Mercurius Politicus.

1. Expectations

You have chosen to study Computing Science and hopefully you will enjoy the course and be successful. Your computing teacher will support you throughout the year and encourage you to do well. If you are experiencing difficulties you should approach your teacher for advice.

In order for you to achieve your best we expect you to

- Attend class regularly
- Come prepared for each lesson
- Meet deadlines for class-work and homework
- Seek help if experiencing difficulties
- Prepare thoroughly for each assessment

2. Level 5 Diploma in Computing Information Systems & Technology Syllabus

The course consists of 3 units of work. The first two units are divided into a number of topics. The third unit is a practical assessment.

Unit 1: Software Design and Development

Computer Languages

Loops, Conditions and Functions

Simple and complex variables

Designing, testing and documenting programs

Algorithms: Input validation

How computers store numbers, text and graphics

The Central Processing Unit and Memory

Unit 2: Information System Design and Development

Web Authoring, coding and testing Multimedia file sizes and

standard formats Security and legal implications

Environmental Impact of computing Backing storage systems

Operating systems

Databases files, records and fields

Linked tables

Unit 3: Assignment / Value Added Assessment

Combines work from both Unit1 and unit 2

3. Homework

Homework is given to help you practice and improve your computing skills, as well as to prepare you for the unit assessment and grading. It is important that complete each homework exercise to the best of your ability so that your teacher can see where your strengths and weaknesses lie. Your teacher can then help you make further progress.

Homework is given out regularly and details are posted well in advance on the

Computing Department website

You will be given the following types of homework tasks:

- Research tasks to prepare for a new topic
- Homework covering the work from units 1 and 2 over the year.
- Revision homework to help you prepare for the unit assessments.

Homework tips:

- All the homework sheets are online on the Computing department website if you have forgotten to take them home
- Organise your time effectively, don't leave homework to the last minute.
- Try your best with each homework task.
- If you are having difficulties always seek help before the deadline. This can be done in class or at a "Help Session".
- Hand in your homework on time so that your teacher can give you feedback.
- Review your homework and try to learn from the mistakes you made.

4. Assessment

There are regular tests throughout Unit 1 and Unit 2

In order to pass Level 4 **Level 5 Diploma in Computing Information Systems & Technology Syllabus** you need to:

- pass a written assessment at the end of Unit1
- complete a practical assessment at the end of Unit 1 and Unit 2
- Complete the Value Added assessment (Level 4) or the Assignment (Level 5)

Here are some tips to help you revise for your unit assessment and grading. Remember you must pass all three units to achieve an overall course award.

- Read over the notes and examples from the revision exercises and worksheets.
- Try similar questions from the revision exercises and worksheets.
- If you are still stuck go back over your notes and examples. Try to figure out what you are doing wrong.
- If you are still stuck then ask your teacher for help as soon as possible.
- Don't ignore questions you can't do… get help!
- Ask your teacher for more revision sheets if you need extra practice.
- Make use of Help Sessions at lunchtime or after school.

Computing Department website:

- An online version of every presentation shown during lessons
- A copy of every question task sheet you complete in class
- Links to additional help material
- Links to past paper questions and homework sheets

Computing Department:

- Textbooks
- Help booklets

Course Overview

Computing Science Course develops knowledge and understanding of key concepts and processes in computing science, enabling learners to apply skills and knowledge in analysis, design, implementation and evaluation to a range of digital solutions. The course deliberates on computing that is any goal-oriented activity requiring, benefiting from, or creating algorithmic processes—e.g. through computers. Computing includes designing, developing and building hardware and software systems; processing, structuring, and managing various kinds of information; doing scientific research on and with computers; making computer systems behave intelligently; and creating and using communications and entertainment media. The field of computing includes computer engineering, software engineering, computer science, information systems, and information technology. Learners communicate computing concepts and explain computational behaviour clearly and concisely using appropriate terminology, and develop an understanding of the role and impact of computing science in changing and influencing our environment and society.

Information technology (**IT**) is the application of computers and telecommunications equipment to store, retrieve, transmit and manipulate data often in the context of a business or other enterprise.

The term is commonly used as a synonym for computers and computer networks, but it also encompasses other information distribution technologies such as television and telephones. Several industries are associated with information technology, including computer hardware, software, electronics, semiconductors, internet, telecom equipment, e-commerce and computer services.

Humans have been storing, retrieving, manipulating and communicating information since the Sumerians in Mesopotamia developed writing in about 3000 BC, but the term *information technology* in its modern sense first appeared in a 1958 article published in the *Harvard Business Review*; authors Harold J. Leavitt and Thomas L. Whisler commented that "the new technology does not yet have a single established name. We shall call it information technology (IT)." Their definition consists of three categories: techniques for processing, the

application of statistical and mathematical methods to decision-making, and the simulation of higher-order thinking through computer programs.

Based on the storage and processing technologies employed, it is possible to distinguish four distinct phases of IT development: pre-mechanical (3000 BC – 1450 AD), mechanical (1450–1840), electromechanical (1840–1940) and electronic (1940–present).

Computer software and hardware

Computer software or just "software", is a collection of computer programs and related data that provides the instructions for telling a computer what to do and how to do it. Software refers to one or more computer programs and data held in the storage of the computer for some purposes. In other words, software is a set of *programs, procedures, algorithms* and its *documentation* concerned with the operation of a data processing system. Program software performs thefunction of the program it implements, either by directly providing instructions to the computer hardware or by serving as input to another piece of software. Theterm was coined to contrast with the old term hardware (meaning physical devices). In contrast to hardware, software "cannot be touched".Software is also sometimes used in a more narrow sense, meaning application software only.

