legal action * fines * reputational damage * withdrawal of licence to practise * loss of business. |
| **5.4 Technical change management** | |
| 5.4.1 | Understand the internal factors that trigger change in organisations:   * internal factors:   + organisational restructuring   + expansion   + downsizing   + new strategic objectives:     - diversification     - rebranding     - additional features or services. |
| 5.4.2 | Understand the external factors that trigger change in organisations:   * political:   + change in government   + conflict   + shift in government priorities * economic:   + provision of new services   + recession   + inflation   + interest rates   + consumer trends   + new competitors   + entering new markets * social:   + changes in demographics   + market/social trends   + adapting to remote working   + cultural expectations * technological:   + emergence of new technologies   + retirement of obsolete technologies   + system failure   + zero-day vulnerabilities |

|  |  |
| --- | --- |
|  | * legal:   + new legislation   + changes to legislation * environmental:   + sustainability issues   + pandemics   + natural disasters. |
| 5.4.3 | Understand how organisations can respond to change:   * new or amended policies * new or amended business processes:   + change in staffing numbers   + change in delivery schedules   + change in opening hours * new or amended products or services:   + completely new products or services   + next generation products or services   + minor updates to existing products or services * new or improved digital systems:   + back-end systems   + customer-facing systems * improved training * restructuring:   + change in management structure   + redrawing of boundaries. |
| 5.4.4 | Understand the processes, benefits and drawbacks of the change management process:   * identifying type of change:   + new system   + amendment to existing system * role of change advisory board (CAB):   + prioritise change requests   + review change requests   + stages of approval   + monitor change process:     - collate and analyse change data     - check change implementation     - take action to accelerate change   + provide feedback * identifying the changes to be made:   + using SMARTER objectives (specific, measurable, achievable, realistic, time-bound, evaluated, reviewed) |

|  |  |
| --- | --- |
|  | * identifying impact of change:   + measure/forecast positive and negative impact   + analysis of positive and negative impact * allocation of resources:   + budget   + time   + staffing   + hardware and software * identifying and communicating potential risks and desired impact(s) to stakeholders:   + gain acceptance   + ensure compliance * configuration of the new system or process:   + integration with legacy systems   + maintaining service during change * importance of fully testing new systems:   + reproducibility of results   + test environment, including hardware and software * method of implementing change:   + parallel   + phased   + direct   + pilot * documenting the change process:   + ensuring requirement traceability, including responsibility and accountability   + maintaining up-to-date information   + recording of all decisions   + retaining change documentation   + user training manuals * importance of rollback planning:   + backup methodology   + backup location   + recover plan * identify training needs:   + new training requirements   + refresher courses * identify methods of monitoring progress:   + post-progress review * version control software. |

|  |  |
| --- | --- |
| 5.4.5 | Understand the factors that determine the feasibility of a digital project:   * benefits and drawbacks:   + financial savings   + cost of implementing change   + impact on processes, including productivity gains, improved communication and security   + provision of new products   + impact on company reputation * risks:   + resistance to change from workforce   + misuse of new systems   + inadequate support for new system   + inadequate knowledge of new system   + disruption caused by implementation of new systems * constraints:   + budget   + time   + human resources and technological resources. |

|  |  |
| --- | --- |
| **Content area 6: Data** | |
| Students will develop fundamental knowledge and understanding of data relevant to digital software development in order to communicate with other professionals.  Students will understand how to store, access, quality assure, manipulate, analyse and process data. | |
| **6.1 Data, information and knowledge** | |
| 6.1.1 | Know and understand the differences and relationships between:   * data * information * knowledge. |
| 6.1.2 | Know and understand sources for generating data:   * humans: surveys, forms * artificial intelligence (AI)/machine learning: dangers of feedback loop * sensors: temperature, accelerometer, vibration, sound, light, pressure * Internet of Things (IoT): smart objects (thermostats, lights, security camera, trackers * transactions: customer data, membership, timing, basket. |
| 6.1.3 | Know and understand ethical data practices and the metrics to determine the value of data:   * quantity * timeframe * source * veracity. |

