MCAST

MQF/EQF Level 6

IT6-A2-21

Bachelor of Science (Honours) in Software Development

Course Specification

Course Description

The degree in Software Development is intended to prepare learners to work in the industry of application development and engineering and explores the skills required for designing IT enterprise solutions, building robust backend systems, and solving complex problems using the latest paradigms and modern technologies. The programme covers aspects related to software engineering, frontend and backend software technologies, data organisation, persistence and data analysis, software development of portable devices and software quality aspects such as security and testing. Modern trends such as Cloud, Devops, Blockchain, Machine Learning and Computer Vision are also explored. The degree includes work-placements that will allow students to gain experience within the industry and work in a professional environment. At the end of the degree, the students culminate their studies with a research dissertation in an area of their interest. The course is intended for individuals who are keen in solving problems through technology and would like to pursue an exciting career in one of the fundamental pillars of the economy.

Programme Learning Outcomes

At the end of the programme the learner will be able to:

1. *Design, implement and document the underlying data infrastructure to support software applications;*
2. *Design, implement and document the back-end of enterprise applications for a given requirement;*
3. *Revise a software design/implementation to optimise its use of resources;*
4. *Test and secure the software application and its content to conform to industry standards.*

Entry Requirements

MCAST Advanced Diploma in IT
Recommended stream: Software Development
OR
MCAST Advanced Diploma in Electronics (Computer Engineering)
OR
2 A-Level passes and 2 I-Level passes
Compulsory A-Level: Computing
AND
Compulsory A-Level or I-Level: Mathematics (Pure or Applied) or Physics

Key Information

Awarding Body – MCAST

Accreditation Status – Accredited via MCAST's Self Accreditation Process (MCAST holds Self-Accrediting Status as per 1st schedule of Legal Notice 296/2012)

Type of Programme: Qualification

MQF Level	Examples of Qualifications	'Qualification' Minimum Credits Required	'Award' Credits Required
Level 8	Doctoral Degree Third Cycle Bologna Process	NA	NA
Level 7	Masters Second Cycle Bologna Process	90-120	Less than 30
	Post-Graduate Diploma	60	
	Post-Graduate Certificate	30	
Level 6	Bachelor[23]/Bachelor (Hons.)[24] First Cycle Bologna Process	180-240	Less than 180
Level 5	Short Cycle Qualification	120	Less than 60
	Undergraduate Higher Diploma	90	
	Undergraduate Diploma	60	
	Undergraduate Certificate	30	
	VET Level 5 Programme[25]	60-120	
Level 4	Pre-Tertiary Certificate	30	Less than 120
	VET Level 4 Programme[26]	120	
	MATSEC Certificate	NA	
Level 3	VET Level 3 Programme[27]	60	Less than 60
	General and Subject Certificate	NA	
Level 2	VET Level 2 Programme[28]	60	Less than 60
	General and Subject Certificate	NA	
Level 1	VET Level 1 Programme[29]	40	Less than 40
	General and Subject Certificate	NA	
Introductory Level A	Preparatory Programme	30	Less than 30
Introductory Level B	Pre-entry Basic Skills Course	30	Less than 30

Table 1: Minimum number of credits for 'Qualifications' and parameters for 'Awards'

Fig.1: p56, Ministry for Education and Employment & National Commission for Further and Higher Education Malta (2016). *Referencing Report, 4th Edition*. NCFHE.

Total number of Hours: 4500 hours

Mode of attendance: Fully Face-to-Face Learning

Duration: 3 Years

Target audience for MCAST full-time courses is 16 to 65+

The official language of instruction at MCAST is English. All notes and textbooks are in English (except for language courses, which will be in the respective language being instructed). International candidates will be requested to meet English language certification requirements for access to the course.

This course will be offered at

MCAST has four campuses as follows:

MCAST Main Campus
Triq Kordin, Paola, Malta

All courses except for the Institute for the Creative Arts, Centre of Agriculture, Aquatics and Animal Sciences are offered here.

Institute for the Creative Arts
Mosta Campus
Misraħ Għonoq Tarġa Gap,
Mosta

Institute of Applied Sciences,
Centre of Agriculture, Aquatics and Animal Sciences,
Luqa Road, Qormi

Gozo Campus
J.F. De Chambray Street
MCAST, Għajnsielem
Gozo

Teaching, Learning and Assessment

The programmes offered are vocational in nature and entail both theoretical lectures delivered in classes as well as practical elements that are delivered in laboratories, workshops, salons, simulators as the module requirements dictate.

Each module or unit entails a number of in person and/or online contact learning hours that are delivered by the lecturer or tutor directly (See also section 'Total Learning Hours).

Access to all resources is provided to all registered students. These include study resources in paper or electronic format through the Library and Resource Centre as well as tools, software, equipment and machinery that are provided by the respective institutes depending on the requirements of the course or module.

Students may however be required to provide consumable material for use during practical sessions and projects unless these are explicitly provided by the College.

All Units of study are assessed throughout the academic year through continuous assessment using a variety of assessment tools. Coursework tasks are exclusively based on the Learning Outcomes and Grading Criteria as prescribed in the course specification. The Learning Outcomes and Grading Criteria are communicated to the Student via the coursework documentation.

The method of assessment shall reflect the Level, credit points (ECTS) and the schedule of time-tabled/non-timetabled hours of learning of each study unit. A variety of assessment instruments, not solely Time Constrained Assignments/Exams, are used to gather and interpret evidence of Student competence toward pre-established grading criteria that are aligned to the learning outcomes of each unit of the programme of study.

Grading criteria are assessed through a number of tasks, each task being assigned a number of marks. The number of grading criteria is included in the respective Programme Specification.

The distribution of marks and assessment mode depends on the nature and objectives of the unit in question.

Coursework shall normally be completed during the semester in which the Unit is delivered.

Time-constrained assignments may be held between 8 am and 8 pm during the delivery period of a Unit, or at the end of the semester in which the Unit is completed. The dates are notified and published on the Institute notice boards or through other means of communication.

Certain circumstances (such as but not limited to the Covid 19 pandemic) may lead Institutes and Centres to hold teaching and assessment remotely (online) as per MCAST QA Policy and Standard for Online Teaching, Learning and Assessment (Doc 020) available via link https://www.mcast.edu.mt/college-documents/

The Programme Regulations referenced below apply. (DOC 005 available at: link https://www.mcast.edu.mt/college-documents/)

Total Learning Hours

The total learning hours required for each unit or module are determined as follows:

Credits (ECTS)	Indicative contact hours	Total Student workload (hrs)	Self-Learning and Assessment Hours
1	5 – 10 hrs	25 hrs	20-15 hrs*
2	10 – 20 hrs	50 hrs	40-30 hrs*
3	15 – 30 hrs	75 hrs	60-45 hrs*
4	20 – 40 hrs	100 hrs	80-60 hrs*
6	30 – 60 hrs	150 Hrs	120-90 hrs*
9	45 – 90 hrs	225 hrs	180-135 hrs*
12	60 – 120 hrs	300 hrs	240-180 hrs*

* The 'Self-Learning and Assessment Hours' amount to the difference between the contact hours and total student workload.

Grading system

All MCAST programmes adopt a learner centred approach through the focus on Learning Outcomes. The assessment of MCAST programmes is criterion-referenced and thus assessors are required to assess learners' evidence against a pre-determined set of Learning Outcomes and assessment criteria.

For a student to be deemed to have successfully passed a unit, a minimum of 50% (grade D) must be achieved. In case of part time programmes, the student must achieve a minimum of 45% to successfully pass the unit.

All units are individually graded as follows:
A* (90-100)
A (80-89)
B (70-79)
C (60-69)
D (50-59)
Unsatisfactory work is graded as 'U'.

Work-based learning units are graded on a Pass/Fail basis only.