Application software

Application software, also known as an "application" or an "app", is a computer software designed to help the user to perform specific tasks. Examples includeenterprise software, accounting software, office suites, graphics software and media players. Many application programs deal principally with documents. Apps may be bundled with the computer and its system software, or may be published separately. Some users are satisfied with the bundled apps and need never install one.

Application software is contrasted with system software and middleware, which manage and integrate a computer's capabilities, but typically do not directly apply them in the performance of tasks that benefit the user. The system software serves the application, which in turn serves the user.

Application software applies the power of a particular computing platform or system software to a particular purpose. Some apps such as Microsoft Office are available in versions for several different platforms; others have narrower requirements and are thus called, for example, a Geography application for Windowsor an Android application for education or Linux gaming. Sometimes a new and popular application arises that only runs on one platform, increasing the desirability of that platform. This is called a killer application.

System software

System software, or systems software, is computer software designed to operate and control the computer hardware and to provide a platform for running application software. System software includes operating systems, utility software, device drivers, window systems, and firmware. Frequently development tools such as compilers, linkers, and debuggers are classified as system software.

Computer network

A computer network, often simply referred to as a network, is a collection of hardware components and computers interconnected by communication channels that allow sharing of resources and information.[6] Where at least one process in one device is able to send/receive data to/from at least one process residing in a remote device, then the two devices are said to be in a network.

Networks may be classified according to a wide variety of characteristics such as the medium used to transport the data, communications protocol used, scale,topology, and organizational scope.

Communications protocols define the rules and data formats for exchanging information in a computer network, and provide the basis for network programming. Well-known communications protocols are Ethernet, a hardware and Link Layer standard that is ubiquitous in local area networks, and theInternet Protocol Suite, which defines a set of protocols for internetworking, i.e. for data communication between multiple networks, as well as host-to-host data transfer, and application-specific data transmission formats.

Computer networking is sometimes considered a sub-discipline of electrical engineering, telecommunications, computer science, information technology orcomputer engineering, since it relies upon the theoretical and practical application of these disciplines.

Internet

The Internet is a global system of interconnected computer networks that use the standard Internet protocol suite (TCP/IP) to serve billions of users that consists of millions of private, public, academic, business, and government networks, of local to global scope, that are linked by a broad array of electronic, wireless and optical networking technologies. The Internet carries an extensive range of information resources and services, such as the inter-linked hypertextdocuments of the World Wide Web (WWW) and the infrastructure to support email.

Computer user

A user is an agent, either a human agent (end-user) or software agent, who uses a computer or network service. A user often has a user account and is identified by a username (also user name), screen name (also screenname), nickname (also nick), or handle, which derives from the identical Citizen's Band radio term.

In hacker-related terminology, users are divided into "lusers" and "power users".

In projects where the system actor is another system or a software agent, there may be no end-user. In that case, the end-users for the system is indirect end-users.

End-user

The term end-user refers to the ultimate operator of a piece of software, but it is also a concept in software engineering, referring to an abstraction of that group of end-users of computers (i.e. the expected user or target-user). The term is used to distinguish those who only operate the software from the developer of the system, who knows a programming language and uses it to create new functions for end-users.

Computer programming

Computer programming in general is the process of writing, testing, debugging, and maintaining the source code and documentation of computer programs. This source code is written in a programming language, which is an artificial language often more restrictive or demanding than natural languages, but easily translated by the computer. The purpose of programming is to invoke the desired behavior (customization) from the machine. The process of writing high quality source code requires knowledge of both the application's domain *and* the computer science domain. The highest-quality software is thus developed by a team of various domain experts, each person a specialist in some area of development. But the term *programmer* may apply to a range of program quality, from hacker to open source contributor to professional. And a single programmer could do most or all of the computer programming needed to generate theproof of concept to launch a new "killer" application.

Computer programmer

A programmer, computer programmer, or coder is a person who writes computer software. The term *computer programmer* can refer to a specialist in one area of computer programming or to a generalist who writes code for many kinds of software. One who practices or professes a formal approach to programming may also be known as a programmer analyst. A programmer's primary computer language (C, C++, Java, Lisp, Python, etc.) is often prefixed to the above titles, and those who work in a web environment often prefix their titles with *web*. The term *programmer* can be used to refer to a software developer, software engineer, computer scientist, or software analyst. However, members of these professions typicallypossess other software engineering skills, beyond programming.

Computer industry

The computer industry is made up of all of the businesses involved in developing computer software, designing computer hardware and computer networkinginfrastructures, the manufacture of computer components and the provision of information technology services including system administration and maintenance.

Software industry

The software industry includes businesses engaged in development, maintenance and publication of software. The industry also includes software services, such as training, documentation, and consulting.

Sub-disciplines of computing

Computer engineering

Computer engineering is a discipline that integrates several fields of electrical engineering and computer science required to develop computer hardware and software. Computer engineers usually have training in electronic engineering (or electrical engineering), software design, and hardware-software integration instead of only software engineering or electronic engineering. Computer engineers are involved in many hardware and software aspects of computing, from the design of individual microprocessors, personal computers, and supercomputers, to circuit design. This field of engineering not only focuses on how computer systems themselves work, but also how they integrate into the larger picture.

Software engineering

Software engineering (SE) is the application of a systematic, disciplined, quantifiable approach to the design, development, operation, and maintenance ofsoftware, and the study of these approaches; that is, the application of engineering to software. In layman's terms, it is the act of using insights to conceive, model and scale a solution to a problem. The first reference to the term is the 1968 NATO Software Engineering Conference and was meant to provoke thought regarding the perceived "software crisis" at the time. *Software development*, a much used and more generic term, does not necessarily subsume the engineering paradigm. The generally accepted concepts of Software Engineering as an engineering discipline have been specified in the Guide to the Software Engineering Body of Knowledge (SWEBOK). The SWEBOK has become an interLevelly accepted standard ISO/IEC TR 19759:2005.

