Cambridge TECHNICALS

Accredited

OCR LEVEL 3 CAMBRIDGE TECHNICAL

CERTIFICATE/DIPLOMA IN

IT

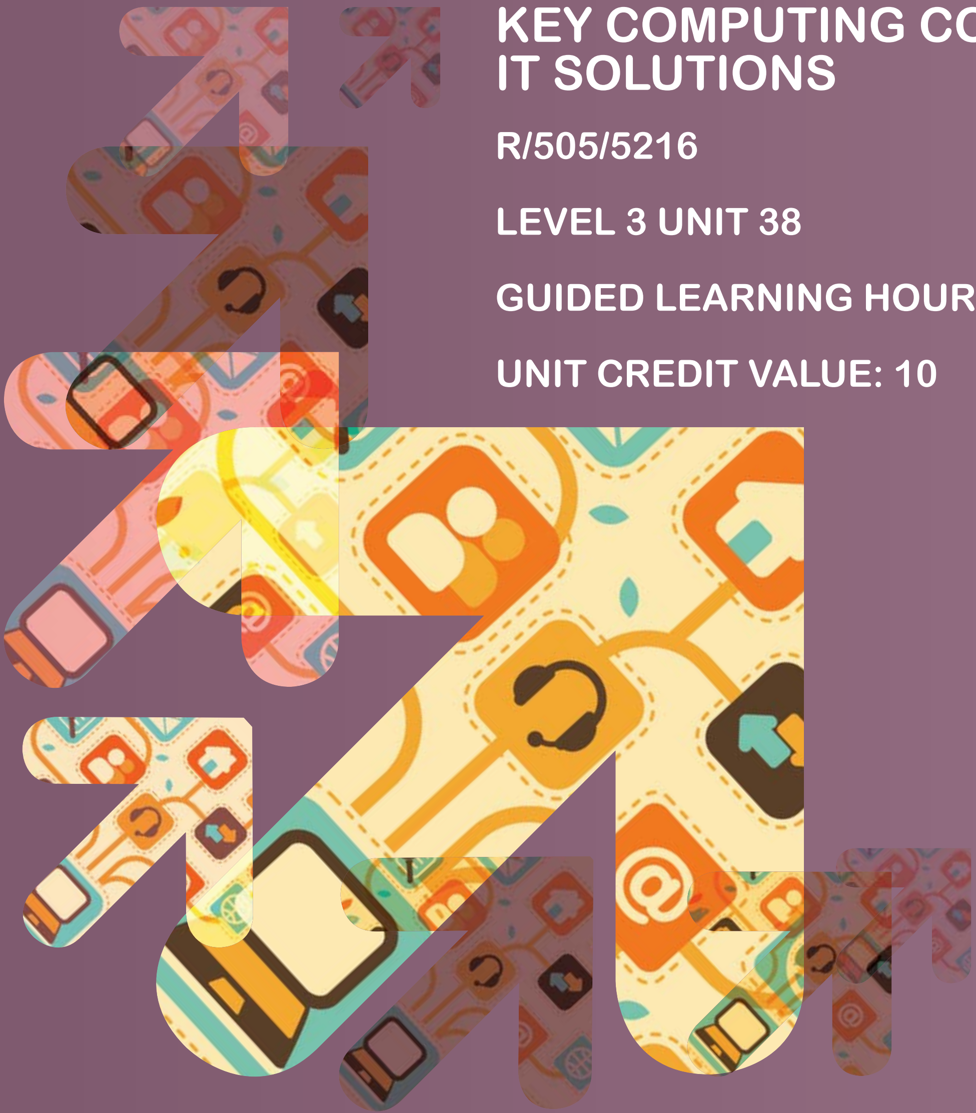
KEY COMPUTING CONCEPTS FOR IT SOLUTIONS

R/505/5216

LEVEL 3 UNIT 38

GUIDED LEARNING HOURS: 60

UNIT CREDIT VALUE: 10



OCR

KEY COMPUTING CONCEPTS FOR IT SOLUTIONS

R/505/5216

LEVEL 3 UNIT 38

AIM AND PURPOSE OF THE UNIT

This unit will enable learners to develop a foundation in the key concepts required for computing and programming in relation to designing IT solutions. They will also learn about developmental processes and the project life cycle for IT solutions.

