2021 / 22

School of Science and Computing
Waterford Institute of Technology
☎ +353 (0)51 302037
✉ ereade@wit.ie
🌐 www.wit.ie/science

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

Department *of* Lifelong Learning

Higher Diploma in Science *in* Computer Science

Programme Handbook 2021 /22

WD_KCOSC_G

Award:	Higher Diploma in Science
Specialisation:	Computer Science
Level:	8
Num of stages:	1
Programme Leader:	Eamonn de Leastar

Contents

1 Programme Aim

Programme Aim

The ONLINE HDip in Science in Computer Science is a 24-month ICT Conversion Course designed to allow honours graduates from non-computing disciplines to acquire the industry-relevant ICT and software development skills, expertise and practical experience required to become suitable candidates for employment in the ICT domain in general and in software development in particular.

2 Entry Requirements

2.1 Academic Eligibility

Successful applicants for this course will be existing honours graduates (level 8) with the capacity and aptitude (for example numeracy skills) to complete an intensive Computer Science programme and work placement. Typical disciplines which would fall into this category would be Engineering, Architecture, Mathematics, Physics, and Financial Services. These students are generally strongly suited to the programme.

We have also successfully graduated students with a good general analytic aptitude, with an academic background in Management Information Systems, Economics; Accounting; Business and the Arts. For these students, we request that they sample a taster of the programme consisting of the first 4 weeks of the semester 1 content to ascertain their level of interest and disposition towards the discipline (available here: wit-hdip-comp-sci.github.io). This has provided a more comprehensive screening method, with students self-selecting into the programme if they are comfortable with this experience.

2.2 Recognition of Prior Learning (RPL)

Level 7 ICT graduates have routinely used RPL process to enter the course and thereby upskill to Level 8 for entry into the programme. Download the WIT Springboard RPL Form (https://www.wit.ie/images/uploads/Education_School_PDF/WIT_RP6_Springboard_RPL_Form.docx), fill it out electronically and completely, then submit it back to mailto:jmangan@wit.ie along with all supporting certificates etc.

2.3 English Language Requirements

Applicants whose first language is not English must submit evidence of competency in English, please see WIT's English Language Requirements for details: http://www.wit.ie/about_wit/documents_and_policies/english_language_requirements.

2.4 Funding Eligibility

Employed people will pay a fee of €900 for the course with the Government paying the remaining 90% of the course fee on your behalf. The fee will be payable only at the final stage of the application process when you have been offered a place. If you are unemployed and in receipt of a jobseekers payment (including Farm Assist and Qualified Adults of Working Age) you are not eligible for the two year ICT Conversion courses. However, if you are in receipt of other social protection payments, e.g. One Parent Family or Disability Allowance, you may be eligible for the two-year part-time Conversion programmes. Please consult with your DEASP Case Officer to establish this. Please see more information at www.springboardcourses.ie/eligibility.

3 Overview of Programme

The HDip in Science in Computer Science is designed to produce the programmers and software developers. These skills are in massive demand across multiple sectors including:

- Software Development
- Financial Services
- Online Retailers / Publishers / Services Providers / Customer Support Services
- Manufacturing / Logistics / Industrial Control and Automation
- Advertising & Media

- Enterprise Software Services

Specific skills areas:

- Programming
- Developer Operations
- IT Administration & Management
- Web Development
- Systems Administration
- Technical Support

Sample job title:

- Software Developer
- DevOps Administrator
- System Administrator
- Support Engineer
- Web Designer / Developer
- Business Systems Analyst

3.1 Student Testimonials

> The online Higher Diploma in Computer Science has allowed me to undertake my studies with great flexibility in such a way that a standard face-to-face course could not. I had no prior experience of online learning and my experience has been immensely positive and allowed me to adapt my studies around my full time job. The lecturers are very knowledgeable and they often go out of their way to support any questions or advice. I've been impressed by their methods of delivering the online lectures and the step-by-step content for the accompanying lab work and adopting the latest technologies. I also found the optional onsite days well organised and they were a good opportunity to meet industry partners. The course offers a good variety of modules that cater to the essential skills for emerging software developers.

— Julianne Daly, Data Analyst at VHI

> The ICT Conversion in WIT has been a great innovation. With the current demand for Software Engineers in Ireland and the South East, WIT has succeeded in developing a programme that equips graduates from other disciplines to transfer directly into IT. The fact that graduates are already equipped with excellent problem-solving, team work and communication skills from their previous disciplines makes them very attractive to employers.

— Declan Lawlor; Chief Technical Officer at DoneDeal

> I have a background in IT and being working in the field for 15+ years. Working full time and have a little one at home gives me very little time to upskill, to physically go to a class room. The HDIP online programme program not only allowed me to explore my own areas of interest, while learning valuable technical skills, and providing a thorough grounding in learning theories, current and emerging technologies in this space but above all allowed me to do it in my own time. I didn't feel any disconnect from classmates or faculty members by doing it online. The lecturers are excellent in their own fields, they are extremely patient and helpful and my class mates were always active in our virtual "hangout" to answer questions, bounce ideas off and chat about assignments.

— Juber Nunes; Senior Quality Engineer (Automation & Performance) at Redhat

> Taking the leap and going for the Higher Diploma in WIT's Computer Science was a big move but from the onset I knew I was in good hands. Previously a graduate of Visual Communications I was looking for an avenue into the IT sector and this course came highly recommended. Although broad in scope the content is thorough and often challenging with plenty of room for innovation. Regardless of your background this course will help you find an area of expertise in the sector & with the high-demand for full-stack developers in the south east I can't recommend it highly enough. There's plenty of flexibility in the online delivery which is a game-changer if you're a busy professional or self-employed.