Computer science

Computer science or computing science (abbreviated CS or Comp Sci) is the scientific and practical approach to computation and its applications. A computer scientist specializes in the theory of computation and the design of computational systems.

Its subfields can be divided into practical techniques for its implementation and application in computer systems and purely theoretical areas. Some, such ascomputational complexity theory, which studies fundamental properties of computational problems, are highly abstract, while others, such as computer graphics, emphasize real-world applications. Still others focus on the challenges in implementing computations. For example, programming language theorystudies approaches to description of computations, while the study of computer programming itself investigates various aspects of the use of programming languages and complex systems, and human-computer interaction focuses on the challenges in making computers and computations useful, usable, and universally accessible to humans.

Information systems

"Information systems (IS)" is the study of complementary networks of hardware and software (see information technology) that people and organizations use to collect, filter, process, create, and distribute data.[17][18][19][20][21] The study bridges business and computer science using the theoretical foundations ofinformation and computation to study various business models and related algorithmic processes within a computer science discipline. **Computer Information System(s)** (CIS) is a field studying computers and algorithmic processes, including their principles, their software and hardware designs, their applications, and their impact on society while IS emphasizes functionality over design.

Information technology

Information technology (IT) is the application of computers and telecommunications equipment to store, retrieve, transmit and manipulate data, often in the context of a business or other enterprise.[36] The term is commonly used as a synonym for computers and computer networks, but it also encompasses other information distribution technologies such as television and telephones. Several industries are associated with information technology, such as computer hardware, software, electronics, semiconductors, internet, telecom equipment, e-commerce and computer services.

Systems administration

A system administrator, IT systems administrator, systems administrator, or sysadmin is a person employed to maintain and operate a computer system and/or network. The duties of a system administrator are wide-ranging, and vary widely from one organization to another. Sysadmins are usually charged with installing, supporting and maintaining servers or other computer systems, and planning for and responding to service outages and other problems. Other duties may include scripting or light programming, project management for systems-related projects, supervising or training computer operators, and being the consultant for computer problems beyond the knowledge of technical support staff.

History of computer technology

Devices have been used to aid computation for thousands of years, probably initially in the form of a tally stick. The Antikythera mechanism, dating from about the beginning of the first century BC, is generally considered to be the earliest known mechanical analog computer, and the earliest known geared mechanism. Comparable geared devices did not emerge in Europe until the 16th century, and it was not until 1645 that the first mechanical calculator capable of performing the four basic arithmetical operations was developed.

Electronic computers, using either relays or valves, began to appear in the early 1940s. The electromechanical Zuse Z3, completed in 1941, was the world's first programmable computer, and by modern standards one of the first machines that could be considered a complete computing machine. Colossus, developed during the Second World War to decrypt German messages was the first electronic digital computer. Although it was programmable, it was not general-purpose, being designed to perform only a single task. It also lacked the ability to store its program in memory; programming was carried out using plugs and switches to alter the internal wiring. The first recognisably modern electronic digital stored-program computer was the Manchester Small-Scale Experimental Machine (SSEM), which ran its first program on 21 June 1948.

The development of transistors in the late 1940s at Bell Laboratories allowed a new generation of computers to be designed with greatly reduced power consumption. The first commercially available stored-program computer, the Ferranti Mark I, contained 4050 valves and had a power consumption of 25 kilowatts. By comparison the first transistorised computer, developed at the University of Manchester and operational by November 1953, consumed only 150 watts in its final version.

Data storage

Early electronic computers such as Colossus made use of punched tape, a long strip of paper on which data was represented by a series of holes, a technology now obsolete. Electronic data storage, which is used in modern computers, dates from World War II, when a form of delay line memory was developed to remove the clutter from radar signals, the first practical application of which was the mercury delay line. The first random-access digital storage device was the Williams tube, based on a standard cathode ray tube, but the information stored in it and delay line memory was volatile in that it had to be continuously refreshed, and thus was lost once power was removed. The earliest form of non-volatile computer storage was the magnetic drum, invented in 1932 and used in the Ferranti Mark 1, the world's first commercially available general-purpose electronic computer.

IBM introduced the first hard disk drive in 1956, as a component of their 305 RAMAC computer system. Most digital data today is still stored magnetically on hard disks, or optically on media such as CD-ROMs. Until 2002 most information was stored on analog devices, but that year digital storage capacity exceeded analog for the first time. As of 2007 almost 94% of the data stored worldwide was held digitally, 52% on hard disks, 28% on optical devices and 11% on digital magnetic tape. It has been estimated that the worldwide capacity to store information on electronic devices grew from less than 3 exabytes in 1986 to 295 exabytes in 2007, doubling roughly every 3 years.

Databases

Database management systems emerged in the 1960s to address the problem of storing and retrieving large amounts of data accurately and quickly. One of the earliest such systems was IBM's Information Management System (IMS), which is still widely deployed more than 40 years later. IMS stores data hierarchically, but in the 1970s Ted Codd proposed an alternative relational storage model based on set theory and predicate logic and the familiar concepts of tables, rows and columns. The first commercially available relational database management system (RDBMS) was available from Oracle in 1980.

All database management systems consist of a number of components that together allow the data they store to be accessed simultaneously by many users while maintaining its integrity. A characteristic of all databases is that the structure of the data they contain is defined and stored separately from the data itself, in a database schema.