|  |  |
| --- | --- |
| 6.1.4 | Understand how organisations use data and information:   * analysis to identify patterns * system performance analysis: load, outage, throughput, status * user monitoring: login/logout, resources accessed * targeted marketing: discounts, upselling * threat/opportunity assessment: competitors, security, compliance. |
| 6.1.5 | Understand the interrelationships between data, information and the way it is generated, and make judgements about the suitability of data, information and the way it is generated in digital software development. |
| **6.2 Methods of transforming data** | |
| 6.2.1 | Know and understand methods of transforming data:   * manipulating * analysing * processing. |
| **6.3 Data taxonomy** | |
| 6.3.1 | Know the definition of each category, understand its purpose, and understand that data is categorised as:   * quantitative * qualitative. |
| 6.3.2 | Know the definition for structured data, understand its purpose and understand that quantitative data is structured. |
| 6.3.3 | Know the definition for unstructured data, understand its purpose and understand that qualitative data is unstructured. |
| 6.3.4 | Know the definition for each representation and understand the representation of quantitative data:   * discrete values * continuous values * categorical values. |
| 6.3.5 | Know and understand the properties of qualitative data:   * stored and retrieved only as a single object * codified into structured data. |
| 6.3.6 | Understand the interrelationships between data categories, data structure and transformation, and make judgements about the suitability of data categories, data structure and transformation in digital software development. |

|  |  |
| --- | --- |
| **6.4 Data types** | |
| 6.4.1 | Know the definition of common data types and understand their purpose and when each is used:   * integer * real * character * string * Boolean * date * Blob. |
| 6.4.2 | Understand the interrelationships between structured data, unstructured data and data type. |
| 6.4.3 | Understand the interrelationships between data type and data transformation. |
| 6.4.4 | Be able to make judgements about the suitability of using structured data, unstructured data, data types and data transformations in digital software development. |
| **6.5 Data formats** | |
| 6.5.1 | Know the definition of common data formats and understand their purpose and when each is used:   * JSON * Text file * CSV * UTF-8 * ASCII * XML. |
| 6.5.2 | Understand the interrelationships between data format and data transformation and make judgements about the suitability of using data formats in digital software development. |
| **6.6 Structures for storing data** | |
| 6.6.1 | Understand the role of metadata in providing descriptions and contexts for data. |
| 6.6.2 | Know the definition of file-based and directory-based structures and understand their purposes and when they are used. |
| 6.6.3 | Know the definition of hierarchy-based structure and understand its purpose and when it is used. |
| 6.6.4 | Understand the interrelationships between storage structures and data transformation. |

|  |  |
| --- | --- |
| **6.7 Data dimensions and maintenance** | |
| 6.7.1 | Know the definitions of the six Vs (dimensions) and understand the six Vs (dimensions) of Big Data and their impact on gathering, storing, maintaining and processing:   * volume * variety * variability * velocity * veracity * value. |
| 6.7.2 | Know the definition of Big Data and understand that it has multiple dimensions. |
| 6.7.3 | Understand the impact of each dimension on how data is gathered and maintained. |
| 6.7.4 | Know the definitions of data quality assurance methods and understand their purpose and when each is used:   * validation * verification * reliability * consistency * integrity * redundancy. |
| 6.7.5 | Know and understand factors that affect how data is maintained:   * time * skills * cost. |
| 6.7.6 | Understand the interrelationships between the dimensions of data, data quality assurance methods and factors that impact how data is maintained, and make judgements about the suitability of maintaining, transforming and quality assuring data in digital software development. |
| **6.8 Data systems** | |
| 6.8.1 | Know the definition of data wrangling and understand its purpose and when it is used. |
| 6.8.2 | Know and understand the purpose of each step of data wrangling:   * structure * clean * validate * enrich * output. |

|  |  |
| --- | --- |
| 6.8.3 | Know and understand the purpose of each core function of a data system:   * input * search * save * integrate * organise (index) * output * feedback loop. |
| 6.8.4 | Know the types of data entry errors and understand how and why they occur:   * transcription errors * transposition errors. |
| 6.8.5 | Know and understand methods to avoid data entry errors:   * validation of user input * verification of user input by double entry * drop-down menus * pre-filled data entry boxes. |
| 6.8.6 | Know and understand the factors that impact data entry:   * time needed to create the screens * expertise needed to create screens * time needed to enter the data. |
| 6.8.7 | Understand the relationship between factors that impact data entry and data quality and make judgements about the suitability of methods to reduce data entry errors in digital software development. |
| 6.8.8 | Understand the relationship between factors that impact implementation of data entry and make judgements about the suitability of implementing data entry in digital software development. |
| **6.9 Data visualisation** | |
| 6.9.1 | Know and understand data visualisation formats and when they are used:   * graphs * charts * tables * reports * dashboards * infographics. |