Detailed information regarding the grading system may be found in the following document: DOC 005 available at: link https://www.mcast.edu.mt/college-documents/

Intake Dates

•MCAST opens calls for application once a year between July and August of each year for prospective applicants residing in MALTA.
•Applications to full-time courses from international students not residing in MALTA are accepted between April and Mid-August.
•For exact dates re calls for applications please follow this link https://www.mcast.edu.mt/online-applications-2/

Course Fees

MCAST course are free for Maltese and EU candidates. International candidates coming from outside the EU need to pay fees for the respective course. Course fees are set on a per-level and course duration basis. For access to course fee structure and payment methods please visit https://www.mcast.edu.mt/fee-payments-for-non-eu-candidates/.

Method of Application

Applications to full-time courses are received online via the College Management Information System. Candidates can log in using Maltese Electronic ID (eID) or European eIDAS (electronic identification and trust services) to access the system directly and create an account as the identity is verified electronically via these secure services.

Non-EU candidates need to request account creation though an online form by providing proof of identification and basic data. Once the identity is verified and the account is created the candidate may proceed with the online application according to the same instructions applicable to all other candidates.

Non-EU candidates require a study visa in order to travel to Malta and joint the course applied for. For further information re study-visa please access https://www.identitymalta.com/unit/central-visa-unit/.

For access to instructions on how to apply online please visit https://www.mcast.edu.mt/online-applications-2/

Contact details for requesting further information about future learning opportunities:

MCAST Career Guidance
Tel: 2398 7135/6
Email: career.guidance@mcast.edu.mt

Current Approved Programme Structure

Unit Code	Unit Title	ECTS	Year	Semester
ITSFT-506-2006	Object Oriented Programming	6	1	A
ITSFT-506-1606	Software Engineering	6	1	A
ITMTH-506-1601	Discrete Mathematics	6	1	A
ITDBS-506-1603	Database Programming I	6	1	A
CDKSK-503-1907	English I	3	1	A
ITSFT-506-2009	Mobile Applications Development	6	1	B
ITSFT-506-1608	Data Structures and Algorithms	6	1	B
ITSFT-506-1609	Low Level Programming	6	1	B
ITDBS-506-2003	Database Programming II	6	1	B
ITSFT-506-2007	Software Test Automation	6	1	B
CDKSK-503-1908	English II	3	1	B
ITSFT-506-2010	Interactive Mobile Development	6	2	A
ITSFT-506-2003	Client Side Scripting I	6	2	A
ITSFT-506-1612	Server Side Scripting	6	2	A
ITSYS-506-2005	Systems Programming	6	2	A
ITSFT-506-2011	Enterprise Programming	6	2	A
ITSFT-506-2012	Securing Applications	6	2	B
ITBSI-506-1601	Business Intelligence and Reporting	6	2	B
ITRSH-506-2101	Research Design 1	6	2	B
CDKSK-604-1909	Entrepreneurship	4	2	B
CDKSK-602-2105	Community Social Responsibility	2	2	B
CDWBL-506-1901	Work Based Learning I	6	2	B
ITIMG-606-1601	Image Processing and Computer Vision	6	3	A
ITSFT-606-1618	Applied Computational Intelligence	6	3	A
ITSFT-606-1619	Data Structures and Algorithms II	6	3	A
ITSTA-606-1601	Statistics for Computer Science	6	3	A
ITSFT-606-2101	Distributed Programming	6	3	B
ITSFT-606-1620	Programming for the Cloud	6	3	B
ITRSH-606-2102	Research Design 2	6	3	B
CDWBL-506-1902	Work Based Learning II	6	3	B
ITDIS-612-1601	Dissertation	12	3	B
Total ECTS		**180**	**/**	**/**

ITSFT-506-2006: Object Oriented Programming

Unit Level (MQF/EQF): 5

Credits: 6

Delivery Mode: Fully Face-to-Face Learning

Total Learning Hours: 150

Unit Description

This unit aims to build up learners from basic concepts in OOP to more advanced object-oriented concepts. Basic OOP concepts include objects and classes, access modifiers, properties, constructors and use of methods (parameters and returning data). Following this, more advanced concepts will be introduced such as encapsulation, polymorphism and abstraction. Focus will be given in particular to possible relationships between different objects. Such relationships include inheritance, association, aggregation and composition. Such concepts will be applied to more advanced scenarios, ensuring that there is an in depth understanding of the different concepts.

Finally, the unit covers persistence. In this part of the unit, learners will integrate with and store data in a database using the object-relational language LINQ. Learners will appreciate the differences between a data-centric approach and an object-centric approach when designing the model layer of an application.

During the learning process for this unit, learners will be given the necessary tools to acquire skills for proper management of building an Object-Oriented solution. Such management will be including preparation and design, time management, presentation, handling queries and as error handling.

Learning Outcomes

On completion of this unit the learner will be able to:

1. *Build object-oriented solutions using both fundamental and advanced object-oriented concepts to be able to address business requirements.*
2. *Implement different relationships between objects found in a scenario and show proper understanding of such existing relationships.*
3. *Implement persistence in created applications to allow created applications to store and read data from multi-user database management systems.*
4. *Show management skills in the process of building and deploying an Object-Oriented solution.*

ITSFT-506-1606: Software Engineering

Unit Level (MQF/EQF): 5

Credits: 6

Delivery Mode: Fully Face-to-Face Learning

Total Learning Hours: 150

Unit Description

This unit has been designed to introduce learners to the main concepts behind the science of software engineering. Throughout the course of their studies, students will acquire the skills to understand and support the complete life cycle of a software system – from inception, requirements elicitation and design, through the various stages until release and maintenance. Students will gain an understanding of different software development techniques and will learn how to critically select which technique is best suited to the development of different systems.

The unit places focus on some of the more recent software development processes, making particular emphasis on the Agile philosophy of software development. Students will understand the agile process and its constituent components, its applicability to modern software development and the various actors involved in the process together with their roles and responsibilities. Another core component of this module will be that of introducing students to the Unified Modelling Language, UML, as a tool to facilitate and speed up the software development process. The various constructs of this modelling language will be covered, together with explanations of how they can be utilised to specify and document the software and business processes.

This unit will also present students with a range of advanced software engineering concepts and approaches which will give them the skills required to be able to support new and evolving developments. Students will be introduced to a number of different software architectures and design approaches and they will be encouraged to analyse which setups are most adequate as solutions for diverse scenarios

Learning Outcomes

On completion of this unit the learner will be able to:

1. *Plan and tackle a small software design project as part of a team using an Agile approach.*
2. *Perform a requirements acquisition exercise in order to identify the main functional and non-functional requirements of a proposed software system.*
3. *Identify and construct the most applicable UML modelling diagrams to use in particular phases in a software system's development process to achieve a specified goal.*
4. *Design a solution to a problem by proposing the most suitable architecture and utilising known design patterns.*

ITMTH-506-1601: Discrete Mathematics

Unit Level (MQF/EQF): 5

Credits: 6

Delivery Mode: Fully Face-to-Face Learning

Total Learning Hours: 150

Unit Description

This unit is designed to introduce students to the mathematical foundations of computer science. Writing a computer program goes beyond knowledge of the programming language utilised. Most non-trivial software programs involve the use of complex data structures and mathematical algorithms to function. All of these cannot be created without an understanding of the underlying mathematical concepts. In this unit, focus is placed on encouraging students to think logically and mathematically and on giving them the tools necessary to apply these concepts in their future work as computer scientists and software developers.

Learners will be introduced to logic as the language with which reasoning can be explained. Both propositional and predicate logic will be covered, with students learning how to interpret logical statements and, in turn, how to utilise formal proof systems to validate given propositions. Discrete structures are presented through coverage of set theory, relations and the classification of different types of relations. The numbering system we utilize on a day to day basis will also be placed under scrutiny with learners understanding the foundations behind enumeration and the standard numeric operators applied on natural numbers. Proof techniques such as contraposition, induction, reduction and contradiction will be introduced throughout the course of this unit giving students the necessary tools to handle the problems set.