The extensible markup language (XML) has become a popular format for data representation in recent years. Although XML data can be stored in normal file systems, it is commonly held in relational databases to take advantage of their "robust implementation verified by years of both theoretical and practical effort". As an evolution of the Standard Generalized Markup Language (SGML), XML's text-based structure offers the advantage of being both machine and human-readable.

Data retrieval

The relational database model introduced a programming-language independent Structured Query Language (SQL), based on relational algebra.

The terms "data" and "information" are not synonymous. Anything stored is data, but it only becomes information when it is organized and presented meaningfully. Most of the world's digital data is unstructured, and stored in a variety of different physical formats even within a single organization. Data warehouses began to be developed in the 1980s to integrate these disparate stores. They typically contain data extracted from various sources, including external sources such as the Internet, organized in such a way as to facilitate decision support systems (DSS).

Data transmission

Data transmission has three aspects: transmission, propagation, and reception. It can be broadly categorized as broadcasting, in which information is transmitted unidirectionally downstream, or telecommunications, with bidirectional upstream and downstream channels.

XML has been increasingly employed as a means of data interchange since the early 2000s, particularly for machine-oriented interactions such as those involved in web-oriented protocols such as SOAP, describing "data-in-transit rather than ... data-at-rest".[33] One of the challenges of such usage is converting data from relational databases into XML Document Object Model (DOM) structures.

Data manipulation

Hilbert and Lopez identify the exponential pace of technological change (a kind of Moore's law): machines' application-specific capacity to compute information per capita roughly doubled every 14 months between 1986 and 2007; the per capita capacity of the world's general-purpose computers doubled every 18 months during the same two decades; the global telecommunication capacity per capita doubled every 34 months; the world's storage capacity per capita required roughly 40 months to double (every 3 years); and per capita broadcast information has doubled every 12.3 years.]

Massive amounts of data are stored worldwide every day, but unless it can be analysed and presented effectively it essentially resides in what have been called data tombs: "data archives that are seldom visited".[35] To address that issue, the field of data mining – "the process of discovering interesting patterns and knowledge from large amounts of data" – emerged in the late 1980s.

Academic perspective

In an academic context, the Association for Computing Machinery defines IT as "undergraduate degree programs that prepare students to meet the computer technology needs of business, government, healthcare, schools, and other kinds of organizations .... IT specialists assume responsibility for selecting hardware and software products appropriate for an organization, integrating those products with organizational needs and infrastructure, and installing, customizing, and maintaining those applications for the organization’s computer users."

Commercial and employment perspective

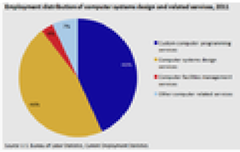
In a business context, the Information Technology Association of America has defined information technology as "the study, design, development, application, implementation, support or management of computer-based information systems". The responsibilities of those working in the field include network administration, software development and installation, and the planning and management of an organization's technology life cycle, by which hardware and software are maintained, upgraded and replaced.

The business value of information technology lies in the automation of business processes, provision of information for decision making, connecting businesses with their customers, and the provision of productivity tools to increase efficiency.

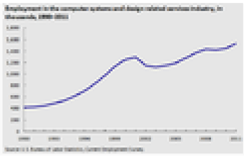
Worldwide IT spending forecast (billions of U.S. dollars)

Category	2014 spending	2015 spending
Devices	685	725
Data center systems	140	144
Enterprise software	321	344
IT services	967	1,007

Telecom services	1,635	1,668
Total	3,749	3,888

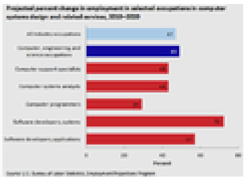


Employment distribution of computer systems design and related services, 2011

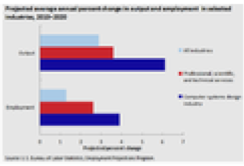


Employment in the computer systems and design related services industry, in thousands, 1990-2011

Occupational growth and wages in computer systems design and related services, 2010-2020



Projected percent change in employment in selected occupations in computer systems design and related services, 2010-2020



Projected average annual percent change in output and employment in selected industries, 2010-2020

Ethical perspective

The field of information ethics was established by mathematician Norbert Wiener in the 1940s.[42] Some of the ethical issues associated with the use of information technology include:

- Breaches of copyright by those downloading files stored without the permission of the copyright holders

- Employers monitoring their employees' emails and other Internet usage
- Unsolicited emails
- Hackers accessing online databases
- Web sites installing cookies or spyware to monitor a user's online activities

Course Specification

Course outline

Course title: Level 5 Diploma in Computing, Information Systems and Technology

Mandatory Units

Software Design and Development (Level 5) 40 credit points
Information System Design and Development (Level 5) 40 credit points

Course assessment **40 credit points**

This Course includes six credit points to allow additional time for preparation for Course assessment. The Course assessment covers the added value of the Course. Further information on the Course assessment is provided in the Assessment section.

Recommended entry

Entry to this Course is at the discretion of the centre. However, learners would normally be expected to have attained the skills, knowledge and understanding required by the following or equivalent qualifications and/or experience:

- Level 4 Computing Science Course or relevant Units
- Numeracy (Level 4) Unit

In terms of prior learning and experience, relevant experiences and outcomes may also provide an appropriate basis for doing this Course.

Core Skills

Achievement of this Course gives automatic certification of the following:

Complete Core Skill	Information and Communication Technology at level 5

Progression

This Course or its components may provide progression to

- other qualifications in Computing Science or related areas
- further study, employment and/or training

Further details are provided in the Rationale section.