|  |  |
| --- | --- |
| 6.9.2 | Know and understand the benefits and drawbacks of data visualisation formats based on:   * type of data * intended audience * brief. |
| **6.10 Data models** | |
| 6.10.1 | Know the types of data models and understand how they organise data into structures:   * hierarchical * network * relational. |
| 6.10.2 | Know and understand the factors that impact the selection of data models for organising data:   * efficiency of accessing individual items of data * efficiency of data storage * level of complexity in implementation. |
| 6.10.3 | Understand the benefits and drawbacks of different data models and make judgements about the suitability of data models based on efficiency and complexity. |
| 6.10.4 | Be able to draw and represent data models:   * hierarchical models with blocks, arrows and labels * network models with blocks, arrows and labels * relational models with tables, rows, columns and labels. |
| **6.11 Data access across platforms** | |
| 6.11.1 | Understand the features, purposes, benefits and drawbacks of accessing data across platforms:   * permissions: * authorisation * privileges * access rights * rules * access mechanisms:   + role-based access (RBAC)   + rule-based access control (RuBAC)   + Application Programming Interfaces (API). |
| 6.11.2 | Know and understand the benefits and drawbacks of methods to access data across platforms. |
| 6.11.3 | Understand the interrelationships between data access requirements and data access methods and make judgements about the suitability of accessing data in digital software development. |

|  |  |
| --- | --- |
| **6.12 Data analysis tools** | |
| 6.12.1 | Know data analysis tools and understand their purpose and when they are used:   * storing Big Data for analysis:   + data warehouse   + data lake   + data mart * analysis of data:   + data mining   + reporting * use of business intelligence gained through analysis:   + financial planning and analysis   + customer relationship management (CRM):     - customer data analytics     - communications. |
| 6.12.2 | Understand the interrelationships between data analysis tools and the scale of data. |

|  |  |
| --- | --- |
| **Content area 7: Digital environments** | |
| **7.1 Hardware** | |
| 7.1.1 | Understand the features and use of different types of physical computers:   * personal computers * mobile devices (smartphones and tablets) * servers * embedded devices. |
| 7.1.2 | Understand the features and use of different types of hardware devices:   * input devices * output devices * processors:   + number of cores   + clock speed   + cache size   + mobile processors * main memory:   + RAM (Random Access Memory)   + ROM (Read-only Memory) * secondary storage:   + magnetic   + solid state   + optical * motherboard |

|  |  |
| --- | --- |
|  | * graphics processing units * network interface devices:   + PCI (Peripheral Component Interconnect)   + USB (Universal Serial Bus) * cooling:   + air cooling   + liquid cooling * sensors. |
| **7.2 Software** | |
| 7.2.1 | Understand the features and use of operating systems:   * batch:   + non-interactive applications   + high volume   + scheduling * multitasking:   + concurrent execution of multiple tasks   + time-slicing   + interrupts * real-time operating system:   + monitoring and control applications   + transaction processing * network operating system:   + resource sharing   + user management   + communication * mobile operating system:   + smartphones and tablets   + lower processing requirements   + increased battery life. |
| 7.2.2 | Understand the features and use of common utilities:   * file management * defragmenters * file compression * package managers * protection software * backup software. |

|  |  |
| --- | --- |
| 7.2.3 | Understand the features and use of common code development tools:   * integrated development environments:   + code editing tools   + debugging tools   + screen design tools * compilers * interpreters. |
| 7.2.4 | Understand the features and use of common application software:   * word processors * spreadsheets * databases * email * project management software. |
| **7.3 Networks** | |
| 7.3.1 | Understand the benefits and drawbacks of connecting devices to form networks. |
| 7.3.2 | Understand the features of different types of networks:   * number of users * connection media * coverage media * network types:   + personal area network (PAN)   + local area network (LAN)   + metropolitan area network (MAN)   + wide area network (WAN)   + virtual private network (VPN). |
| 7.3.3 | Understand the features, characteristics, benefits and drawbacks of connectivity methods:   * wired:   + copper/ethernet   + fibre-optic * wireless:   + wireless access points. |
| 7.3.4 | Understand the features, benefits and drawbacks of the common network topologies:   * star * mesh * tree * logical versus physical. |