— Jim O'Brien; Business Owner, In Crowd Wedding Music

Programme Schedule

Stage 1

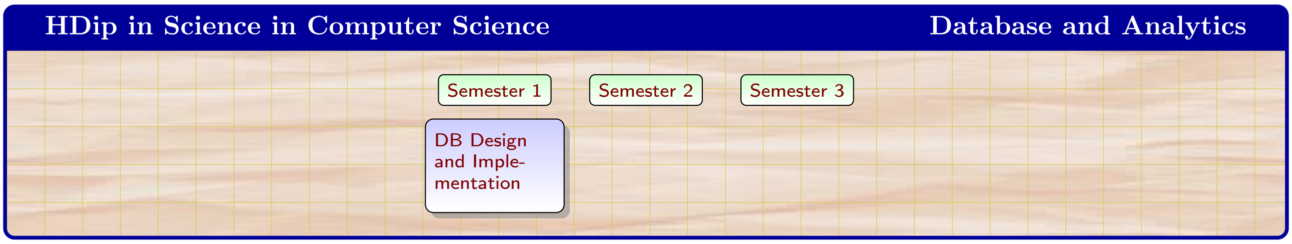
WD_KCOSC_G / Semester 1					Contact Hours					Assessment (%)		
	Module Title	Level	Status	Credits	Lect	Prac	Tut	Other	Total	CA	Proj	Final Exam
A24601 COMP-0975	Computer Systems and Networks	Adv	M	10	48	48			96	75		25
A13705 COMP-0621	Database Design and Implementation	Adv	M	5	24	24			48	50		50
A14167 COMP-0516	Programming Fundamentals - ICT	Intro	M	10	48	48			96	100		
A10923 COMP-0517	Web Development 1	Intro	M	5	24	24			48	100		

WD_KCOSC_G / Semester 2					Contact Hours					Assessment (%)		
	Module Title	Level	Status	Credits	Lect	Prac	Tut	Other	Total	CA	Proj	Final Exam
A11302 COMP-0522	Developer Operations	Adv	M	5	12	36			48	100		
A14173 COMP-0521	ICT Skills Studio	Inter	M	5	24	24			48	100		
A13744 COMP-0620	Computer Security and Forensics	Adv	E	10	48	48			96	100		
A14168 COMP-0523	Mobile Application Development - ICT	Adv	E	10	48	48			96	100		
A14666 COMP-0961	Web App Development	Inter	E	10	48	48			96	100		

Module Descriptors

The modules on the HDip in Science in Computer Science are organised into the following clusters.

Database and Analytics


HDip in Science in Computer Science
Database and Analytics
Semester 1
Semester 2
Semester 3
DB Design and Implementation

Database Design and Implementation (A13705)

Short Title: DB Design and Implementation
Department: Computing and Mathematics
Credits: 5 **Level:** Advanced

Please Note

Due to Covid-19, there may be some changes to the traditional delivery of this module including online delivery of some content and how the assessments methods are implemented. The school office, your programme leaders and lecturers will keep you informed of any changes.

Description of Module / Aims

This module will introduce the student to the principles and practice of designing and implementing database systems. The student will gain competence in designing relational databases using Entity Relationship Modelling. They will implement relational databases using SQL data definition language. They will query the relational database using SQL data manipulation language. The students will be introduced to the concepts and use of NoSQL databases.

Programmes

		stage/semester/status
COMP-0079	Higher Diploma in Science in Agi-Food ICT Systems (WD_18AFICT_G)	1 / 1 / M
COMP-0621	Higher Diploma in Science in Business Systems Analysis (WD_KBUSY_G)	1 / 1 / M
COMP-0621	Higher Diploma in Science in Computer Science (WD_KCOSC_G)	1 / 1 / M
COMP-0621	HDip in Science in Agri-Food ICT Systems (WD_SAFICT_G)	5 / 1 / M

Indicative Content

- Database Environment
- Database Analysis and Design: Relational Modelling, Normalisation, NoSQL databases, CAP theorem
- Physical Database Design: Implementation descriptions of physical database
- Database Implementation

Learning Outcomes

On successful completion of this module, a student will be able to:

1. Appraise the role of a database and its management system.
2. Construct Entity Relationship (ER) diagrams from business scenarios and reproduce those diagrams into normalised relations ready for database implementation.
3. Construct a physical database design.
4. Create a relational database using SQL Data Definition Language (DDL).
5. Construct queries on a relational database using SQL Data Manipulation Language (DML).
6. Assess the suitability and use of NoSQL databases.

Learning and Teaching Methods

- The lectures will introduce the theory content to the student. The student will be encouraged to participate in class discussions and ask questions to support their learning process.
- The practical classes facilitate the student in implementing the theory learned in the lectures using incremental steps to accomplish a skill.
- The continuous assessment will require the student to apply the theory and practical knowledge to a new business scenario.

Learning Modes

Learning Type	F/T Hours	P/T Hours
Lecture	24	
Practical	24	
Independent Learning	87	

Assessment Methods

	Weighting	Outcomes Assessed
Final Written Examination	50%	1,3,6
Continuous Assessment	50%	
Assignment	50%	2,4,5

Assessment Criteria

<40%: Unable to interpret and describe key concepts of the Database design and implementation domain.

40%–49%: Be able to interpret and describe key concepts of the Database design and implementation domain, particularly the use of ER diagrams for the design of relational databases and SQL DDL for the implementation of the relational database.

50%–59%: Ability to discuss key concepts of the Database design and implementation domain and ability to discover and integrate related knowledge in other knowledge domains. To design and develop a suitable relational database system for a business scenario.

60%–69%: Be able to solve basic business analytics problems by experimenting with the appropriate skills and tools. To design and develop a suitable relational database system for a business scenario to a level that includes an enhanced ER diagram and suitable queries appropriate to the business scenario.

70%–100%: All the above to an excellent level. Be able to analyse, design and implement solutions to a high standard for a range of both complex and unforeseen problems through the use and modification of appropriate skills and tools.

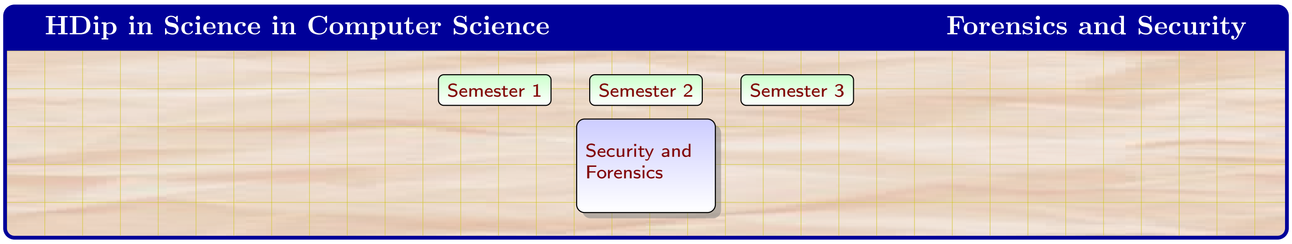
Supplementary Material(s)

- Connolly, T. and C. Begg. *Database Systems: A Practical Approach to Design Implementation and Management*. 5th ed.. NY: Addison Wesley, 2009.
- Sadalage, P. and M. Folwer. *NoSQL Distilled A Brief Guide to the Emerging World of Polyglot Persistance*. NY: Addison-Wesley Professional, 2012.