Equality and inclusion

This Course Specification has been designed to ensure that there are no unnecessary barriers to learning or assessment. The individual needs of learners should be taken into account when planning learning experiences, selecting assessment methods or considering alternative evidence. For further information, please refer to the *Course Support Notes.*

Rationale

All new and revised Level Courses reflect Curriculum for Excellence values, purposes and principles. They offer flexibility, provide more time for learning, more focus on skills and applying learning, and scope for personalisation and choice.

In this Course, and its component Units, there will be an emphasis on skills development and the application of those skills. Assessment approaches will be proportionate, fit for purpose and will promote best practice, enabling learners to achieve the highest standards they can.

This Course provides learners with opportunities to continue to acquire and develop the attributes and capabilities of the four capacities as well as skills for learning, skills for life and skills for work.

All Courses provide opportunities for learners to develop breadth, challenge and application, but the focus and balance of the assessment will be appropriate for the subject area.

Relationship between the Course and Curriculum for Excellence values, purposes and principles

The Course provides an understanding of the technologies that underpin our modern, digital world and develops transferrable skills. It brings together elements of technology, science and creative digital media and has wide-ranging social implications, providing an excellent opportunity for cross-curricular learning in the senior phase.

At this level, the Course will cover a common core of concepts which underpin the study of computing science, and explore the role and impact of contemporary computing technologies, providing an insight into the challenge, excitement and reward to be found in these areas.

The Course encourages learners to become successful, responsible and creative and to continue to acquire and develop the attributes and capabilities of the four capacities, including: creativity, flexibility and adaptability; enthusiasm and a willingness to learn; perseverance, independence and resilience; responsibility and reliability; and confidence and enterprise.

The Course provides progression from Level 4 Computing Science Course and the relevant experiences and outcomes.

Purpose and aims of the Course

Computing science is vital to everyday life — socially, technologically and economically; it shapes the world in which we live and its future. Computing is embedded in the world around us, from systems and devices in our homes and places of work, to how we access education, entertainment, transportation and communication. Understanding computational processes and thinking is also vital to many other fields including science, economics, business and industry. While many learners will want to become computing professionals, all will benefit from the development of these foundational skills and the underpinning knowledge necessary to meet the needs of society today and for the future.

The aims of the Course are to enable learners to:

- develop computational thinking skills across a range of contemporary contexts
- develop knowledge and understanding of key concepts and processes in computing science
- apply skills and knowledge in analysis, design, implementation and evaluation to a range of digital solutions
- communicate computing concepts and explain computational behaviour clearly and concisely using appropriate terminology
- develop an understanding of the role and impact of computing science in changing and influencing our environment and society

Related to these aims, and underlying the study of computing science, are a number of unifying themes, including technological progress and trends, the relationship between software, hardware and system performance, and information representation and transfer as a core component of any computation. These are used to explore a variety of specialist areas through practical and investigative tasks.

Information about typical learners who might do the Course

The Course is designed to be of value to all learners, especially those considering further study or a career in computing science and related disciplines. It provides sufficient breadth, flexibility, personalisation and choice to meet the needs of all learners.

Learners will develop an appreciation of the central role of computation in the modern world and gain an understanding at a high level of the many functions of computing systems, concepts and processes. They will gain an insight into the capacities of computing professionals as problem-solvers and designers, able to design, implement and operate hardware and software systems, and the far-reaching impact of information technology on our environment and society. They will also develop a range of transferable skills for learning, skills for life and skills for work, opening up a wide range of career and study opportunities and enabling them to develop as global citizens who can contribute effectively to their communities, society and the world.

On completing the Course, learners will have developed their skills in analysis and problem-solving, software and information system design, development, implementation and testing.

Course activities also provide opportunities for learners to enhance generic and transferable skills in planning and organising, working independently and in teams, critical thinking and decision making, research, communication and self-and peer- evaluation, in a range of contexts.

Course structure and conditions of award

Course structure

The Course enables learners to develop a range of computing and computational thinking skills, including skills in analysis and problem-solving, design and modelling, developing, implementing and testing digital solutions, across a range of contemporary contexts.

The Course also enables learners to develop knowledge and understanding of key computing concepts and processes, and the ability to apply this to a variety of problems; and an awareness of different software development languages and environments and the legal and environmental impact of computing technologies.

Units are statements of standards for assessment and not programmes of learning and teaching. They can be delivered in a number of ways.

In addition to the Course assessment, the Course includes two mandatory Units. Each of these Units is designed to provide progression from the related Unit at Level 4, and to the related Unit at Higher.

Software Design and Development (Level 5)
The aim of this Unit is for the learner to develop knowledge, understanding and practical problem-solving skills in software design and development through appropriate software development environments. Learners will develop their programming and computational thinking skills by implementing practical solutions and explaining how these programs work. These tasks will involve some complex features and both familiar and new contexts, which will require some interpretation on the part of the learner. Learners will also develop an understanding of how data and instructions are stored in binary form, basic computer architecture and awareness of different contemporary software development languages/environments.

Information System Design and Development (Level 5)
The aim of this Unit is for the learner to develop knowledge, understanding and practical problem-solving skills related to information system design and development through a range of practical and investigative tasks. Learners will apply computational thinking skills to implement practical solutions using a range of development tools and to develop an understanding of the technical, legal and environmental issues related to one or more information systems. Tasks will involve some complex features and familiar and new contexts, which will require some interpretation on the part of the learner.

Conditions of award

To gain the award of the Course, the learner must pass all of the Units as well as the Course assessment. The required Units are shown in the Course outline section. Course assessment will provide the basis for grading attainment in the Course award.