On completing this unit learners will gain an understanding of how IT and software work together and be able to apply key concepts when designing IT solutions. They will learn about mathematics for computing including number systems, series and matrices. Learners will understand how to apply this knowledge of key concepts together with techniques and processes to establish the building blocks for operating systems, software applications and a range of IT solutions.

ASSESSMENT AND GRADING CRITERIA

Learning Outcome (LO) The learner will:	**Pass** The assessment criteria are the pass requirements for this unit. The learner can:	**Merit** To achieve a merit the evidence must show that, in addition to the pass criteria, the learner is able to:	**Distinction** To achieve a distinction the evidence must show that, in addition to the pass and merit criteria, the learner is able to:
1 Know key concepts of computing and programming used to design IT solutions	P1 describe how different types of data are used in computer systems P2 explain how programming language is used to control computing hardware P3 describe the systems architecture for a specified IT solution	M1 describe how information is converted between computer systems and the analogue world	D1 compare different programming approaches
2 Be able to apply mathematical concepts to computing	P4 perform calculations using different number systems P5 explain the use of number sequences and series in IT solutions P6 use basic Boolean algebra to define outputs		D2 use matrices to support different IT solutions
3 Understand developmental techniques for IT solutions	P7 explain the developmental techniques used when creating IT solutions P8 describe how different platforms are used to meet client requirements across a range of IT applications	M2 design an IT solution to meet a client requirement	
4 Understand the developmental cycle for IT solutions	P9 describe the project life cycle for a software development project P10 explain the use of contingencies in systems implementation	M3 design the stages of systems integration for a bespoke system solution	

TEACHING CONTENT

The unit content describes what has to be taught to ensure that learners are able to access the highest grade.

Anything which follows an i.e. details what must be taught as part of that area of content.

Anything which follows an e.g. is illustrative, it should be noted that where e.g. is used, learners must know and be able to apply relevant examples to their work though these do not need to be the same ones specified in the unit content.

LO1 Know key concepts of computing and programming used to design IT solutions

- **Information types and representation**
  - Use and storage of information as digital data in binary and/or hexadecimal values
  - SI Units and prefixes e.g. nano, micro, milli, centi, kilo, mega, giga, tera, peta
  - Numerical data (e.g. fixed, floating point)
  - Digital media (e.g. images, graphics, audio, video)
  - Structure of data and program files (e.g. with headers, data content, usage rights, security, control instructions)
- **Control of computer hardware e.g. using component hardware address, data, registers, logic, memory maps, interrupts, I/O**
  - Analogue to digital conversion (e.g. for analogue signal inputs that are converted into computer system data)
  - Digital to analogue conversion (e.g. for computer system outputs that are converted into an analogue signal)
- **Programming approaches and languages**
  - Event driven i.e. a programming approach that recognises conditions and changes that trigger responses or programming 'events'
  - Procedural i.e. a programming approach based on a sequence of instructions that includes what is to be done and in what order, as defined by the programmer
  - Object oriented i.e. a programming approach that is based on self contained objects (which contain both program routines and data processing) that are linked together
- **Systems architecture**
  - Component parts e.g. CPU/microprocessor, memory, interface devices, address decoding, graphics controller, communications devices such as USB, network, wireless
  - Structure e.g. system and data buses are used to interconnect devices, address buses decoded to enable specific components, memory maps, synchronised to system clocks, interrupts generated by devices that are monitored and handled by the microprocessor
  - Interconnections e.g. data and address buses to and from system components as well as external devices
  - Communications e.g. USB, network, wireless interfaces
  - Data management and storage e.g. RAM, Flash memory, external storage
  - Inputs, outputs e.g. for data and control signals and interfaces
  - User interface e.g. GUI for host computer system