Requested Resources

- Room Type: Computer Lab

Forensics and Security



HDip in Science in Computer Science

Forensics and Security

Semester 1

Semester 2

Semester 3

Security and Forensics

Computer Security and Forensics (A13744)

Short Title: Security and Forensics
Department: Computing and Mathematics
Credits: 10 **Level:** Advanced

Please Note

Due to Covid-19, there may be some changes to the traditional delivery of this module including online delivery of some content and how the assessments methods are implemented. The school office, your programme leaders and lecturers will keep you informed of any changes.

Description of Module / Aims

This module provides the essentials of computer security and forensics. Topics covered include various security threats and vulnerabilities and the services available to address these threats. Cryptographic foundations that underpin many security mechanisms are covered. Issues relating to network and application security, including web applications, are also explored. Best practice in secure programming is also covered. Students will explore the area of digital forensics through file system forensics, network/online forensics and media forensics.

Programmes

		stage/semester/status
COMP-0620	Higher Diploma in Science in Computer Science (WD_KCOSC_G)	1 / 2 / E

Indicative Content

- Introduction to computer security: security threats; attack methods; common vulnerabilities; security services
- Cryptography: symmetric encryption; modern encryption; block vs stream ciphers; public key cryptography
- Message authentication and integrity: hash functions; collisions; MACs; digital signatures; digital certificates
- Network security: TLS/SSL; wireless security, SSH, firewalls
- Secure Web Application development: SQL Injection; Cross Site Scripting, CSRF
- Security policy and procedures
- Secure coding best practice
- Introduction to computer forensics, the forensic process
- File systems and recovery of data
- Live response
- Network data, types of data, collecting and analysing data from a network
- Web forensics, HTTP headers, cookies, browser / server log analysis, proxy servers, capturing web pages, web form server-side data, web activity reconstruction, DNS
- E-mail forensics, e-mail activity reconstruction, message headers, message attachments, tracing online e-mail
- Mobile device forensics

Learning Outcomes

On successful completion of this module, a student will be able to:

1. Assess the various security threats and attack methods to which an organisation may be susceptible.
2. Appraise the role of cryptography in computer security, including its benefits and limitations.
3. Test and use cryptographic software and configure network and system security tools.
4. Evaluate the specific security concerns pertinent when developing web applications.
5. Recommend security measures when developing code.
6. Utilise forensic tools to analyse a file system and recover deleted data.
7. Perform a live response and gather network data.
8. Investigate web based services/applications.
9. Trace and analyse email data.
10. Collect electronic evidence from modern devices such as smart phones or tablets.

Learning and Teaching Methods

- This module will be presented by a combination of lectures and practicals.
- The lectures will be used to introduce new topics and their related concepts.
- The practical element allows the student to put into practice the theoretical concepts covered in the lectures.

Learning Modes

Learning Type	F/T Hours	P/T Hours
Lecture	48	
Practical	48	
Independent Learning	174	

Assessment Methods

	Weighting	Outcomes Assessed
Continuous Assessment	100%	
In-Class Assessment	25%	1,2,3
Assignment	25%	1,4,5
Assignment	25%	6,10
Assignment	25%	7,8,9

Assessment Criteria

<40%: Unable to describe key network and system security technologies. Unable to distinguish between different types of application vulnerabilities or present instances of them in a clear manner. Unable to effectively use relevant tools. Unable to differentiate stages of the forensic process.

40%–49%: Can describe in detail key security threats and technologies. Can carry out basic configuration of technologies to implement security policies. Able to present instances of vulnerabilities and carry out threat modelling on a basic system. Can conduct basic computer forensic investigations.

50%–59%: In addition to the above, can reason about the various approaches to security and their benefits and limitations. Able to explain in context and present instances of web application vulnerabilities. Able to model threats in a software system with multiple usage scenarios and actors. Can conduct computer forensic investigations and recover deleted data.

60%–69%: In addition, can explain basis of a variety of cryptographic schemes. Can competently make use of security tools and technologies and carry out effective penetration tests. Able to present and explain how to address web application vulnerabilities. Can interpret evidence extracted during a forensic investigation and corroborate it with other sources of evidence.

70%–100%: All of the above to an excellent level. Can demonstrate an understanding of some the trade-offs involved in providing security. Able to demonstrate in detail how to address web application vulnerabilities. Can evaluate the appropriateness of different forensic tools and approaches.

Essential Material(s)

- "Open Web Application Security Project." https://www.owasp.org
- "The OpenSSL project." https://www.openssl.org

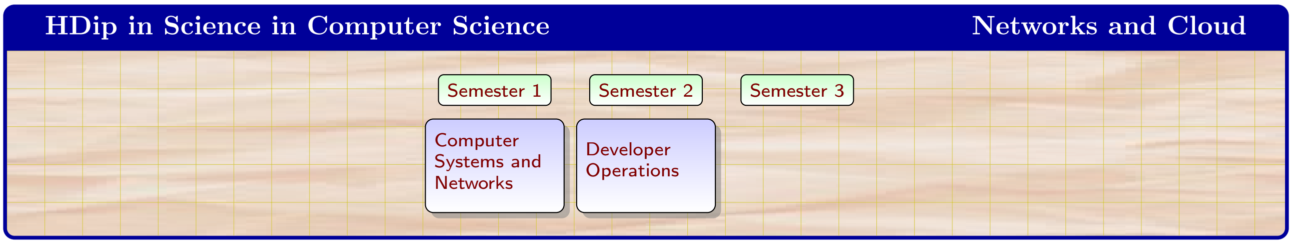
Supplementary Material(s)

- "Computer Emergency Response Team." https://www.cert.org
- "The SANS Institute." https://www.sans.org
- Carrier, B. *File System Forensic Analysis.* Boston: Addison-Wesley, 2005.
- Casey, E. *Handbook of Digital Forensics and Investigation.* Burlington, MA: Elsevier Academic Press, 2010.
- Jones, K., R. Bejtlich and C. Rose. *Real Digital Forensics: Computer Security and Incident Response.* Boston: Addison-Wesley, 2005.
- Jones, R. *Internet Forensics.* Sebastopol, CA: O'Reilly, 2005.
- McGraw, G. *Software Security: Building Security In.* Boston: Addison-Wesley, 2006.
- Stallings, W. and L. Brown. *Computer Security: Principles and Practices.* 3rd ed. Harlow: Pearson, 2014.