Learning Outcomes

On completion of this unit the learner will be able to:

1. *Use formal reasoning and notation to explain the correctness of mathematical arguments.*
2. *Discuss the basic concepts of set and graph theory, relations and functions and reason mathematically about these discrete structures.*
3. *Demonstrate an understanding of number theory and an ability to produce rigorous proofs centred on the subject*
4. *Apply proofs using induction techniques to solve problems set.*

ITDBS-506-1603: Database Programming I

Unit Level (MQF/EQF): 5

Credits: 6

Delivery Mode: Fully Face-to-Face Learning

Total Learning Hours: 150

Unit Description

This unit provides the basis of advanced database theory and design principles that shall be used in future modules. Following this module, a student should be confident in concepts about relational theory and database design. The theory presented is independent of any specific database management system, however the database design conforms to agreed-upon notation easily adopted to other relational database management systems.

The unit starts with the data manipulation via data manipulation language. In this part of the unit, the learner will learn how to build databases in a relational database management system by creating tables and choosing indices. Attention is given to the enforcement of data integrity rules and their associated delete and update repercussions via foreign keys.
Following this, data query via data query language is discussed. Basic select statements are covered including selecting from multiple tables and using conditions (such as WHERE). This is then extended to include predicates and combining predicates as well as filtering character, date and time data. Finally, data is ordered and paged using appropriate techniques.

Advanced concepts then follow, including performing different types of joins based on the database content as well as using set operations. As part of this topic, different types of joins will also be discussed. At this point, data insertion, updates and deletion will be discussed using the appropriate SQL keywords.
Grouping (via GROUP BY) and windowing are also discussed in this topic, allowing for both single and multiple grouping as well as pivoting. For windowing, aggregation, ranking and offsets will be discussed. The unit concludes with a discussion and practical use of views, as well as how inline functions can be incorporated into SQL to extract calculated fields.

Learning Outcomes

On completion of this unit the learner will be able to:

1. *Create the structure of a relational database.*
2. *Retrieve data from a database.*
3. *Manipulate the data in a database.*
4. *Prepare advanced reports from data within a database.*

ITSFT-506-2009: Mobile Applications Development

Unit Level (MQF/EQF): 5

Credits: 6

Delivery Mode: Fully Face-to-Face Learning

Total Learning Hours: 150

Unit Description

Mobile Phones have evolved significantly in technology and application within the short period of their existence. From the original purpose of providing voice calls for travelling office executives these devices have now evolved into handheld, multi-media, smart devices boasting an array of sensors, multi-core processing power and a market penetration across young and old, personal and professional use. A significant factor behind the success of smart phone devices is the ability for the user to personalise with well designed, productive mobile applications. This unit is designed to introduce the learner to key aspects in mobile application development from Operating System and development environment through to application state and data storage.
This is a skills based unit and will allow learners to demonstrate they have the necessary skills to be able to design, program and test a mobile application. The unit will guide learners through the basics of Operating Systems (OS), development environments, device sensors and data storage. Learners will use software programming concepts and as a consequence should able to operate effectively at more than a basic level of competence before commencing this unit.

Outcome 1 introduces the mobile Operating System of choice as well as the development tools available. The learner will familiarise themselves with the development environment and test device/simulator.

Outcome 2 concentrates on implementing the application (Graphical) User Interface (UI). Learners will explore the fundamentals behind building a GUI for the particular OS. They will investigate the various libraries and classes as well as designing and using OS menus.

Outcome 3 focuses on working with differing user states. Learners will understand how to handle application start up, background and resuming as well as considering state changes associated with screen rotation or notifications.

Outcome 4 looks at various options for storing/loading context data internally or externally (through the use of memory cards) and cloud based services.

Learning Outcomes

On completion of this unit the learner will be able to:

1. *Examine the fundamental components of mobile application.*
2. *Develop a practical mobile application user interface.*
3. *Manipulate a mobile application depending upon state change.*
4. *Use the appropriate methods to store and load data in a mobile application.*

ITSFT-506-1608: Data Structures and Algorithms

Unit Level (MQF/EQF): 5

Credits: 6

Delivery Mode: Fully Face-to-Face Learning

Total Learning Hours: 150

Unit Description

The basis of solving a problem requires an understanding of how to break it down into a series of much more manageable small parts. In order to do this, students need to be able to assess the complexity of a problem. Once the algorithm has been broken down into smaller sections, a student should start to write logical instructions using pseudocode. Each instruction in turn will manipulate data, which may be for instance array structures, which are similar to vectors in mathematics or abstract data types such as pointers, these are similar to how machine code uses memory addresses to access data.

In this unit students will learn about writing algorithms for common problems such as Queues and thereby choosing the most appropriate data structure.
Students need to implement a series of algorithms which are well known in Computer Science. For a given algorithm a student will need to analyse the complexity and make a decision on how this may affect the efficiency of the algorithm in terms of run time. Although computers now have very powerful processors, students still need to estimate the time it would take for their algorithm to process a given amount of data. In particular as the amount of data becomes larger the amount of time it takes to process the data can grow exponentially.

Students will learn and appreciate that algorithms can be translated into programming code. This in turn will give them an insight into solving problems on paper before typing their code into a text editor for a given programming language. This experience will allow them to see how their programs run as originally indented in the specification. Also it has been written in such a way that it runs efficiently, avoiding complexity in their solution as well as making best use of the processing power of their computer system.

Data structures such as pointers, which allow a programmer to use memory addresses to access data, give the student a much more flexible method to manipulate data. For each algorithm a student needs to select the most appropriate data structure, in order to produce a solution which will carry out the required tasks as set out in the specification for a computer program

Learning Outcomes

On completion of this unit the learner will be able to:

1. *Construct programs using Abstract Data Types and Structured Data types.*
2. *Design efficient algorithms for commonly encountered problems using existing examples.*
3. *Make use of algorithm analysis to determine the efficiency of an algorithm.*
4. *Compare algorithms in terms of their correctness, proof and intractability.*

ITSFT-506-1609: Low Level Programming

Unit Level (MQF/EQF): 5

Credits: 6

Delivery Mode: Fully Face-to-Face Learning

Total Learning Hours: 150

Unit Description

A comprehensive understanding of Low Level Programming allows engineers to implement efficient and resilient software within computer based solutions as well as providing a strong grounding in the various microprocessor/microcontroller architectures available. The 'C' low level programming language is a widely used and versatile general purpose computer programming language found in many engineering applications such as Embedded Computing. It has many derivatives within areas such as GUI development, gaming and control.

This is a skills based unit and will allow learners to demonstrate they have the necessary skills to be able to design, program and test 'C' based low level programs for various applications. The unit will guide learners through the principles of constructing programs using libraries, header files and pre-processor directives, working with data types and data management as well as dealing with memory management, interrupts, Networks and Files. This unit is a continuation from Programming Concepts 1 (Level 4) and it is expected that Learners will be familiar with this unit's content.

Outcome 1 concentrates on the typical requirements for embedded low level programs. The learner will familiarise themselves with the process of compiling and linking 'C' programs including libraries as well as using header files and pre-processor directives. The learner will also use an Integrated Development Environment tool set to test and debug code.

Outcome 2 focuses on dealing with the application structure and code logic of a 'C' program. Learners will use a range of data types and manipulate them using common 'C' commands including bitwise logic and arithmetic operations as well as string manipulation.

Outcome 3 emphasises flow control, arrays and pointers. Learners will build an understanding of various flow control techniques including, if-else, switch-case, while and for loops as well as investigating types of arrays and pointer, their construction and manipulation.