LO2 Be able to apply mathematical concepts to computing

- **Number systems**
  - Binary
  - Octal
  - Decimal
  - Hexadecimal
- **Conversions between binary, decimal and hexadecimal numbers**
- **Techniques used to perform basic numerical operations in binary and hexadecimal**
  - Add
  - Subtract
  - Multiply
  - Divide
- **Matrices**
  - Use of matrices to store data
  - Indexing and referencing of specific values
  - Matrix operations e.g. add, subtract, multiply, transpose
- **Number sequences and series together with how and where used**
  - Arithmetic
  - Geometric
- **Logic including how and where used e.g. for address decoding**
  - Boolean algebra e.g. NOT/AND/OR/XOR
  - Use of truth tables to determine outputs

LO3 Understand developmental techniques for IT solutions

- **Developmental processes and techniques**
  - Requirements specifications
  - System architecture design
  - Use of algorithms
  - Use of pseudo code
  - Project and team working
- **Systems choices**
  - Platform, operating system and systems software
  - Bespoke/custom solutions e.g. industrial automation, robotics, web services. As used on both computer systems and tablet apps
  - Requirements e.g. communications, data management, data flow, inputs/outputs, protocols, security
- **Platforms**
  - PC servers and workstations
  - Local and cloud based servers
  - Mainframes
  - Programmable Logic Controllers (PLC)
  - Embedded systems and controllers e.g. Raspberry Pi, Arduino
  - Smart phones and tablets
  - Game consoles
- **Operating systems**
  - Windows
  - Linux
  - Unix
  - Android
  - OS X
- **Systems software**
  - Commercial software
  - Open source software
  - Online software
- **Storage devices and peripherals (e.g. fixed, removable)**
  - Mechanical and flash drives
  - NAS (network addressable storage)
  - Multimedia storage devices and drives
  - Cloud storage
- **Security considerations for storage devices**

LO4 Understand the developmental cycle for IT solutions

- **Project life cycle:**
  - Feasibility study
  - Analysis
  - Design
  - Implementation
  - Testing
  - Installation
  - Maintenance
- **Working with development hardware**
  - Off the shelf
  - Bespoke/custom (potential issues with hardware design problems and impact on software debugging)
- **Systems Integration testing (SIT) i.e. the integration of different component parts of a system to produce the final product and how these are tested as a whole (e.g. a bespoke circuit board or electronics with a programming solution)**
- **Prototyping (e.g. ahead of final build or multiple build of data terminals)**
- **Testing and debugging processes**
- **Contingency planning**
- **Impact on development project**

DELIVERY GUIDANCE

This unit is about developing a solid foundation in computing concepts and processes. A large part of the content is theoretical rather than developing practical skills and the delivery approach should reflect this. However, practical exercises, activities and case studies can still be used to demonstrate how the concepts are applied to real world solutions. Learners may have completed other units and established some basic awareness or knowledge that underpins this unit. This may need to be given due consideration to enable a solid foundation in the overall computing concepts to be fully achieved.

Know key concepts of computing and programming used to design IT solutions

To introduce the concepts of computing and programming the tutor could provide presentations and worksheets on the different areas in learning outcome 1. Reference books, papers and websites may be used to assist self-directed learning.

The use of different types of data could, for example, include a comparison of an audio file vs. a graphic file or spread sheet. The hex or ASCII characters that these files are made up from can be viewed rather than opening in a suitable software application. This comparison would identify the structure of the data files with headers and data areas. This knowledge will help learners to identify how to save data in particular file formats when creating programming solutions and also where key information is found when interrogating data files. The process of converting programming instructions and language into executable code for use on a target platform can be covered with hand outs.

The systems architecture and interfaces to external devices can be demonstrated using practical hardware examples. To create an audio file, show how the process is completed with a microphone to capture the analogue sounds which is then converted to a digital format using an ADC (analogue to digital converter). Physical examples of ADC chips and/or 'black box' devices (e.g. with audio ports and a USB output) can be used. Data sheets on devices and chips could also be made available or found using research activities. In general a sound card for a computer system could also be used as an example of both ADC and DAC functions (i.e. microphone input and speaker output ports).

Circuit diagrams for embedded controllers such as Raspberry Pi or Arduino boards or similar will provide opportunities to explore hardware components, especially when the physical boards are also available. The circuit diagrams can be used to annotate specific components and functions such as address bus, data bus, peripheral interfaces etc. Although an understanding of electronics is not required by the unit, an awareness of what devices and components are used and look like will be useful in understanding concepts if IT systems.