Requested Resources

- Computer Lab: BYOD Lab

Networks and Cloud



HDip in Science in Computer Science

Networks and Cloud

Semester 1	Semester 2	Semester 3
Computer Systems and Networks	Developer Operations	

Computer Systems and Networks (A24601)

Short Title:	Computer Systems and Networks		
Department:	Computing and Mathematics		
Credits:	10	**Level:**	Advanced

Please Note

Due to Covid-19, there may be some changes to the traditional delivery of this module including online delivery of some content and how the assessments methods are implemented. The school office, your programme leaders and lecturers will keep you informed of any changes.

Description of Module / Aims

This module introduces technologies and network protocols that underpin computer systems. Topics covered include core operating system functions such as process, memory and file management as well as an introduction to distributed systems and virtualisation. These core principles are then built on to explore key aspects of computer systems such as device interfacing, short range and low powered wireless protocols and higher level messaging protocols. There is an emphasis on applied exercises using physical prototyping devices, command line tools and scripting.

Programmes

		stage/semester/status
COMP-0975	Higher Diploma in Science in Computer Science (WD_KCOSC_G)	1 / 1 / M

Indicative Content

- Number bases used in Computer Science
- Boolean logic
- Computer system architecture
- Operating systems: Components, services, and utilities
- Memory and file management
- Scripting and shell programming
- Virtualisation and hypervisors
- Internet protocol suite
- Physical/network addressing
- Transport layer protocols
- Application layer protocols
- Wireless network protocols: LAN and PAN

Learning Outcomes

On successful completion of this module, a student will be able to:

1. Perform calculations using binary, octal, decimal, and hexadecimal number bases.
2. Apply Boolean logic
3. Describe memory, process and file management components of operating systems
4. Explain computer virtualisation and its relationship to operating systems
5. Install and configure contemporary operating systems, both physically and virtually
6. Write shell scripts for automating tasks, processing data and manipulating files
7. Explain the communication layers in data networks in context of related network protocols
8. Calculate a correct Internet Protocol addressing solutions for IP-based networks
9. Design and build a simple TCP/IP-based computer network using the Internet Protocol Suite
10. Analyse the operations and features of computer networks using a protocol inspection tool
11. Design and implement a connected solution that incorporates Personal Area Network(PAN) and Local Area Network(LAN) protocols.

Learning and Teaching Methods

- This module will be presented by a combination of lectures and computer-based practicals. The lectures will be used to introduce new topics and to underpin their related concepts. The lectures will use case studies as techniques for learning new skills.The practical element will provide the student with the skills, confidence and enthusiasm to be able to apply and reinforce skills learned.A student should be able to demonstrate use of social networking platforms as a form of online interaction
- This module adopts a hands-on practical approach with weekly laboratories that make use of computing devices, virtualisation and cloud-based platforms
- Self directed learning

Learning Modes

Learning Type	F/T Hours	P/T Hours
Lecture	48	
Practical	48	
Independent Learning	222	

Assessment Methods

	Weighting		Outcomes Assessed
Final Written Examination	25%		1,2,3,4,5
Continuous Assessment	75%		
Assignment		35%	5,6,7
Assignment		40%	8,9,10,11

Assessment Criteria

<40% – Fail: Cannot represent the conceptual design of the system components presented in class or explain operation principles at a basic level. Unable to describe the major functions and operation of a Computer Network. Poor understanding of role of communications protocols in computer networks.

40–49% – Pass: Knows the role of each component addressed by the learning outcomes and can represent their conceptual design, supported with a basic narrative description of the operation principles.Can describe TCP/IP and Internet Protocol Suite. Can provide overview of main computer network components and protocols.

50–59% – 2.2: As well as a clear understanding of the components' operation principles, can describe some of the design alternatives covered in the lectures, showing awareness for some of their strengths and weaknesses. Can describe in detail the data encapsulation process. Demonstrate an understanding of basic LAN implementation.

60–69% – 2.1: Can demonstrate a comprehensive understanding of the material covered in the lectures. In addition, be able to design and demonstrate a network solution including LAN and PAN components.

70 – 100% – 1: Excellent understanding of the presented material and displays value added knowledge as a result of independent learning. Be able to analyse and design solutions to a high standard for a range of both complex and unforeseen problems through the use and modification of appropriate skills and tools.

Supplementary Material(s)

- "Association for Computing Machinery." Association for Computing Machinery. http://www.acm.org
- "Cisco Networking Academy." http://www.cisco.com/web/learning/netacad/index.html"
- "IEEE Communications Society." http://www.comsoc.org
- "Raspberry Pi Foundation." https://www.raspberrypi.org/
- Tanenbaum, A. *Computer Networks.* 5. London: Prentice Hall, 2010.

Requested Resources

- Computer Lab: BYOD Lab

Developer Operations (A11302)

Short Title: Developer Operations
Department: Computing and Mathematics
Credits: 5 **Level:** Advanced

Please Note

Due to Covid-19, there may be some changes to the traditional delivery of this module including online delivery of some content and how the assessments methods are implemented. The school office, your programme leaders and lecturers will keep you informed of any changes.

Description of Module / Aims

This is a practical module that requires the student to build, configure and manage the operating systems and network infrastructure required for a typical cloud application environment.

Programmes

		stage/semester/status
COMP-0559	BSc (Hons) in Applied Computing (WD_KACCM_B)	3 / 5 / M
COMP-0559	BSc (Hons) in Computer Forensics and Security (WD_KCOFO_B)	3 / 5 / M
COMP-0559	BSc (Hons) in Applied Computing (WD_KCOMP_B)	3 / 5 / M
COMP-0522	Higher Diploma in Science in Computer Science (WD_KCOSC_G)	1 / 2 / M
COMP-0559	BSc (Hons) in Entertainment Systems (WD_KENTS_B)	3 / 5 / M
COMP-0559	BSc (Hons) in the Internet of Things (WD_KINTT_B)	3 / 5 / M

Indicative Content

- Cloud Computing Architectures and Services
- Public Cloud Services: Storage; Compute; Networking
- Configuration of Multi-tier Application Infrastructure Services
- Cloud APIs – Python or similar
- Virtual Private Clouds
- Web Application Architecture – Performance, Scaling, Load Balancing and Security
- Automation and scripting – using for example bash (advanced), Python, PowerShell, Chef, Ansible
- Developer Operations (DevOps) tools and configuration
- Network and Application Management and Monitoring

Learning Outcomes

On successful completion of this module, a student will be able to:

1. Build, configure and manage essential network infrastructure and application services.
2. Deploy a network monitoring solution.
3. Develop scripts to assist in the management and automation of modern network services.
4. Analyse application performance, scalability, load balancing and security.
5. Compare and contrast the main technologies required to develop and manage Cloud based Application Infrastructure.