|  |  |
| --- | --- |
| 7.3.5 | Understand the features, benefits and drawbacks of different network models:   * client-server * thin client * peer-to-peer. |
| 7.3.6 | Understand the role of common components of a network:   * server * client * router * network switch * internet connection/internet backbone. |
| 7.3.7 | Understand the seven-layer OSI (open systems interconnection) model, including the function and related protocols of each layer:   * application layer * presentation layer * session layer * transport layer * network layer * data link layer * physical layer. |
| 7.3.8 | Understand the four-layer TCP/IP (transmission control protocol/ internet protocol) model, including the function and related protocols of each layer:   * application layer * transport layer * internet layer * network layer. |
| 7.3.9 | Understand the role of data packets in transmitting over a network, including:   * contents and structure of a data packet * role of the components of a data packet * packet switching:   + causes of packet loss * error handling:   + cyclic redundancy check (CRC). |

|  |  |
| --- | --- |
| 7.3.10 | Understand the role of common network protocols:   * web protocols:   + HTTP   + HTTPS * mail protocols:   + SMTP   + POP   + IMAP * routing protocols:   + RIP   + OSPF * application protocols:   + FTP   + SFTP   + DHCP   + DNS. |
| 7.3.11 | Understand the concepts of bandwidth and latency, and their effect on the performance of networks and connected systems. |
| **7.4 Virtual environments** | |
| 7.4.1 | Understand the role and characteristics of common virtual environment components:   * virtual machines:   + clients (virtual PC, virtual switch, virtual router)   + servers * hypervisors:   + type 1   + type 2. |
| 7.4.2 | Understand the key features of virtual environments:   * increased security * managed execution * sharing * aggregation * emulation * isolation * portability. |

|  |  |
| --- | --- |
| 7.4.3 | Understand the benefits of the use of virtual environments:   * cost effectiveness for large environments * easy management * resilience * potentially lower carbon footprint * improved disaster recovery options * better testing environments * provision of education and training options. |
| 7.4.4 | Understand the drawbacks of the use of virtual environments:   * extra hardware load * slower execution time * potential for false representation of performance. |
| **7.5 Cloud environments** | |
| 7.5.1 | Understand different types of cloud:   * private * public. |
| 7.5.2 | Understand the benefits of the use of cloud:   * portability * elasticity * fewer storage limitations * cost effectiveness. |
| 7.5.3 | Understand common cloud delivery models, their advantages and disadvantages, and the way in which responsibility and ownership of resources are distributed between the client and the cloud provider:   * Infrastructure as a Service (IaaS):   + client manages application software, system software (middleware and operating system), runtime, data and user accounts   + cloud provider manages virtualisation and hardware (servers, network and storage) * Platform as a Service (PaaS):   + client manages application software, data and user accounts   + cloud provider manages virtualisation, hardware (servers, network and storage) and systems software (middleware and operating system) and runtime * Software as a Service (SaaS):   + client manages user accounts and data   + cloud provider manages virtualisation, hardware (servers, network and storage) systems software (middleware and operating system), runtime and application software. |

|  |  |
| --- | --- |
| **7.6 Resilient digital environments** | |
| 7.6.1 | Understand the benefits of resilient environments and the impact on organisations and clients:   * increased security:   + data security (storage and transfer)   + reduced vulnerabilities * increased reputation:   + protect brand/image   + retain customer confidence * reduction in downtime. |
| 7.6.2 | Understand methods used to improve the resilience of digital environments:   * software updates/upgrades:   + planned updates/upgrades   + patches in response to new vulnerabilities * hardware replacement:   + rolling replacement plans   + secure disposal * data and system redundancy * device hardening:   + removal of unneeded ports, applications, permissions and access * backup systems and recovery procedures:   + onsite   + remote/offsite   + cloud * hot, cold and warm sites * standard operating procedures:   + effective staff training   + induction   + new digital systems   + new or updated policies. |

|  |  |
| --- | --- |
| **Content area 8: Security** | |
| **8.1 Security risks** | |
| 8.1.1 | Know the type of confidential information held by organisations:   * Human Resources:   + salaries and benefits   + staff personal details |

|  |  |
| --- | --- |
|  | * commercially sensitive information:   + client details   + stakeholder details   + intellectual property   + sales numbers   + contracts * access information:   + usernames   + passwords   + multi-factor authentication (MFA) details   + personal identification number (PIN)   + access codes   + passphrases   + biometric data. |
| 8.1.2 | Understand why information must be kept confidential by organisations:   * salaries and benefits:   + prevent competitors from offering higher wages to attract staff   + prevent employees from comparing salaries/demanding comparable pay * staff details:   + protect privacy   + prevent competitors from directly contacting them * intellectual property:   + prevent competitors from copying designs * client details:   + prevent competitors from contacting clients   + protect client privacy * sales numbers * access information:   + prevent unauthorised access. |
| 8.1.3 | Understand the potential impact to an organisation of failing to maintain privacy and confidentiality:   * non-compliance with regulations:   + loss of licence to practise * loss of trust * damage to organisation’s image * financial loss:   + fines   + refunds   + loss of earnings/termination of contracts * legal action * reduced security. |