Be able to apply mathematical concepts to computing

It should be clear that all forms of data (and programming) are based on binary values that can be expressed in different number systems. Hand outs and sample questions could be used to develop knowledge of mathematics for computing. This should include how programming approaches are used to perform these calculations. Examples of different number systems could be represented in non-computing focussed examples.

Separate workshops should be completed on different parts of this learning outcome. For example, a workshop on each topic of number systems, matrices, number series and logic needs to be included. The workshop on number systems should allow learners to use both manual and calculator based methods for conversions and numerical; operations. Online tools are also available for this purpose and additional research could be completed to locate useful apps where applicable. Handouts that demonstrate how these calculations are performed will form the basis of reference material for future projects, in this and other units.

Although calculators could be used for part of this learning outcome, learners should understand the mechanics of how numerical operations are performed since this is a concept that will be applied to computing and programming in other units. This also applies to Boolean algebra. The practical application of logic is recommended whereby learners decode an address bus that could be used to select a particular component. The use of a memory map (included as part of LO1) will also assist with the structure of a system so that an understanding is gained of where and how this is used.

Understand developmental techniques for IT solutions

Learners will need to understand how IT solutions are designed and the process that is followed. This could be approached using a workshop style activity with practical examples and exercises to design basic systems. Supporting information, hand outs and data sheets on platforms, operating systems, peripherals and security measures will be needed to support this activity.

The tutor could explain in the form of a presentation how, where and why specific processes are important. Examples would be the use of algorithms and pseudo code when designing a system. Learners will need to know how pseudo code is produced in response to a requirements specification, producing examples for themselves. Activities should be included so that learners are able to evaluate the pseudo code of others so identify whether it is meeting the objectives and hence whether suitable for use by a programmer. There are a range of simple and more complex applications that allow learners to develop code visually, using simple techniques, and this will assist them in understanding the structure and process.

Some exercises and activities should be completed as group working so that learners further develop their skills in being members of a project development team. One example here would be the choices to be made of platform, operating and system software when designing solutions. Individual tasks could be allocated since this is likely to build on existing knowledge but up to date knowledge of platforms, software versions, releases and apps is highly recommended.

In general, this learning outcome should be covered using a range of 'hands on' practical activities that is also supported by further research and exploration. The tutor should make available a number of different platforms, components and peripheral devices that can be used to explore, analyse and develop learners' knowledge and understanding of IT and computing concepts.

Understand the developmental cycle for IT solutions

This learning outcome is primarily based around knowledge and understanding and can be supported by a series of hand outs, presentations and work sheets. However, the physical platforms and peripherals used earlier in this unit will provide a practical and applied context to see how they are combined into system solutions. The use of guest speakers or visits to software and hardware developers will greatly support this unit and learning outcome. To see the integration and testing of a bespoke system in a real world situation will be highly beneficial to gain an understanding of the processes and concepts.

The project life cycle can be explained in a presentation and then applied to a specific project using a case study. Different types of software development projects should be covered e.g. PC based systems, embedded controllers such as industrial control or robotics, cloud based applications and bespoke hardware designs. Case studies for these different types should be made available so that comparison can be made with the demands of development, testing and integration. Where case studies are not available, simulated projects could be used within reason but would need to be detailed and realistic. At all stages, learners should be actively encouraged to carry out their own research to expand their understanding as well as their knowledge of the system design process.

Links to other units on project planning and programming could be made so that a more holistic overview of the processes is gained by the learners.

SUGGESTED ASSESSMENT SCENARIOS AND TASK PLUS GUIDANCE ON ASSESSING THE SUGGESTED TASKS

Assessment Criteria P1, P2, P3, M1, D1

For P1 learners must describe how different types of data are used in computer systems. This could be evidenced by a presentation or report on the underlying concepts of computing and programming. This should include the different types and structure of data and program files.

For P2 learners must explain how programming language is used to control computing hardware.   A presentation on how programming language is used to control computing hardware would be a good approach. Examples should be included for different sectors, identifying the hardware and the programming language and the purpose of the programme.