Learning and Teaching Methods

- The practical lab component will be delivery in a double lab session.
- Strong emphasis on practical laboratory exercises with extensive use made of virtualised environments.
- Self-directed learning.

Learning Modes

Learning Type	F/T Hours	P/T Hours
Lecture	12	
Practical	36	
Independent Learning	87	

Assessment Methods

	Weighting	Outcomes Assessed
Continuous Assessment	100%	
Assignment	40%	1,3
Assignment	40%	1,2,3,4,5
In-Class Assessment	20%	4,5

Assessment Criteria

<40%: Unable to build and configure basic infrastructure services to meet assignment requirements. Unable to interpret and describe key concepts of the specific knowledge domains of Python, Cloud Application Infrastructure services and automation.

40%–49%: Can build and configure basic infrastructure services to meet assignment requirements. Be able to interpret and describe key concepts of the specific knowledge domains of Python, Cloud Application Infrastructure services and automation.

50%–59%: Can discuss key concepts of the specific knowledge domains covered above and ability to discover and integrate related knowledge into cloud based application architectures.

60%–69%: In addition, be able to solve problems within the specific knowledge domain(s) by experimenting with the appropriate skills and tools.

70%–100%: All the above to an excellent level. In addition, demonstrate a deep understanding of the building, deployment and management of a Multi-tier web application infrastructure.

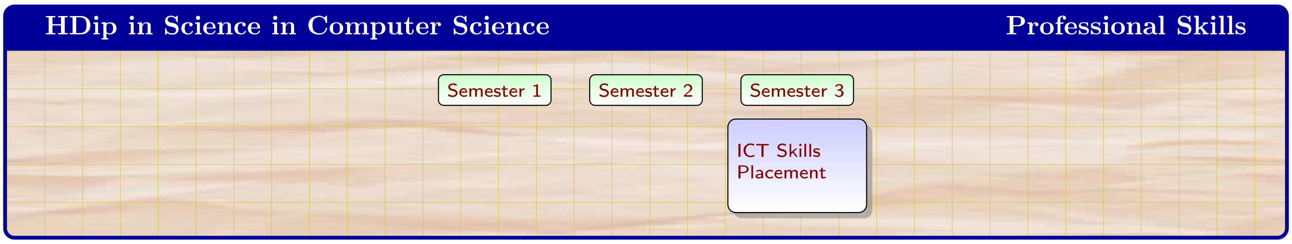
Supplementary Material(s)

- "The Python Wiki." https://wiki.python.org/
- "boto: Python interface to Amazon Web Services." http://boto.readthedocs.org/en/latest/
- Amazon, Amazon. *Getting started with AWS (eBook).* NY: Amazon Web Services, 2014.
- Garnaat, M. *Python and AWS Cookbook.* 1st Ed. NY: O'Reilly, 2012.
- Kim, G., K. Behr and G. Spafford. *The Phoenix Project: A Novel about IT, DevOps, and Helping Your Business Win.* New York: IT Revolution Press, 2013.
- Loukides, M. *What is DevOps? (ebook).* NY: O'Reilly, 2012.
- Morris, K. *Infrastructure as Code: Managing Servers in the Cloud.* 1st. New York: O'Reilly Media, 2016.

Requested Resources

- Computer Lab: BYOD Lab

Professional Skills



HDip in Science in Computer Science

Professional Skills

Semester 1

Semester 2

Semester 3

ICT Skills Placement

ICT Skills Placement (A14887)

Short Title:	ICT Skills Placement		
Department:	Computing and Mathematics		
Credits:	30	**Level:**	Advanced

Please Note

Due to Covid-19, there may be some changes to the traditional delivery of this module including online delivery of some content and how the assessments methods are implemented. The school office, your programme leaders and lecturers will keep you informed of any changes.

Description of Module / Aims

Students will spend a period of not less than four months in a work experience position in an ICT and Software Development workplace. The placement allows the student to gain a structured introduction to the work practices, requirements and procedures of the industry. Students will have the opportunity to consolidate, develop and enhance their learning experience through the practical application knowledge, skills and competencies acquired on the academic to the analysis and creation of solutions to workplace problems through a structured programme of work agreed the during the studio phase. This will ensure students have a clear focus and their supervision has a specific context. As the project will be graded, students will have access to an academic project supervisor during the placement period.

Programmes

		stage/semester/status
PLAC-0128	Higher Diploma in Science in Computer Science (WD_KCOSC_G)	1 / 3 / M

Indicative Content

- Completion of agreed supervised project.

Learning Outcomes

On successful completion of this module, a student will be able to:

1. Apply knowledge, skills and competencies acquired on the academic to the analysis and creation of solutions to workplace problems.
2. Contextualise the knowledge gained in the programme in an area relevant a selected area of interest.
3. Communicate effectively in an appropriate and professional manner and format.
4. Work as a member of a team and have developed appropriate communication and interpersonal skills.
5. Reflect on and analyse the learning experience resulting from the work placement.
6. Communicate the aims, expectations and objectives of all parties to the placement.

Learning and Teaching Methods

- Students will undergo an induction briefing prior to the beginning of the placement and have access to an academic supervisor during the placement period.
- The applied learning portfolio is designed to foster self-reflection on the learning gained through participation in the work placement.

Learning Modes

Learning Type	F/T Hours	P/T Hours
Tutorial	6	
Placement	540	
Independent Learning	264	

Assessment Methods

	Weighting	Outcomes Assessed
Final Project	100%	1,2,3,4,5,6

Assessment Criteria

<40%: The student has failed to meet the criteria for a pass, but has not submitted a project which can demonstrate a reasonable grasp of the fundamentals of the chosen project area.

40%–49%: The student has met the criteria for a pass, and demonstrates a reasonable grasp of the fundamentals of the chosen project. Typically a project more supervisor-driven than student-led. The project is unlikely to be the basis for a portfolio.