Outcome 4 highlights the implementation of functions or subroutines. Learners will also demonstrate their mastery of the Unit by developing a 'C' language program implementing all of the features they have studied in order to solve a specified engineering proble

Learning Outcomes

On completion of this unit the learner will be able to:

1. *Construct a viable Low Level Language development environment ('C').*
2. *Demonstrate the ability to manipulate various data types using arithmetic means.*
3. *Produce 'C' programs with the ability to change flow and search lists.*
4. *Construct 'C' programs using subroutine structures (functions).*

ITDBS-506-2003: Database Programming II

Unit Level (MQF/EQF): 5

Credits: 6

Delivery Mode: Fully Face-to-Face Learning

Total Learning Hours: 150

Unit Description

This unit is a continuation of Database Programming I and furthers the student's knowledge in advance database programming concepts. The unit is based on four main concepts commonly needed in large scale database systems: advanced database objects such as triggers, stored procedures, views and sequences; query optimization; error handling and concurrency support; integration of advanced techniques such as JSON support and integration with external applications. This unit empowers the student with the necessary information to design a scalable database and integrate it with external applications, which is a very common and realistic industry need.
It is important to note that this unit is a continuation of Database Programming I and so the latter is a pre-requisite. This unit delves into advanced techniques commonly faced by backend software developers and systems architects.

Business Intelligence topics have not been included in this unit since these are considered as more advanced topics which merit a unit of their own. It is also strongly recommended that such a unit would follow this unit, since this unit offers a solid basis for the advanced BI topics.

Learning Outcomes

On completion of this unit the learner will be able to:

1. *Improve a database design using advanced techniques needed for support of big systems.*
2. *Optimise database and query performance.*
3. *Design a resilient database system via error handling and concurrency support.*
4. *Integrate support for external data and applications with an existing database.*

ITSFT-506-2007: Software Test Automation

Unit Level (MQF/EQF): 5

Credits: 6

Delivery Mode: Fully Face-to-Face Learning

Total Learning Hours: 150

Unit Description

Like for other products, it is essential that software works correctly and carries out the specific tasks which were promised by the developers. In order for software to meet quality standards, it is important to thoroughly test programs using well known techniques in Computer Science. While manual testing will always have a role, the industry is moving towards automating significant parts of the software testing.

Test automation is the practice of using tools to repeatedly run tests to ensure the tested software meets the requirements. Such automation brings several advantages including the repeatability of the tests and the speed at which tests can be executed. Continuous Integration and Continuous Delivery (CI/CD) tools are essential for testing to become an integral part of the development cycle.

Test Driven Development (TDD) is an approach, where a test written before writing the code which will make sure that the source code is tested well and the resulting code is modular. The resultant code is flexible to changes and easier to extend and adapt to future requirements. The stages involved in TDD include: adding a test, running all tests and checking if the new code fails, writing code, running automated tests, re-factoring code and finally repeating the cycle as in previous steps with a new test.

The initial testing process would start by resolving syntax errors and producing a runnable program. Secondly, test data would need to be used check whether the program has any logic errors and confirm that the program meets the specification. The computer program needs to ensure the integrity of the data is maintained by means of a data verification and validation processes.

This module is geared towards enabling students to familiarise with the current methods of automated software testing, and learn about the different levels at which testing can be automated including Unit Testing, Integration Testing and System Testing.
The ideas of Alpha and Beta in acceptance testing are also covered. Alpha testing is the initial testing carried out internally to identify issues that were not found originally

through previous tests. Beta testing involves the distribution of pre-release software to a select group of users so that they can test it in a realistic environment.

Modern software development processes are highly and responsive and agile, with software being delivered in much shorter cycles than the past. This makes achieving the required level of software quality more challenging and only achievable using automated tests. Software developers need to choose the right test automation tools and integrating them in their processes, typically a CI/CD.

Learning Outcomes

On completion of this unit the learner will be able to:

1. *Compose tests using Test Driven Development methodology.*
2. *Explain the process involved in the different levels of automated testing.*
3. *Analyse the Fundamentals of Testing.*
4. *Use the automated tests in relation to developing a computer system.*
5. *Produce tests to Verify and Validate software.*

ITSFT-506-2010: Interactive Mobile Development

Unit Level (MQF/EQF): 5

Credits: 6

Delivery Mode: Fully Face-to-Face Learning

Total Learning Hours: 150

Unit Description

Mobile Phones have evolved significantly in technology and application within the short period of their existence. From the original purpose of providing voice calls on the move these devices have now evolved into handheld, multi-media, smart devices boasting multi-core processing power, a wide range of sensors, and a market penetration across young and old, personal and professional use. A significant factor behind the success of smart phone devices is the ability for developers to innovate upon the device hardware and Operating System in order to provide beneficial and lucrative mobile applications (Apps).

This unit is a continuation to Mobile Applications Development (Level 6) and it is expected that learners will be familiar with the unit's content. The topic cover within this module make a mobile application more interactive with a better and engaging UI/UX. This is a skill based unit and will allow learners to demonstrate they have the necessary skills to be able to design, program, consume web services, use OS's services and content provider and make use of the on-board hardware.

Learning Outcomes

On completion of this unit the learner will be able to:

1. *Creating mobile application with appealing UI.*
2. *Use Background Services and Other Features.*
3. *Develop a mobile application to utilize Web API's.*
4. *Use on-board hardware.*

ITSFT-506-2003: Client Side Scripting I

Unit Level (MQF/EQF): 5

Credits: 6

Delivery Mode: Fully Face-to-Face Learning

Total Learning Hours: 150

Unit Description

This unit is designed to introduce learners the fundamental and intermediate concepts of client-side scripting. In particular emphasis will be placed on manipulating the DOM, creating a responsive web application and using a JavaScript framework/library that allows the developer to build highly dynamic interactive web applications. It is intended for learners who already have some understanding of HTML, CSS and JavaScript.

The unit starts with a brief introduction of why developers felt the need to create frameworks and a revision of client-side scripting, whereby the learner is shown how HTML, CSS and JavaScript can be used together to build the presentation layer of a web application. This includes basic tasks covered in pure JavaScript and being able to react to user input and update the screen dynamically without the need to refresh the page. Learners are introduced to the Single-Page-Application (SPA) approach where the Server sends an HTML page and thereafter the JavaScript framework/library takes over and controls the UI. Following this, learners will be introduced to the Document Object Model (Model) which encompasses not only the structure of a document but also the behavior of a document and the objects of which it is composed. Then learner will be introduced to CSS and JavaScript frameworks/libraries, which simplify the development of a responsive single-page application. Learners will learn how to bind to DOM elements via a JavaScript framework/library and hence how to manipulate DOM objects. They will also learn how to listen for changes in the DOM and react accordingly with JavaScript functions. Learners will also be introduced to asynchronous JavaScript whereby pages load faster since the browser isn't waiting for the server to respond in the middle of a page render since requests and transfers happen in the background. Using these libraries and JavaScript, the learners are then shown how to create and consume data in a JavaScript Object Notation (JSON) format.

Learning Outcomes

On completion of this unit the learner will be able to:

1. *Apply basic requirements to build the presentation layer of a web application using HTML, CSS and JavaScript.*
2. *Develop a web application with state management in place, providing users with authentication.*
3. *Implement using a JavaScript Framework/Library a Web Application that fetches and consumes data.*
4. *Deploy a web application with proper navigation which can handle and validate user*
5. *Input data through forms.*

ITSFT-506-1612: Server Side Scripting

Unit Level (MQF/EQF): 5

Credits: 6

Delivery Mode: Fully Face-to-Face Learning

Total Learning Hours: 150

Unit Description

Server side scripting is a powerful and customizable technology for creating dynamic web pages. It has several benefits over client side scripting. For example it allows controllable environment for executing server side scripts whereas user environment cannot be controlled when using client side scripting. Server side scripts are able to change the HTML output according to the web browsers. Server- side technologies are installed on the web servers to process the scripts and HTML stream is then returned to the client's web browser. This unit does not require the use of any particular server side scripting language and the user can choose depending upon their knowledge and skills.
This is a skills based unit and will allow learners to demonstrate that they have the necessary skills to be able to design, program and test server side scripts. The unit will guide learners through the process of constructing dynamic web pages using relevant server side technology.