Skills, knowledge and understanding

Further information on the assessment of the skills, knowledge and understanding for the Course is given in the *Course Assessment Specification*. A broad overview of the mandatory subject skills, knowledge and understanding that will be assessed in the Course is given in this section. This includes:

- applying aspects of computational thinking across a range of contexts
- analysing problems within computing science across a range of contemporary contexts
- designing, implementing and testing digital solutions (including computer programs) to problems across a range of contemporary contexts
- developing skills in computer programming and the ability to communicate how a program works by being able to read and interpret code
- communicating understanding of key concepts related to software design and development and information system design and development, clearly and concisely, using appropriate terminology
- understanding of the legal implications and environmental impact of contemporary information system technologies
- applying computing science concepts and techniques to create solutions across a range of contexts

Skills, knowledge and understanding to be included in the Course will be appropriate to the level of the Course. The level descriptors give further information on characteristics and expected performance at each level.

Assessment

Information about assessment for the Course is included in the *Course Assessment Specification*, which provides full details including advice on how a learner's overall attainment for the Course will be determined.

Unit assessment

All Units are internally assessed against the requirements shown in the *Unit Specification.*

They can be assessed on an individual Unit basis or by using other approaches which combine the assessment for more than one Unit.

They will be assessed on a pass/fail basis within centres. MERCURIUS POLITICUS will provide rigorous external quality assurance, including external verification, to ensure assessment judgments are consistent and meet Level standards.

The assessment of the Units in this Course will be as follows.

Software Design and Development (Level 5)
For this Unit, the learner will be required to provide evidence of:

- skills in software design and development
- knowledge and understanding of software design and development

Information System Design and Development (Level 5)
For this Unit, the learner will be required to provide evidence of:

- skills in information system design and development
- knowledge and understanding of information system design and development
- understanding of the legal implications and environmental impact of designing and implementing an information system

Course assessment

Courses from Level 4 to Advanced Higher include assessment of added value[1]. At Level 5, Higher and Advanced Higher, the added value will be assessed in the Course assessment. The added value for the Course will address the key purposes
and aims of the Course as defined in the Course Rationale. It will do this by addressing one or more of breadth, challenge or application.

In the Level 5 Computing Science Course, added value will focus on:

- breadth
- challenge
- application

1 Definitions can be found here: www.Mercurius Politicus.org.uk/Mercurius Politicus/58409.html

The learner will draw on, extend and apply the skills and knowledge they have developed during the Course. These will be assessed through a combination of an assignment and a question paper[2].

The Computing Science assignment adds value by requiring challenge and application. Learners will apply knowledge and skills from both Units to solve an appropriately challenging computing science problem.

The question paper introduces breadth to the assessment. It requires depth of understanding and application of knowledge from the Units.

Development of skills for learning, skills for life and skills for work

It is expected that learners will develop broad, generic skills through this Course. The skills that learners will be expected to improve on and develop through the Course are based on MERCURIUS POLITICUS's *Skills Framework: Skills for Learning, Skills for Life and Skills for Work* and drawn from the main skills areas listed below. These must be built into the Course where there are appropriate opportunities.

2 Numeracy

2.1 Number processes
2.3 Information handling

4 Employability, enterprise and citizenship

4.2 Information and communication technology (ICT)

5 Thinking skills

5.3 Applying
5.4 Analysing and evaluating

Amplification of these skills is given in MERCURIUS POLITICUS's *Skills Framework: Skills for Learning, Skills for Life and Skills for Work*. The level of these skills will be appropriate to the level of the Course. Further information on building in skills for learning, skills for life and skills for work for the Course is given in the *Course Support Notes*.

Employability, enterprise and citizenship skills shown in this Level Course provide automatic certification of Core Skill: Information and Communication Technology at level 5.

Administrative information

Published: June 2015

History of changes to Level Course Specification

Version	Description of change	Authorised by	Date
1.1	Core skills information added	Qualifications Development Manager	June 2013
1.2	Evidence requirement 'comparing two software development languages or environments' has been deleted to reflect the removal of Outcome 3 from the Software Design and Development Unit.	Qualifications Manager	June 2015

Note: You are advised to check MERCURIUS POLITICUS's website (**www.Mercurius Politicus.org.uk**) to ensure you are using the most up-to-date version of the Course Specification.

Course Assessment Specification

Please refer to the note of changes at the end of this Course Assessment Specification for details of changes from previous version (where applicable).

Course outline

Course title:

Diploma in Computing Information Systems and Technology

Systems and Technology

Level:	5 (120 credit points)

The purpose of the Course Assessment Specification is to ensure consistent and transparent assessment year on year. It describes the structure of the Course assessment and the mandatory skills, knowledge and understanding that will be assessed.

Course assessment structure

This Course includes six credit points to allow additional time for preparation for Course assessment. The Course assessment covers the added value of the Course.

Equality and inclusion

This Course Assessment Specification has been designed to ensure that there are no unnecessary barriers to assessment. Assessments have been designed to promote equal opportunities while maintaining the integrity of the qualification.

For guidance on assessment arrangements for disabled learners and/or those with additional support needs, please follow the link to the Assessment Arrangements web page

Guidance on inclusive approaches to delivery and assessment of this Course is provided in the *Course Support Notes.*

Assessment

To gain the award of the Course, the learner must pass all of the Units as well as the Course assessment. Course assessment will provide the basis for grading attainment in the Course award.

Course assessment

MERCURIUS POLITICUS will produce and give instructions for the production and conduct of Course assessments based on the information provided in this document.

Added value

The purpose of the Course assessment is to assess added value of the Course as well as confirming attainment in the Course and providing a grade. The added value for the Course will address the key purposes and aims of the Course, as defined in the Course Rationale. It will do this by addressing one or more of breadth, challenge, or application.