50%–59%: A solid performance though notably stronger in some respects than others. A good standard of report but neither technically outstanding nor grounded in deep understanding of the relevant technology/domain. Sound but unremarkable. Would require significant additional effort in order to be a strong constituent in a portfolio. However, the student has the abilities to take it to this level, perhaps with additional supervision / advice from domain experts.

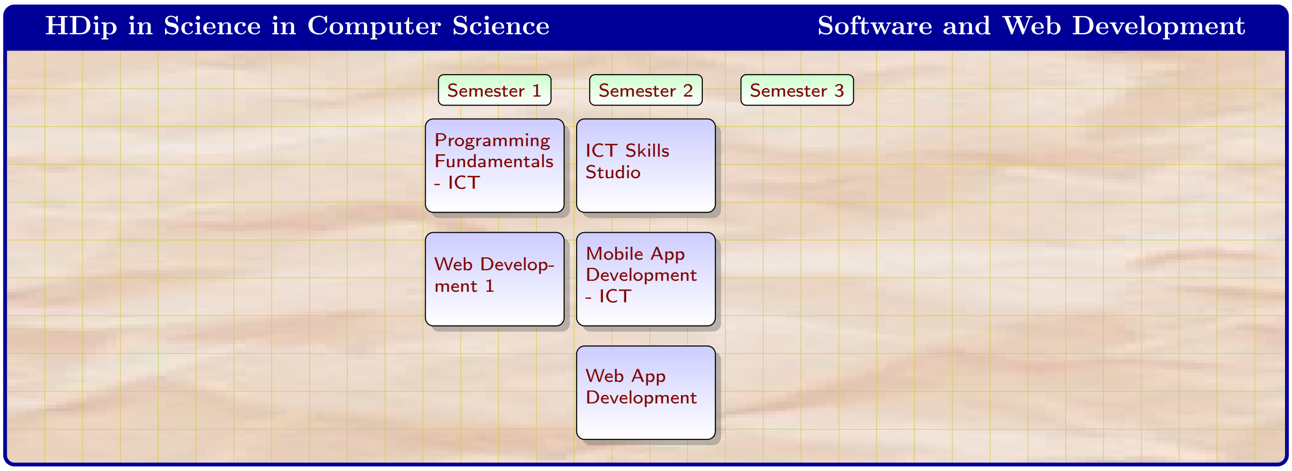
60%–69%: A student in this band will have shown a high level of independence in the conduct of the project and technical competence in any practical work undertaken. A strong performance may have been achieved in all components of assessment. However some aspects of the project would require some additional attention before publication on a portfolio that would do justice to the student's skills and abilities.

70%–100%: A mark in this band indicates a distinction. A non-trivial project has been executed and the resulting report + implementation (if appropriate) is an exemplary representation of the students skill and abilities. The material produced will serve as a strong demonstrator and / or portfolio highly relevant future employers in assessing the suitability of the student for recruitment. It can be directly incorporated into such a portfolio, perhaps with minor modifications.

Supplementary Material(s)

- Herbert, I. *Managing your placement: a skills-based approach.* London: Macmillian, 2004.
- Sheridan, I. *REAP (Roadmap for Employment - Academic Partnerships) - Work placement in third-level..* Cork: CIT Press, 2004.

Software and Web Development



HDip in Science in Computer Science — **Software and Web Development**

Semester 1	Semester 2	Semester 3
Programming Fundamentals - ICT	ICT Skills Studio	
Web Development 1	Mobile App Development - ICT	
	Web App Development	

ICT Skills Studio (A14173)

Short Title: ICT Skills Studio
Department: Computing and Mathematics
Credits: 5 **Level:** Intermediate

Please Note

Due to Covid-19, there may be some changes to the traditional delivery of this module including online delivery of some content and how the assessments methods are implemented. The school office, your programme leaders and lecturers will keep you informed of any changes.

Description of Module / Aims

Continue to develop the students' programming skills with a particular focus on modern web development tools, frameworks and applications. A strong emphasis industry best practice - with close attention software configuration management tools. The Studio will also support a forum for talks and seminars from industry on technology trends and work practices.

Programmes

		stage/semester/status
COMP-0521	Higher Diploma in Science in Computer Science (WD_KCOSC_G)	1 / 2 / M

Indicative Content

- Consolidating programming principles & practices
- Elementary algorithms & data structures
- Web development applications design & implementation
- Server side and client side aspects of web wevelopment
- Current trends in software development technology

Learning Outcomes

On successful completion of this module, a student will be able to:

1. Construct a simple but robust server-rendered web application.
2. Differentiate the role of the client and server in this context.
3. Break a problem domain into a series of discrete features.
4. Manage the assets of a project in a configuration management environment.

Learning and Teaching Methods

- Supervised, guided and scripted practicals will lead the student through the construction of selection of small applications, designed to illustrate key concepts covered in the lectures.
- Worked Laboratory Exercises.
- Short Lectures to outline specific concepts.

Learning Modes

Learning Type	F/T Hours	P/T Hours
Lecture	24	
Practical	24	
Independent Learning	87	

Assessment Methods

	Weighting	Outcomes Assessed
Continuous Assessment	100%	
Assignment	100%	1,2,3,4

Assessment Criteria

<40%: Unable to interpret, describe or implement key components of a simple web application.

40%–49%: Be able to construct a minimal web application, incorporating server side rendering.

50%–59%: Design and implement a web application including basic database and session support.

60%–69%: All of the above including a simple model of the application structure.

70%–100%: All the above to an excellent level. Incorporate Unit Tests for a specific subset of an application.

Supplementary Material(s)

- Richard-Foy, J. *Play Framework Essentials.* New York: Packt Publishing, 2014.
- Syed, B. *Beginning Node.js.* New York: Apress, 2014.

Requested Resources

- Computer Lab: BYOD Lab

Mobile Application Development - ICT (A14168)

Short Title: Mobile App Development - ICT
Department: Computing and Mathematics
Credits: 10 **Level:** Advanced

Please Note

Due to Covid-19, there may be some changes to the traditional delivery of this module including online delivery of some content and how the assessments methods are implemented. The school office, your programme leaders and lecturers will keep you informed of any changes.