Outcome 1 concentrates on the typical requirements for running server side scripts and understanding the difference between static and dynamic web pages. The learner will familiarise themselves with the requirements and installation process. The learner will also understand the process of configuring and setting up the web server to test web pages.

Outcome 2 focuses on dealing with the anatomy of a dynamic web page using server side scripting and how this technology works. Learners will understand different data types supported by their chosen programming language through the use of Server side controls/web controls while creating simple web pages.

Outcome 3 focuses on the use of control structures in the chosen programming language. Learners will understand the use of operators, branching and looping structures. They will also understand the practical aspects of the event driven programming. They will also understand the use of functions by passing parameters and returning values. They will learn how to identify problems when loading the web page and testing its functionality.

Outcome 4 emphasises on theoretical and practical aspects related to the use of databases in dynamic web pages. They will understand the use of data handling controls to fetch data for web pages. Learners will also be given the opportunity to demonstrate their mastery of the unit by developing a program using their chosen programming language implementing all of the features they have studied in order to design a dynamic data handling web application

Learning Outcomes

On completion of this unit the learner will be able to:

1. *Construct a viable environment to create dynamic web pages with reference to client-server architecture.*
2. *Develop asynchronous dynamic web pages using server side scripting language.*
3. *Demonstrate the ability to implement OOP event driven programming and handling errors.*
4. *Demonstrate the ability to integrate server side scripting with databases to manage and manipulate data.*

ITSFT-506-2012: Securing Applications

Unit Level (MQF/EQF): 5

Credits: 6

Delivery Mode: Fully Face-to-Face Learning

Total Learning Hours: 150

Unit Description

Understanding the attacker's perspective is key to have securely implemented applications. Thus, in this unit the main objective will be that of exposing and determining what are the top and most popular vulnerabilities, providing also examples and what damages these could do if exploited by an attacker. Secondly, the unit will focus on building a secure application following well established standards and mitigations, it's not about development but also about design. Security should be a design requirement for a project even though there are time and cost constraints to the project. In order to develop secure applications, important security considerations should be adapted throughout the software development life cycle. Lastly but as much as important, time will also be dedicated for testing and making use of widely available tools to ensure that practices that have been adopted are enforcing the security measures applied.
Most of the application attacks are based on common vulnerabilities. OWASP which stands for Open Web Application Security Project publishes a list of most ten most popular programming mistakes whilst developing an application. These vulnerabilities are common to hackers and are exploited frequently. OWASP top 10 will be used as a guide throughout this unit. Many of these threats can be mitigated by applying some well-known security standards, and throughout this unit, the learner will be provided with practical examples on how these vulnerabilities can be mitigated. Thus the unit will be focused on a more hands-on approach

Learning Outcomes

On completion of this unit the learner will be able to:

1. *Analyze systems for vulnerabilities and security flaws.*
2. *Implement practical countermeasures for the detected threats.*
3. *Apply a variety of cryptographic algorithms to encrypt data.*
4. *Research and make use of tools to help in highlighting the importance of proper (penetration) testing.*

ITSYS-506-2005: Systems Programming

Unit Level (MQF/EQF): 5

Credits: 6

Delivery Mode: Fully Face-to-Face Learning

Total Learning Hours: 150

Unit Description

Smart devices embedded with powerful microprocessors running sophisticated low level programs have now become part of the mainstream. These devices are finding their way into more and more sophisticated but non-intrusive use-cases such as machine-to-machine Internet of Things. As such, the modern software engineer requires a broad yet in depth understanding of the requirements for Low Level programming of these systems, associated hardware and networking.

This skills based unit provides learners with a platform to learn and demonstrate the skills required to engineer involved, low level programs for a range of relevant use case scenarios. The unit will develop learner's abilities within structure and compilation of programs, the effective design and use of system memory, provision of real-time functionality with the use of interrupts, culminating with practical investigations into networking. This unit builds on what was covered in Low Level Programming 1 (Level 5) and it is expected that Learners will be familiar with this unit's content.

Outcome 1 concentrates on the typical requirements for embedded low level programs. The learner will familiarise themselves with the process of compiling and linking programs using multiple source files, cover common build problems and the use of 'Make' or 'Build' processes. The learner will also use typical compiler directives.

Outcome 2 focuses on dealing with the application structure of a low level program as well as memory management implementations. Compile and run time memory management will be investigated using memory allocation commands, complex pointer operations and data structures (e.g. linked lists).

Outcome 3 emphasises application features such as concurrency, multi-tasking and File Input/output.

Outcome 4 investigates the implementation of network programming using sockets. Learners will also be given the opportunity to develop a program implementing all the features they have studied in order to solve a specified engineering problem.

Learning Outcomes

On completion of this unit the learner will be able to:

1. *Construct a multiple-source program for a specific hardware platform.*
2. *Develop a memory managed low level application.*
3. *Demonstrate the ability to implement a multi-tasking operating application.*
4. *Demonstrate the ability to implement a networked application using Sockets.*

ITSFT-506-2011: Enterprise Programming

Unit Level (MQF/EQF): 5

Credits: 6

Delivery Mode: Fully Face-to-Face Learning

Total Learning Hours: 150

Unit Description

Enterprise applications are software solutions that provide business logic and tools to satisfy the needs of an organisation to improve productivity and efficiency. These type of applications are commonly the central hub for a large variety of data received from a number of sources.

This unit will provide Learners the theoretical and practical knowledge to design and develop enterprise applications. This will provide Learners the knowledge to create well defined business logic that can meet the clients' requirements.

Learners will learn various enterprise software architecture types including different enterprise software design patterns. Learners will also learn the role which software architecture plays in the development of larger scale enterprise systems. Moreover, Learners will gain an understanding of the common enterprise software features offered by most enterprises to adhere to common entities or policies set up by clients. Furthermore, Learners will gain the necessary knowledge to identify common frameworks that follow common standards of design and development practices for developing enterprise software applications.

Through this unit, Learners will gain an insight in designing and developing business logic that will enable them to understand how to meet clients' requirements. This will help them grasp a better understanding of what is involved in the actual configuration management and scalability needs for such large scale applications. Moreover, this unit will enable the Learners to understand basic security requirements to safeguard and share enterprise knowledge amongst various applications

Learning Outcomes

On completion of this unit the learner will be able to:

1. *Identify the importance of software architecture and its role within enterprise applications.*
2. *Examine the different software patterns used to design enterprise software.*
3. *Discuss the capabilities, configuration and management processes involved in designing and delivering enterprise software.*
4. *Outline the use of cloud services and the method of uploading content.*

ITBSI-506-1601: Business Intelligence and Reporting

Unit Level (MQF/EQF): 5

Credits: 6

Delivery Mode: Fully Face-to-Face Learning

Total Learning Hours: 150

Unit Description

This unit is designed to provide students with the skills necessary to understand and participate in data warehousing projects and to support the analytical reporting tasks that would successively be carried out on these data stores. Technological advances made in recent years have led to an explosion in the amount of data being collected and stored within organisations. Most businesses now recognize the fact that they can harness the power of big data to increase their competitiveness and/or improve their processes, and this is not limited only to commercial scenarios. The ability to analyse voluminous data sets has been an important step forward for the most disparate of fields spanning government institutions, medicine and health, astronomy and biology and many more. During the course of this unit, students will be introduced to Business Intelligence as a collection of tools and techniques that allow for the extraction of knowledge from large sources of data.

Due to its central role in any BI solution, focus will be placed on the data source itself with the first part of the unit dedicated to the data warehouse. The architecture and logical and physical design of the data warehouse will be covered both theoretically as well as through practical exercises. Students will learn to identify data sources and analyse and transform data sets in an ETL procedure to populate the data warehouse.