|  |  |
| --- | --- |
| **8.2 Types of threats and vulnerabilities** | |
| 8.2.1 | Understand potential technical threats and their impacts on organisations and individuals including prevention and mitigation methods:   * botnets * denial of service (DoS)/distributed denial of service (DDoS) * malicious hacking:   + hacktivists/nation states/organised crime/individuals   + password cracking/brute force   + cross-site scripting   + SQL injection   + buffer overflow * malware:   + viruses   + worms   + key loggers   + ransomware   + spyware   + remote access trojans * social engineering:   + phishing   + spear phishing   + smishing   + vishing   + pharming   + watering hole attacks   + USB baiting * domain name server attack/redirection of traffic * insecure application programming interfaces (APIs) * man-in-the-middle attacks * open/unsecured Wi-Fi networks. |
| 8.2.2 | Understand potential technical vulnerabilities to systems and data:   * inadequate security processes:   + weak encryption   + inadequate password policy   + failure to use multi-factor authentication * out-of-date components:   + hardware   + software (lack of support/compatibility with legacy systems, zero-day bugs)   + firmware. |

|  |  |
| --- | --- |
| 8.2.3 | Understand potential human threats, including prevention and mitigation methods, to systems and data:   * human error:   + file properties   + confirmation boxes   + staff training * malicious employee:   + immediate removal from premises   + suspend user accounts immediately * disguised criminal:   + accompany all visitors   + check identification of visitors * poor cyber hygiene:   + locking all unattended machines   + not writing down passwords   + poor password management. |
| 8.2.4 | Understand potential physical vulnerabilities, including prevention and mitigation methods, to systems, data and information, including:   * lack of access control:   + entry control systems * poor access control:   + do not allow tailgating   + use complex access codes   + change codes regularly   + monitor access areas   + audit of staff access to secure areas * nature of location:   + protect against shoulder surfing   + protect against the environment   + protect against vandalism * poor system robustness:   + rugged machines * natural disasters. |
| 8.2.5 | Understand the potential impact to an organisation of threats and vulnerabilities:   * loss/leaking of sensitive data * unauthorised access to digital systems * data corruption * disruption of service * unauthorised access to restricted physical areas. |

|  |  |
| --- | --- |
| **8.3 Threat mitigation** | |
| 8.3.1 | Understand the purposes, processes, benefits and drawbacks of common threat mitigation techniques:   * security settings:   + hardware   + software * anti-malware software:   + function   + actions * intrusion detection * encryption:   + hashing   + symmetric   + asymmetric * user access policies * staff vetting * staff training * software-based access control * device hardening * backups:   + type (full, incremental, differential)   + safe storage * software updates * firmware/driver updates * air gaps * certification of APIs (application programme interfaces) * VPNs (virtual private networks) * multi-factor authentication * password managers * port scanning * penetration testing:   + ethical hacking   + unethical hacking. |

|  |  |
| --- | --- |
| 8.3.2 | Understand the processes and procedures that assure internet security and the reasons why they are used:   * firewall configuration:   + rules for traffic (inbound and outbound)   + traffic type rules   + application rules   + IP address rules * network segregation:   + virtual   + physical   + offline network * network monitoring * port scanning. |
| **8.4 Interrelationship of components required for effective security** | |
| 8.4.1 | Understand how the relationships in the CIA triad interrelate:   * confidentiality:   + ensuring that data is kept private by controlling who has access to the data * integrity:   + ensuring that the data has not been tampered with; this can be done by maintaining confidentiality * availability:   + ensuring that data is available and useful; this can be done by ensuring integrity. |
| 8.4.2 | Understand the elements of the Identification Authentication Authorisation Accountability (IAAA) model, including the techniques used and their benefits and drawbacks:   * identification:   + recognising the individual within a digital system   + knowledge-based identification, including username   + possession-based identification methods   + biometric-based ID methods * authentication:   + verifying the identity claimed during the identification phase   + multi-factor authentication methods   + passwords and passphrases   + biometric authentication * authorisation:   + ensuring that authenticated users can only access resources and perform actions that they are permitted to   + role-based, using the role of the user within the digital system   + access control lists |

|  |  |
| --- | --- |
|  | * accountability:   + ensuring that any actions within a system can be traced back to the responsible user   + audit logs   + user activity. |