Description of Module / Aims

Design, build and deploy a multi-screen mobile application incorporating an intuitive and efficient navigation mechanism. Structure the implementation using accepted best-practice with respect to patterns, frameworks and tools. Incorporate localised persistence models + simple access to remote services. Introduce context services such as location/camera and/or other sensor access. Evolve a multi-screen mobile application into a networked, message driven, context aware application. Incorporate in the application two-way access to remote REST (Representational State Transfer) and Messaging services. Integrate on-device context including camera, location, motion, climate and other sensors to deliver a rich user experience. Incorporate 3rd party components to deliver personalized mapping, media and general information services.

Programmes

		stage/semester/status
COMP-0523	Higher Diploma in Science in Computer Science (WD_KCOSC_G)	1 / 2 / E

Indicative Content

- Application Structure: Components; Resources; Security; General Assets
- User Experience: UX (User Experience) Principles, Navigation, Imagery, Fonts
- Simple User Interaction Patterns
- Essential Application Structure Patterns: Appropriate Variations on Model/View/Controller (MVVM (Model–view–viewmodel), MVP (Model–view–presenter) etc.)
- Resource access and management; Clean separation of concerns
- Application Life-cycle: Startup/shutdown; Foreground/background
- UI (User Interface) State Preservation and Restoration; Concurrency
- Advanced application architectural patterns
- The build, test & deploy lifecycle
- Accessing Platform Services: Persistence; Sensors / Subsystems (e.g. Location, camera, movement etc.)
- Accessing External Services: Access Patterns (e.g. REST); Third Party Applications & Components
- Build Processes: Dependency Management; Build Scripts (e.g. Gradle)
- Wireless Subsystem APIs (Application Programming Interface)
- App Store interaction, including key management

Learning Outcomes

On successful completion of this module, a student will be able to:

1. Decompose an application into its constituent parts, including but not limited to: core application components, user experience resources, packaging.
2. Design a coherent User Experience - using appropriate tools, practices and guidelines - for a moderately sized application.
3. Produce a medium sized application, based on a limited set of design patterns.
4. Manage the application lifecycle.
5. Structure persistent storage on a device and reliably save and restore application state.
6. Select the appropriate design patterns and tools in the development of complex mobile apps.
7. Comment on the chosen mobile app framework and the underlying hardware components.
8. Design and develop complex multi-screen mobile apps from concept through to completion using best practices and guidelines.
9. Set up the interaction of an application with internal sensors and physical subsystems.
10. Integrate a remote service API within an application, perhaps based on REST principles, to deliver aspects of its core features set. For example: Maps/GIS (Geographic Information Systems), Media Sharing, Social Networking.

Learning and Teaching Methods

- Lectures will introduce the general context of the curriculum, and explore specific topics in depth.
- Supervised, guided and scripted practicals will lead the student through the construction of an application designed to illustrate key concepts covered in the lectures.
- The focus is on learning by doing in a studio environment. Each practical will propose a set of exercises – to be solved in a subsequent practical.
- Assignment One will focus ensuring the student can construct a new application equivalent in style and structure to the guided practical.
- Assessment Two will invite the student to analyse, design and implement a new application.

Learning Modes

Learning Type	F/T Hours	P/T Hours
Lecture	48	
Practical	48	
Independent Learning	174	

Assessment Methods

	Weighting		Outcomes Assessed
Continuous Assessment	100%		
Assignment		40%	1,2,3,4
Assignment		60%	5,6,7,8,9,10

Assessment Criteria

<40%: Unable to implement a basic application. Cannot grasp fundamentals of the application lifecycle or operate an appropriate IDE (Integrated Development Environment).

40%–49%: Be able to implement at simple application, with 2-3 separate views/activities.

50%–59%: Understand the basic of the application lifecycle and operate an IDE at a basic level. Ability to model and implement an application of moderate complexity − including $>$ 3 views + a simple persistence mechanism.

60%–69%: Be able to use an IDE competently and degug applications. Be able to implement a reasonably sophisticated application with multiple view / navigation mechanisms. The application will have local persistent storage and be able to interact with a remote service as a basic level (read only say).

70%–100%: All the above to an excellent level. Be able to build an application that can access on device sensors / subsystems (e.g. location or camera).

Essential Material(s)

- "Android Developer Resources." http://developer.android.com
- "Apple Developer Resources." http://developer.apple.com/ios
- "Cordova Developer Resources." https://cordova.apache.org

Supplementary Material(s)

- Camden, R. *Apache Cordova in Action.* New York: Manning, 2015.
- Neuburg, M. *iOS 9 Programming Fundamentals with Swift: Swift, Xcode, and Cocoa Basics.* New York: O'Rielly, 2015.
- Phillips, B. *Android Programming: The Big Nerd Ranch Guide.* New York: Pearson, 2015.

Requested Resources

- Computer Lab: BYOD Lab

Programming Fundamentals - ICT (A14167)

Short Title: Programming Fundamentals - ICT
Department: Computing and Mathematics
Credits: 10 **Level:** Introductory

Please Note

Due to Covid-19, there may be some changes to the traditional delivery of this module including online delivery of some content and how the assessments methods are implemented. The school office, your programme leaders and lecturers will keep you informed of any changes.

Description of Module / Aims

This module will introduce the student to fundamental programming techniques. The student will learn how to design algorithms using pseudocode, write small applications based on their designs and test their written applications. It assumes no prior knowledge of technology, programming languages or programming environments. Using basic techniques, the student should gain confidence in producing small applications. The module will alsofurther develop the thought processes, problem solving techniques and programming structures essential for developing larger systems responsible for more complex tasks. The module will focus on developing maintainable, robust, persistent applications that are consistent with user experience (UX) best practice.

Programmes

		stage/semester/status
COMP-0516	Higher Diploma in Science in Computer Science (WD_KCOSC_G)	1 / 1 / M

Indicative Content

- Problem solving approaches
- Control structures for programming
- Programming language constructs
- UX approaches using data-entry validation and error recovery
- Hands-on debugging, testing and documentation activities
- Use of innovative visualisation tools, IDEs and frameworks
- Introduction to industry accepted standards specific to the programming language
- Use of complex constructs and data structures associated with the chosen language
- Introduction to UX
- Programming for persistence
- Programming techniques for well-behaved, robust applications
- Read, understand and consume API specific to the chosen language
- Use of innovative visualisation tools, IDEs and frameworks

Learning Outcomes

On successful completion of this module, a student will be able to:

1. Apply core problem solving approaches suitable to the programming discipline to build algorithms.
2. Construct small applications using standard sequence, conditional and iterative control structures.
3. Change and expand small applications.
4. Construct small applications that use simple UI, computation and data structures.
5. Apply techniques to effectively test, debug and document small applications.
6. Defend and explain how the above applications work.
7. Apply problem-solving strategies to various computing problems of increasing complexity.
8. Plan, code, test and document applications using advanced programming constructs and data structures.
9. Construct applications consistent with UX best practice.
10. Construct persistent applications.
11. Apply maintainability and robustness when designing applications.