OLAP analysis will be covered as a second main topic of this unit. The concept of multidimensional data structures and the various operations that can be carried out on them to analyse data from different viewpoints and at varying level of detail will be highlighted in order to give students an understanding of the benefits brought about by BI technology.

Learning Outcomes

On completion of this unit the learner will be able to:

1. *Discuss the ways in which data warehousing coupled with Business Intelligence technologies help to meet the data requirements of strategic decision makers within organisations.*
2. *Design the schema for a data warehouse to meet a given set of requirements and implement the said model as a relational data warehouse.*
3. *Outline and discuss the main steps involved in an ETL process and support the theoretical knowledge with the design and implementation of such a process to populate a Data Warehouse.*
4. *Explain multi-dimensional data structures and discuss and demonstrate the capabilities of OLAP tools through the practical application of dedicated OLAP analysis software on a data warehouse store.*

ITRSH-506-2101: Research Design 1

Unit Level (MQF/EQF): 5

Credits: 6

Delivery Mode: Fully Face-to-Face Learning

Total Learning Hours: 150

Unit Description

The purpose of this unit is to give the learner the necessary skills to start researching in an area of personal interest yet also of relevance to the area of studies and to the benefit of the local/regional community. This module differentiates itself from the rest in the manner that the criteria focus on how research is to be performed, whilst it is the learner who will determine the subject area and tools to be utilized to build the prototype necessary.

Therefore, this unit requires the learner to identity a theme such as the study of beach deterioration in local beaches, then through guidance, determine the sources of data (Satellite imagery) and tools needed to conduct such research. Every other learner will focus on areas that are either of personal interest, subject areas communicated by other researchers (potential future mentors), topics of funded projects, or recommendations by various parties such as key external partners. This unit will give a structure to how research is conducted in a scientific manner, following industry standards and common practice.

The fundamental objective of this unit is to introduce the learner to hypothesis testing. Therefore, after a theme is selected, a hypothesis needs to be formulated together with research questions. A research pipeline highlighting the methods to be used in order to address the research questions follows. A preliminary literature review will be undertaken by the researcher in order to familiarize themselves with the current state of the art and to justify key decisions made in their individual research.

The learner is expected to work on a proof-of-concept, prototype or working solution in order to gather the necessary data from research experiments so to be able to argue and answer the set research questions. Upon analysis and reflection, the learner should be in a position to understand what a scientific research process is, what is expected of him/her from a dissertation and how to undertake such a research endeavour at a larger scale as expected in the final dissertation.

Learning Outcomes

On completion of this unit the learner will be able to:

1. *Formulate a research hypothesis and research methodology.*
2. *Evaluate, after research, the current state of the art.*
3. *Evaluate project outcomes critically.*
4. *Report project outcomes and recommendations within a structured framework.*

ITSFT-606-2101: Distributed Programming

Unit Level (MQF/EQF): 6

Credits: 6

Delivery Mode: Fully Face-to-Face Learning

Total Learning Hours: 150

Unit Description

Distributed programming focuses on developing programs that run within distributed systems. Distributed systems consist of systems containing application components located on networked computers that communicate with each other through passing messages from one component to another. Application components interact with one another to achieve a common objective. A Service Oriented Architecture (SOA) is an architectural style that supports service orientation, whereby services, a discrete unit of functionality, provide application components that can be accessed remotely through computer networks. Service Oriented Architecture is a common programming style for designing and developing distributed programs. Service Oriented Architecture has evolved into other architecture paradigms such as Microservices Architecture, where services are broken down into several lightweight, independent, loosely-coupled services, and Event-driven Architecture, where communication and messaging amongst services are executed in the form of events.

The purpose of this unit is to provide the learners an understanding on Service Oriented Architecture (SOA), Event-driven Architecture and Microservices Architecture for developing distributed applications. The unit will focus on how a set of components can be set to be invoked remotely using these different architectures. Moreover, the unit will provide the learners the principles, benefits and differences of developing using these architectures. This unit will also give a hands-on approach and will guide learners to create Application Programming Interfaces (APIs) which will enable communication between different distributed applications by utilising technologies such as REST, SOAP, Web Services, RPCs and Events.

Apart from setting up services on a Server, this Unit will also outline how services modelled using these architectures can be consumed from the client by making use of different data serialisations such as JSON and XML. Learners will be taught the differences and similarities between JSON and XML together with the technical detail to be able to construct and parse both JSON and XML data.

In this Unit, learners will also be taught how to makes use of widely used APIs to get information such as weather data, social networking data and transactional

information. Learners will also be exposed to the concept of cloud services and how cloud services can be used when designing distributed applications.

Learning Outcomes

On completion of this unit the learner will be able to:

1. *Describe the main concepts behind a Service Oriented Architecture by utilising SOAP, Web Services and Remote Procedure Calls (RPCs).*
2. *Implement a Microservices Architecture by utilising REST.*
3. *Demonstrate an Event-driven Architecture by utilising Events.*
4. *Use and consume third party APIs and Cloud Services.*

ITIMG-606-1601: Image Processing and Computer Vision

Unit Level (MQF/EQF): 6

Credits: 6

Delivery Mode: Fully Face-to-Face Learning

Total Learning Hours: 150

Unit Description

This unit is designed to give the students a basic yet solid understanding of Image Processing techniques, as well as the application of such techniques in the rapidly-developing area of Computer Vision. The latter has been known to be a difficult problem, due to the complexities involved in the human vision system. Nonetheless, it is finding its way through many sectors such as safety and security, health, and entertainment. Moreover, methods of acquiring such data (digital images and videos) have become even more available and affordable.

The unit first introduces the theory behind Image Processing and Computer Vision, and moves on to the application of fundamental operations in Image Processing, such as quantisation and removal of noise. These fundamental operations are then used in more complex problems, such as finding edges, corners and other interest points or shapes in the image.

Finally, some typical problems of Computer Vision will be examined. These include object detection, recognition and tracking. These problems will serve as challenges as well as motivation to students, as they highlight the relevance and practicality of the material covered. As much as possible the unit will adopt a learn-by-doing approach, in that for most of the topics the underlying theory will be explained followed by a practical example or implementation. This will ensure comprehension and engagement of the student.

Learning Outcomes

On completion of this unit the learner will be able to:

1. *Understand the basic concepts and relevance of Image Processing and Computer Vision.*
2. *Apply fundamental techniques in Image Processing.*
3. *Apply techniques to identify shapes and features in images.*
4. *Demonstrate Object Detection, Recognition and Tracking Techniques.*

ITSFT-606-1618: Applied Computational Intelligence

Unit Level (MQF/EQF): 6

Credits: 6

Delivery Mode: Fully Face-to-Face Learning

Total Learning Hours: 150

Unit Description

This unit builds on the previous data related modules, namely Database Programming I, Database Programming II, and Business Intelligence & Reporting. The purpose of this unit is to focus on the data analysis aspect of artificial intelligence.
The aim of this unit is to provide the learner with the data analytical skills needed to identify patterns from data reports and to solve real world problems commonly encountered by professionals. This unit will highlight the different kind of solutions and approaches that can be taken in specific scenarios.

The first aspect of this module will be to expose the learner to NP-complete kind of problems and explain when to use exact solution algorithms and when to use heuristic algorithms. Then the student should be able to do exploratory analysis on a dataset to determine an appropriate approach to address the problem at hand.
After an exploratory analysis, the student should be able to clean the data and normalise it in preparation for use within an algorithm. Such algorithm will vary from exact algorithms to heuristic algorithms. Such algorithms will be used to proof or disproof a hypothesis.

Finally, a proper analysis of the gathered data will be done in an academic report following a specific conference template. This report will follow a similar structure as that used for the dissertation which will serve as further practice and preparation for their final year project.
For this unit, a statistical analysis software such as Microsoft Excel and R-Programming will be used. Development of a small prototype can be done using any other programming language. Documentation of the final report will be done using a conference template such as IEEE, ideally with a LaTeX editor.