For P3 learners must describe the systems architecture for a specified IT solution. This could be in the form of a presentation, or a system diagram, or a combination of both. It should describe clearly, the systems architecture for an identified IT solution. Learners could create a presentation on system architecture, components and structure.

For merit criterion M1 learners must describe how information is converted between computer systems and the analogue world. Learners must explain the use of digital to analogue (DAC) and analogue to digital converters (ADC). In particular, this should include examples of how analogue values are represented as 8 or 16 bit binary values for instance. Any other resolution of ADC and ADC may be used (e.g. 8/10/12/14/16 bit). This could be evidenced in a presentation or report format.

It is possible that the learner could produce a single report or presentation which could, potentially, meet both the pass and the merit requirements where the systems architecture for an IT solution includes analogue interfaces. Learners must ensure that each assessment criteria is met.

For distinction criterion D1 learners must compare different types of programming approaches. This should include the features of specific languages for different programming approaches and should compare the overall aim and suitability of each type. This could be evidenced in a report or presentation.

Assessment Criteria P4, P5, P6, D2

For P4, learners must perform calculations using different number systems. This should be evidenced across a range of different number systems, including conversions and the calculations performed. The calculations required must be identified to demonstrate the learner has understood and been able to solve the problem.

For P5 learners must explain the use of number sequences and series in IT solutions. Learners could then prepare a presentation or report on the use of number sequences and series with examples of how and where used.

For P6 learners must use basic Boolean algebra to define outputs. Due to the specific nature of examples, this would best be evidenced either practically for an identified purpose, giving examples of how and where it would be used, or as a theoretical document explaining the approaches.

For the distinction criterion D2, learners should use matrices to support different IT solutions.   A practical scenario could be used for this purpose where it is necessary to store data in a matrix which is then used in subsequent operations, and learners must evidence the use of matrices within different IT solutions..

Assessment Criteria P7, P8, M2

For P7 learners must explain the developmental techniques used when creating IT solutions. Learners should evidence the stages, choices and techniques used. This could be evidenced by could prepare a presentation or report containing visual examples or diagrams to support this.

For P8 learners must describe how different platforms are used to meet client requirements across a range of IT applications. This could be in the form of a report for a minimum of three different applications, describing how the client requirements are met for each.

For merit criterion M2 learners must design an IT solution to meet a client requirement. The evidence could take a variety of different formats based on learner's style. The solution produced should cover the categories identified in the teaching content for Learning Outcome 3, and could include pseudo code. The design must be for an entire solution.

Assessment Criteria P9, P10, M3

For P9 learners must describe the project life cycle for a software development project. The evidence for this may largely be theoretical, based on a recognised structure for the software development lifecycle. Learners may choose to include examples at individual stages which will clarify or support their descriptions.

For P10, learners must explain the use of contingencies in systems implementation. This may be an extension to P10 where learners have identified the need for contingencies as part of the project life cycle, but should further explain the requirement for and the application of contingencies in systems. This is best achieved using a range of real examples.

For merit criterion M3, learners must describe the stages of systems integration for a bespoke system solution. This could be an extension to P9, where learners add additional detail and depth to their project life cycles description to identify the stages of system integration for an identified bespoke systems solution.

COMPUTING RESOURCES

Raspberry Pi hardware and peripherals

OCR Raspberry Pi resources

Systems software to produce reports, presentations, pseudo code

MAPPING WITHIN THE QUALIFICATION TO THE OTHER UNITS

Unit 3: Computer systems

Unit 4: Managing networks

Unit 5: Organisational systems security

Unit 7: Computer networks

Unit 8: IT technical support

Unit 10: Developing computer games

Unit 11: Maintaining computer systems

Unit 15: Computer game platforms and technologies

Unit 21: Communication technologies

Unit 22: IT systems troubleshooting and repair

Unit 28: Networked systems security

LINKS TO NOS

4.7 Systems Design

4.8 IT/Technology Infrastructure Design and Planning

5.1 Systems Development

CONTACT US

Staff at the OCR Customer Contact Centre are available to take your call between 8am and 5.30pm, Monday to Friday.
We're always delighted to answer questions and give advice.

Telephone 02476 851509

Email cambridgetechnicals@ocr.org.uk

www.ocr.org.uk