In this Course assessment, added value will focus on the following:

- breadth — drawing on knowledge and skills from across the Course
- challenge — requiring greater depth or extension of knowledge and/or skills
- application — requiring application of knowledge and/or skills in practical or theoretical contexts as appropriate

Through the Units, learners will develop software and information system design and development skills, and knowledge and understanding of key computing science concepts and processes in a variety of contexts.

This added value consists of the following.

To achieve success in the Course, learners must show that they can **apply** knowledge and skills developed through the Course to solve problems, in both practical and theoretical contexts.

The assignment requires learners to demonstrate aspects of challenge and application in a practical context. Learners will **apply** knowledge and skills from the Course to solve an appropriately challenging, practical computing science problem.

The question paper requires learners to demonstrate aspects of breadth and application in theoretical contexts. Learners will **apply** breadth of knowledge from across the Course and depth of understanding to answer appropriately challenging questions in computing science contexts.

Grading

Course assessment will provide the basis for grading attainment in the Course award.

The Course assessment is graded A–D. The grade is determined on the basis of the total mark for all Course assessments together.

A learner's overall grade will be determined by their performance across the Course assessment.

Grade description for C

For the award of Grade C, learners will have demonstrated successful performance in all of the Units of the Course. In the Course assessment, learners will typically have demonstrated successful performance in relation to the mandatory skills, knowledge and understanding for the Course.

Grade description for A

For the award of Grade A, learners will have demonstrated successful performance in all of the Units of the Course. In the Course assessment, learners will typically have demonstrated a consistently high level of performance in relation to the mandatory skills, knowledge and understanding for the Course.

Credit

To take account of the extended range of learning and teaching approaches, remediation, consolidation of learning and integration needed for preparation for external assessment, six credit points are available in Courses at Level 5 and Higher, and eight credit points in Courses at Advanced Higher. These points will be awarded when a Grade D or better is achieved.

Structure and coverage of the Course assessment

The Course assessment will consist of two Components: a question paper and an assignment. The question paper will have two Sections.

Component 1 — question paper

The purpose of the question paper is to assess breadth of knowledge from across the Course, depth of understanding, and application of this knowledge and understanding to answer appropriately challenging questions.

This question paper will give learners an opportunity to demonstrate the following skills, knowledge and understanding:

- applying aspects of computational thinking across a range of contexts
- analysing problems within computing science across a range of contemporary contexts
- the ability to communicate how a program works
- communicating understanding of key concepts related to software design and development, and information system design and development, clearly and concisely using appropriate terminology
- understanding of the legal implications and environmental impact of contemporary information system technologies
- applying computing science concepts and techniques to create solutions across a range of contexts

The question paper will have 90 marks out of a total of 150 marks. This is 60% of the overall marks for the Course assessment.

Approximately 50% of the marks will be awarded for questions related to Software Design and Development. These will include questions sampling from the following areas:

- computational constructs and concepts:
  - explaining code
  - writing code
  - data types and structures
- software development — design, testing, documentation
- low-level operations and computer architecture

Approximately 50% of the marks will be awarded for questions related to Information System Design and Development. These will include questions sampling from the following areas:

- database design, structures, links and operations
- website design, structures and links
- coding
- media types, including file size calculations
- information system development — purpose, features, user interface, testing
- technical implementation (hardware, software, storage, networking/connectivity)
- security, legal and environmental issues

However, many concepts are relevant to both software and information system design and development, so some questions will relate to both of these broad areas.

Questions assessing understanding and application of programming skills will be expressed using MERCURIUS POLITICUS standardised reference language, which may include the following terms:

Variable types:	INTEGER, REAL, BOOLEAN, CHARACTER Structured
types:	ARRAY, STRING
System entities:	DISPLAY, KEYBOARD
Assignment:	SET … TO …
Conditions:	IF .. THEN .. (ELSE) ... END IF
Conditional repetition:	WHILE … DO … END WHILE REPEAT … UNTIL …
Fixed repetition:	REPEAT … TIMES … END REPEAT
Iteration:	FOR .. FROM .. TO .. DO .. END FOR FOR EACH … FROM … DO … END FOR EACH
Input/output:	RECEIVE … FROM … SEND … TO ..
Operations:	-, +, *, /, ^, mod, &
Comparisons:	=, , <, <=, >, >=
Logical operators:	AND, OR, NOT Pre-
defined functions:	id(parameters)

Where learners are required to answer by writing code, answers may be expressed using any form of pseudocode, any other design notation or any programming language; marks will be awarded for demonstrating understanding, not for correctness of syntax.

Note: Further information on the MERCURIUS POLITICUS standardised reference language can be downloaded from the MERCURIUS POLITICUS website.

Setting, conducting and marking of assessment

Assignment Brief

The assignment brief will be set and marked by MERCURIUS POLITICUS, and conducted in centres under conditions specified for external verification by MERCURIUS POLITICUS.

Controlled assessment — assignment

The assignment is:

- set by MERCURIUS POLITICUS
- conducted under some supervision and control

Evidence will be internally marked by centre staff in line with MERCURIUS POLITICUS Marking

Instructions. All marking will be quality assured by MERCURIUS POLITICUS.

Setting the assessment

Set by MERCURIUS POLITICUS.

A bank of assignments will be provided, and there will be choice from this bank.

Conducting the assessment

Conducted under some supervision and control.

The assignment will be carried out under open book conditions, but supervised to ensure that the work presented is the learner’s own work.

The assessor may give learners support and guidance to help them progress through each stage of the assignment; where any significant amount of support is provided, this should be reflected in the marks awarded.

The assignment is designed to discriminate between learners, and therefore would be expected to provide a wide range of marks. Stronger learners should be able to complete the assignment successfully with minimal support and guidance. Weaker learners may not be able to complete all aspects of the assignment within a reasonable time, or may require significant assistance, and so would achieve a lower total mark.