Learning Outcomes

On completion of this unit the learner will be able to:

1. *Understand different solution algorithm types and when to use each.*
2. *Analyse a dataset and identify an appropriate approach to solve the problem.*
3. *Implement several solution algorithms for a given data set and problem.*
4. *Document findings in a technical report.*

ITSFT-606-1619: Data Structures and Algorithms II

Unit Level (MQF/EQF): 6

Credits: 6

Delivery Mode: Fully Face-to-Face Learning

Total Learning Hours: 150

Unit Description

This unit covers data structures and algorithms and builds upon previous data structures and algorithms units. An understanding of basic data structures, O and Θ notation is expected.

The unit covers several topics including: analysis of array growth in array based vectors, graphs and spanning trees and their algorithms, weighted graphs, strings and string searching algorithms, hash tables and other tree algorithms.

The core aim of this unit is to give the learners additional algorithm analysis skills and tools that they can use in their work. In addition to this, the unit aims to introduce the students to a variety of commonly used data structures along with algorithms based on these structures. These can be applied to solve particular problems at the work place or as part of research, such as the research being undertaken in the student's thesis.
At the end of the unit, the student is expected to be able to carry out analysis of new data structures and algorithms, as well as being able to apply new data structures and algorithms to problem domains that the student is working in. The student should be able to independently analyse similar problems and select the ideal structure and/or algorithm to solve a particular problem as well as demonstrate their skills through application.

The lecturer can choose any programming language. A programming language familiar to the students is recommended. Pseudo-code and mathematical notation can also be used where applicable.

Learning Outcomes

On completion of this unit the learner will be able to:

1. *Understand and implement a variety of data structures and algorithms.*
2. *Apply data structures and algorithms to correctly and efficiently solve real-world problems.*
3. *Analyse and compare different data structures and algorithms.*
4. *Synthesise solutions to real-world problems through the application of one or more data structures and algorithms or variants, correctly and efficiently.*

ITSTA-606-1601: Statistics for Computer Science

Unit Level (MQF/EQF): 6

Credits: 6

Delivery Mode: Fully Face-to-Face Learning

Total Learning Hours: 150

Unit Description

This unit covers a variety of statistical concepts and applications to algorithms and computer science.

The unit covers several topics including: probability, randomised algorithms, statistical inference, clustering, high dimensional data analysis, data mining, and artificial intelligence and Monte Carlo algorithms.
The core aim of this unit is to give the learners additional tools related to statistics and its application in computer science. These can be applied to solve particular problems at the work place or as part of research, such as the research being undertaken in the student's thesis.

At the end of the unit, the student is expected to have applied these tools and be able to identify the correct tool and apply it independently.
For this unit, the lecturer can choose any programming language however a programming language for which the students are familiar is recommended. The lecturer may also use pseudo-code and mathematical notation to describe concepts and algorithms within the unit. Mathematical notation is essential for presenting statistical concepts and probability.

Learning Outcomes

On completion of this unit the learner will be able to:

1. *Understand how statistics and randomised algorithms are used to make inferences based on the data and are applied to algorithms.*
2. *Apply and evaluate statistics and randomised algorithms in contexts related to computer science.*
3. *Use statistical inference to extract information from data.*
4. *Implement randomised algorithms.*

ITSFT-606-1620: Programming for the Cloud

Unit Level (MQF/EQF): 6

Credits: 6

Delivery Mode: Fully Face-to-Face Learning

Total Learning Hours: 150

Unit Description

This subject will add on to the concept of cloud computing. The unit will introduce the motivating factors, benefits, challenges and the service models i.e., software-as-a-service, platform-as-a-service and infrastructure-as-a-service. Moreover, the unit will provide the learner hands-on experience with the commonly found tools within the cloud infrastructure industry such as storage technologies, security measures, highly elastic scalability in delivery of enterprise applications and software as a service (SaaS), caching techniques, and different hosting options.

Practical sessions will be the basis for this unit where a number of technologies will be explored, compared, analysed and then selected to be used within a much larger project to make use of the discussed advantages they will bring about in today's applications.

Moreover, the students taking this unit will be provided with a cloud account were they will undertake the task to configure the necessary settings to make use such mentioned technologies. In the end their work should be deployed on this cloud account.

Learning Outcomes

On completion of this unit the learner will be able to:

1. *Describe the main concepts and benefits behind Cloud Computing.*
2. *Use Cloud Storage solutions to store structured and large data.*
3. *Use other services provided by the Cloud services provider.*
4. *Use Cloud services available to host and consume web applications, APIs or other services.*

ITRSH-606-2102: Research Design II

Unit Level (MQF/EQF): 6

Credits: 6

Delivery Mode: Fully Face-to-Face Learning

Total Learning Hours: 150

Unit Description

The main focus of Research Design II unit is to help learners gain a more in-depth understanding of research design in ICT, thereby, enabling them to evaluate the different ways in which research may be conducted and to choose the approaches that most suit their goals based on already published research and academic theories.

Amongst the most important topics to be covered are the analysis of a hypothesis to better define the scope of a research and to clarify the aim and objectives of a research in view of an applied problem in a specific domain. Knowledge of different research approaches and testing strategies is also paramount; this will lead to the selection and justification of the most appropriate approaches/strategies for the chosen research based on sound academic theories and past research published through conference proceedings and journal articles. Such knowledge will provide the learners with the necessary know-how to present data gathered from results in a manner that is clear and effective for inferring patterns and developing sound and unbiased conclusions with regards to their hypothesis. Critical reflection upon decisions taken throughout the research journey, especially with regards to the chosen research approach, methodology and testing strategies is also an important aspect of this unit for evaluating one's own research in the light of future improvements.

The delivery of the unit should also capitalize on the opportunity to foster a collaborative research environment between learners where they can discuss their research ideas with each other, as peers, and provide constructive criticism/suggestions on how a research idea/approach can be improved. Sharing of knowledge, ideas, opinions and academic resources for carrying out such research is to be encouraged and viewed as an integral part of healthy academic discussion and knowledge sharing.

Proper presentation of the learner's own work carried out during research is also an important part of the unit; tools that support the management of references and the formatting of scientific documents to adhere to well known, pre-defined formats suitable for submission of papers for conference proceedings or articles in scientific journals will be explored.

Learning Outcomes

On completion of this unit the learner will be able to:

1. *Propose the most suitable methodology for a chosen research.*
2. *Analyse testing strategies used for validating a hypothesis.*
3. *Analyse collected data to arrive to findings and conclusions for a chosen research.*
4. *Produce scientific documents using appropriate writing styles, document formats and tools.*

CDWBL-506-1901: Work Based Learning 1

Unit Level (MQF/EQF): 5

Credits: 6

Delivery Mode: Fully Face-to-Face Learning

Total Learning Hours: 150

Unit Description

The aim of this unit is to provide learners reading a degree at MCAST with the opportunity for work-based learning (WBL) with a registered MCAST partner. WBL provides learners with real-life work experiences where they can apply academic and technical skills and develop their employability. Work-based learning deliberately merges theory with practice and acknowledges the intersection of explicit and tacit forms of knowing.

Learners will be doing this unit in the summer of their first year and they will have a follow up unit in the summer of their second year so as to provide them with experience as their knowledge of the subject increases. This will enable them to develop holistically in the area they have chosen enabling them to enter the world of work fully prepared and with experience to show in the sector they have chosen. This unit will assist learners in preparing themselves to take responsibility for their own learning in the workplace and to develop the necessary confidence and attitudes to carry out tasks responsibly in real life work situations. Learners are able to gain practical, hands on experience in their chosen field of study whilst producing a work based learning portfolio and journal demonstrating their achievements and learning experiences.

Learning Outcomes

On completion of this unit the learner will be able to:

1. *Evaluate the relationship between theory, college practice and their application and development in a real world work-based activity.*
2. *Evaluate critically own performance and learning experiences at the place of work through a reflective journal.*
3. *Set SMART objectives for own improvement following the reflective exercise.*
4. *Develop an action plan for personal and professional development to reach set objectives.*

CDWBL-506-1902: Work Based Learning II

Unit Level (MQF/EQF): 5

Credits: 6

Delivery Mode: Fully Face-to-Face Learning

Total Learning Hours: 150

Unit Description

Work-based learning is an instructional method that provides a direct link between work experience and college based learning. A key element in such experiences, is the development of critical thinking. The ability to think critically is fundamental and is sought after by employers in various sectors. Critical thinkers will approach and solve problems methodically rather than by intuition or instinct. Critical thinking is important because it helps individuals and teams more effectively diagnose problems and identify possible solutions that aren't entirely obvious at first. WBL exposes learners to real world environments in order to promote and develop critical thinking. Apprentices, particularly at degree level, are also expected to take initiative and propose solutions to different problems that are faced day to day in various workplace settings. Through their apprenticeship experience, learners are expected to develop strong problem solving skills and use particular incidents as learning opportunities.

Learning Outcomes

On completion of this unit the learner will be able to:

1. *Examine the significance of critical thinking in degree apprenticeships.*
2. *Discuss the role of critical reflection within an experiential learning cycle.*
3. *Apply the IDEALS approach to effective thinking and problem solving.*
4. *Evaluate critical incidents and compile a critical incident journal.*

CDKSK-503-1907: English I

Unit Level (MQF/EQF): 5

Credits: 3

Delivery Mode: Fully Face-to-Face Learning

Total Learning Hours: 75

Unit Description

This unit is intended to be run in the first semester of the first year of undergraduate degree programmes and consolidates prior knowledge, skills and competences in English reading, writing, listening and speaking by further strengthening the more academic functions of the language.

English I is intended to be an EAP (English for Academic Purposes), focusing specifically on improving learners' awareness of, and familiarity, with the core skills necessary for successful academic reading and writing in English, especially preparing them for the rigours of extended writing by research and the reading of academic sources of information.

Learners will become familiar with academic features of style and the principles and mechanics of good text structure. They will also learn how to consult, understand and use secondary material from academic sources within their field of study and effectively integrate it as part of a larger argument or body of work.

Learning Outcomes

On completion of this unit the learner will be able to:

1. *Recognise the form, content and style of academic texts;*
2. *Use an academic style of writing when working on assignments and dissertations;*
3. *Reproduce secondary content by means of direct and indirect quoting methods;*
4. *Apply proper referencing conventions when citing secondary content.*

CDKSK-503-1908: English II

Unit Level (MQF/EQF): 5

Credits: 3

Delivery Mode: Fully Face-to-Face Learning

Total Learning Hours: 75

Unit Description

This unit is intended to be run in the second semester of the second year of undergraduate degree programmes and consolidates prior knowledge, skills and competences of Academic English by further strengthening reading, writing, listening and speaking skills as determined by the rigours of pre-dissertation research.

English II is targeted at learners who have successfully completed their degree programme's first year and exposes undergraduate students to a higher level of critical reading and writing skills demanded in the second and final years of the degree programme. This usually involves the identification and select reading of academic texts, their review and their eventual use in a research proposal, dissertation and academic presentation.

It is also the objective of this unit to train learners to be more aware of, and proficient in, spoken Academic English as this becomes a key requirement at this level of studies.

Learning Outcomes

On completion of this unit the learner will be able to:

1. *Evaluate academic sources of information when working on own dissertation;*
2. *Produce texts of an academic nature using appropriate language and style;*
3. *Communicate verbally in a manner which conveys proficiency of the subject being researched;*
4. *Respond effectively to key questions in relation to research in own field.*

CDKSK-602-2105: Community Social Responsibility

Unit Level (MQF/EQF): 6

Credits: 2

Delivery Mode: Fully Face-to-Face Learning

Total Learning Hours: 50

Unit Description

This unit focuses on community and social responsibility skills and provides an opportunity for learners to better understand themselves and others, as well as establish goals in life. Community and social responsibility skills enable learners to understand their strengths and areas that need improvement while preparing them for life, employment and to become active citizens in society.

Moving away from traditional delivery of other units, learners will be empowered to take ownership of their learning process. Hence, this unit will be delivered through a combination of workshops, small-group sessions with mentors and various opportunities to reflect.

The first set of sessions will focus on the self, the ability to work independently and the important values in life. The second set of sessions will focus on working with others, dealing with diversity and conflicts. Furthermore, at the end of the sessions, learners will be introduced to the importance of active citizenship in life.

Learning Outcomes

On completion of this unit the learner will be able to:

1. *Identify personal goals through self-reflection.*
2. *Evaluate how collaboration with others can be more effective.*
3. *Explain the importance of giving and receiving feedback.*
4. *Contribute actively to make a difference in society.*

CDKSK-604-1909: Entrepreneurship

Unit Level (MQF/EQF): 6

Credits: 4

Delivery Mode: Fully Face-to-Face Learning

Total Learning Hours: 100

Unit Description

The working definition of '**entrepreneurship**' employed in this unit is that stated by the European Commission: "*Entrepreneurship refers to an individual's ability to turn ideas into action. It includes creativity, innovation and taking calculated risk, as well as the ability to plan and manage projects in order to achieve objectives. This supports everyone in day-to-day life at home and in society, makes employees more aware of the context of their work and better able to seize opportunities, and provides a foundation for entrepreneurs establishing a social or commercial activity*" (Entrepreneurship in Vocational Education & Training, June 2009).

In line with this definition, the unit places an emphasis on fostering a mind-set that *entrepreneurship* is the vehicle that drives *creativity* and *innovation*. The learner will, amongst others, be encouraged to gain an insight as to how to investigate customer needs and markets to generate an innovative idea for a start-up; participate in the realistic simulation of the creation of a start-up[1]; create and pitch sections of a business plan, as well as draft sections of a business plan for an identified business idea.

The assessment of the unit is designed in a way to provide an opportunity for learners to strengthen transversal competencies which UNESCO highlights as necessary for the 21st century. These include intrapersonal skills, interpersonal skills, critical and innovative thinking, media and information literacy and global citizenship.

[1] 'Doing effective entrepreneurship' is firmly grounded in theory, yet the *chalk and talk* delivery mode is not promoted in this unit. Rather, *actionable theory through practice* is strongly encouraged. *Realistic simulations*, limited not only to in-class activities such as *discussions* of the problems faced in the different phases of a business, especially in the process of commercialisation of innovative products and services, and *on-paper* creative management strategies, are considered essential.

Learners with different backgrounds and experiences are required to contribute actively in a team to prepare the necessary work towards initiating a successful business venture.
In this unit, learners will become familiar not only with the main theories related to entrepreneurship and business start-ups but will have the opportunity to explore, interact and learn from a number of first-hand situations. The challenges of working with diverse team members will provide the learners not only with the possibility to look at entrepreneurship ideas from different perspectives, but also to come up with more creative, original and feasible solutions to challenges that will arise.

The practical and realistic element of the unit will allow learners to engage and interact with different stakeholders from industry and public institutions. This real-life interaction will provide the ideal set up to link theory with practice in the real world. Learners are encouraged to get out of their comfort zone and explore their entrepreneurial spirit by combining creativity, innovation and risk taking to help seize an opportunity, improve current situations or solve problems they encounter in the real world.

Learning Outcomes

On completion of this unit the learner will be able to:

1. *Understand the terms "entrepreneurship" and "entrepreneur" and techniques used to generate and evaluate business ideas;*
2. *Examine important considerations while developing a new business idea;*
3. *Apply business planning and control initiatives while developing a new business idea;*
4. *Contribute effectively in a team to develop a concept prototype of a feasible product/service idea.*