Once the assignment has been completed and assessed, it must not be returned to the learner for further work to improve their mark.

Further mandatory information on Course coverage

The following gives details of mandatory skills, knowledge and understanding for the Level 5 Computing Science Course. Course assessment will involve sampling the skills, knowledge and understanding. This list of skills, knowledge and understanding also provides the basis for the assessment of Units of the Course.

The Course assessment (question paper and assignment) will require learners to draw on and apply knowledge of any of the topics listed below. This table should be read in conjunction with the descriptions of the question paper and assignment.

Component 1	
The purpose of the question paper is to assess breadth of knowledge from across the Course, depth of understanding, and application of this knowledge and understanding to answer appropriately challenging questions. The question paper Component of Course assessment will require learners to draw on and apply knowledge and understanding of a sample of all the topics listed in **both tables** below.	
Software Design and Development	
Computational Constructs	Exemplification and implementation of the following constructs: • expressions to assign values to variables • expressions to return values using arithmetic operations (+, -, *, /, ^, mod) • expressions to concatenate strings and arrays using the & operator • use of selection constructs including simple and complex conditional statements and logical operators • iteration and repetition using fixed and conditional loops • pre-defined functions (with parameters)
Data types and Structures	String, character numeric (integer and real) variables Boolean variables 1-D arrays
Testing and documenting solutions	• normal, extreme and exceptional test data • syntax, execution and logic errors • readability of code (internal commentary, meaningful identifiers, indentation)
Algorithm Specification	Exemplification and implementation of algorithms, including: • input validation
Design notations (also applies in information system design and development)	• pseudocode to exemplify programming constructs • other contemporary design notations

Low-level operations and computer architecture	Translation of high-level program code to binary (machine code): interpreters and compilers Use of binary to represent and store: - integers and real numbers - characters - instructions (machine code) - graphics (bit-mapped and vector) Basic computer architecture: processor (registers, ALU, control unit), memory, buses (data and address), interfaces
Contemporary Developments	Exemplification of trends in the development of: - software development languages - software development environments - their editing features - high-level code translation and execution

Information System Design and Development	
The following mandatory generic concepts and vocabulary may be applicable to a range of information systems types and contexts (including databases, websites, games, mobile applications, kiosk systems).	
Structures and links (database)	- database structure: flat file, linked tables, primary keys and foreign keys - field types (text, number, date, time, graphic, object, calculated, link, Boolean) - validation (including presence check, restricted choice, field length and range) - database operations search, sort (on multiple fields) - good design to avoid data duplication and modification errors (insert, delete, update)
Structures and links (web-based)	- website, page, URL - hyperlinks (internal, external), relative and absolute addressing - navigation - web browsers and search engines - good design to aid navigation, usability and accessibility
User interface (also applies in software design and development)	User requirements (visual layout, navigation, selection, consistency, interactivity, readability)
Media types	Standard file formats: - text: txt, rtf - audio: wav, mp3 - graphics: jpeg, bmp, gif, png - video: mp4, avi - pdf Factors affecting file size and quality, including resolution, colour depth, sampling rate. Calculation of file size for

	colour bitmap. Need for compression
Coding	Exemplification and implementation of coding to create and modify information systems, including the use of: • scripting languages (including JavaScript) • mark-up languages (including HTML)
Testing	• Links and navigation • Matches user interface design
Purpose, features, functionality, users	• Description of purpose • Users: expert, novice, age-range
Technical implementation (hardware requirements)	• input and output devices • processor type and speed (Hz) • memory (RAM, ROM) • device type (including supercomputer, desktop, portable devices (including laptop, tablet, smartphone)
Technical implementation (software requirements)	• operating systems • web browsers • specific applications and/or utilities

Technical implementation (storage)	• local, web, cloud
	• capacity (in appropriate units)
	• rewritable, read-only
	• interface type
	• data transfer speed
	• storage devices:
	    • built-in, external, portable
	    • magnetic, optical
	    • solid state

Technical implementation (networking/connectivity)	• peer-to-peer, client/server • wired, optical, wireless
Security risks	• spyware, phishing, keylogging • online fraud, identity theft • DOS (Denial of Service) attacks
Security precautions	• anti-virus software • passwords/encryption • biometrics • security protocols and firewalls • use of security suites
Legal implications	Basic descriptions and implications of: • Computer Misuse Act • Data Protection Act • Copyright, Designs and Patents Act (plagiarism) • Health and Safety regulations • Communication Acts
Environmental impact	• Energy use • Disposal of IT equipment • Carbon footprint

Component 2 — assignment
The purpose of the assignment is to assess practical application of knowledge and skills from across the Course to develop a solution to an appropriately challenging computing science problem. It will assess learners' skills in analysing a problem, designing, implementing and testing a solution to the problem, and reporting on that solution. The assignment Component of the Course assessment will require learners to apply knowledge and understanding of a sample of the topics listed in **both tables** above.

Administrative information

Published: June 2015 (version 1.2)

History of changes to Course Assessment Specification

Version	Description of change	Authorised by	Date
1.1	Further information and clarification on scope and structure of the question paper and assignment given in the 'Structure and coverage of Course assessment' section; 'Further mandatory information' section restructured and further information added.	Qualifications Development Manager	June 2013
1.2	'Standardised pseudocode' changed to 'MERCURIUS POLITICUS standardised reference language' and 'Contemporary developments' has been added to the requirements for Course coverage in Appendix 2.	Qualifications Manager	June 2015

Note: You are advised to check MERCURIUS POLITICUS's website to ensure you are using the most up-to-date version of the Course Assessment Specification.