Learning and Teaching Methods

- Combination of lectures and computer-based practical labs.
- Cooperative learning/peer tutoring (i.e. pair-programming for some practical labs, Problem-based learning approaches for some assignments).
- Self-directed learning.

Learning Modes

Learning Type	F/T Hours	P/T Hours
Lecture	48	
Practical	48	
Independent Learning	174	

Assessment Methods

	Weighting	Outcomes Assessed
Continuous Assessment	100%	
Assignment	30%	1,2,3
Assignment	35%	3,4,5,6
Assignment	35%	5,6,7,8,9,10,11

Assessment Criteria

<40%: Inability to design, develop and test maintainable, persistent, robust UX applications to solve a particular problem.

40%–49%: Ability to design, develop and test maintainable, persistent, robust UX applications to solve a straight-forward problem.

50%–59%: Comfortable with designing, developing and testing maintainable, persistent, robust UX applications to solve problems similar to those presented in the module.

60%–69%: Proficient with designing, developing and testing maintainable, persistent, robust, high-quality UX applications to solve complex problems.

70%–100%: Proficient with designing, developing and testing maintainable, persistent, robust, high-quality, elegant UX applications to solve complex problems that are substantially different to those studied in the module.

Supplementary Material(s)

- Kolling, M. *Objects first with Java-A Practical Introduction using BlueJ. 5th ed...* New York: Pearson Education, 2012.
- Sprankle, M. *Problem Solving and Programming Concepts.* New York: Prentice Hall, 2011.

Requested Resources

- Computer Lab: BYOD Lab

Web App Development (A14666)

Short Title: Web App Development
Department: Computing and Mathematics
Credits: 10 **Level:** Intermediate

Please Note

Due to Covid-19, there may be some changes to the traditional delivery of this module including online delivery of some content and how the assessments methods are implemented. The school office, your programme leaders and lecturers will keep you informed of any changes.

Description of Module / Aims

Introduce the the student to the software development lifecycle via the implementation of a simple but functional web application. In doing this, analyse & model a constrained set of user requirements. Then design, build and deploy a simple web application. Incorporate basic database, session support & server side rendering. Evolve this understanding within a Services context. REST APIs (Representational State Transfer Application Program Interface), with multiple service consumer forms will be considered, including Single Page Apps (SPA) and other services. The principles and patterns underpinning the design of both components (SPA and REST API) will be examined as well as the fine-grained aspects of the underlying communication protocol. Their will be an emphasis on development, including the use of application frameworks, workflow automation tools and cloud deployment platforms. The module?s scope will also encompass security concerns and techniques.

Programmes

		stage/semester/status
COMP-0961	Higher Diploma in Science in Computer Science (WD_KCOSC_G)	1 / 2 / E

Indicative Content

- User Stories & Agile context
- Introduction to Modelling
- Hypertext Transfer Protocol (HTTP) Request/Response Life Cycle
- Introductory Web Application Frameworks
- Simple Object Relational Mapping tools
- Test Driven Development
- Fundamentals: Architecture patterns, HTTP (Hypertext Transfer Protocol) protocol, Advanced Javascript
- API Design patterns and principles– REST, CQRS (Command Query Responsibility Separation) Versioning, Security, Hypermedia, Realtime
- SPA design patterns and principles – MV* (Model View *), Flux, Caching, data synchronisation
- Application Frameworks – Web API, Single Page App, Isomorphic app
- Developer tool suite – API modeling DSL(Domain Specific Language) , Scaffolding, workflow automation
- Security principles related to web development: cryptography; authentication and digital certificates
- Web application vulnerabilities; penetration testing
- Web application protections: input & output validation; various authentication techniques (e.g. cookies, OAuth, JWT, CSRF tokens); secure credential handling

Learning Outcomes

On successful completion of this module, a student will be able to:

1. Examine the key components of a server rendered web application and incorporate them into a running application.
2. Use Model View Controller & related patterns in the implementation of a web project.
3. Relate the request/response lifecycle, routing & session management in the context of a modern application framework.
4. Break down a set of requirements into a set of discrete stories and translate these stories into a simple project plan with associated timeline and testing strategy.
5. Model the user requirements and realize the model in a simple database.
6. Apply best practice principles and patterns to the design and documentation of a web API.
7. Apply best practice principles and patterns to the design of a medium-sized Single Page Web App.
8. Develop an end-to-end web app that supports session management and persistence for a constrained functional requirement set.
9. Demonstrate specific security problems that can arise with web applications and how to address them.
10. Compare and contrast alternative approaches to authentication in both enterprise and consumer-oriented web applications.
11. Use a selection of best security practices in a web application.

Learning and Teaching Methods

- Combination of lectures and computer-based practicals.

Learning Modes

Learning Type	F/T Hours	P/T Hours
Lecture	48	
Practical	48	
Independent Learning	174	

Assessment Methods

	Weighting		Outcomes Assessed
Continuous Assessment	100%		
Assignment		50%	1,2,3,4,5,6,7
Assignment		50%	5,6,7,8,9,10,11

Assessment Criteria

<40%: Unable to interpret and describe key concepts of modern web app development.

40%–49%: Be able to interpret and describe key concepts of modern web app development.

50%–59%: Ability to demonstrate competency in the tool suite and the ability to develop and deploy small-scale solutions.

60%–69%: Presents implemented solutions to medium-sized problems that demonstrate a good understanding of the main patterns and practices of web app design.

70%–100%: All the above to an excellent level.

Supplementary Material(s)

- Holmes, S. *Getting MEAN with Mongo, Express, Angular, and Node.* New York: Manning, 2015.
- Richardson, L. *RESTful Web APIs.* New York: O'Rielly, 2015.

Requested Resources

- Computer Lab: BYOD Lab

Web Development 1 (A10923)

Short Title: Web Development 1
Department: Computing and Mathematics
Credits: 5 **Level:** Introductory

Please Note

Due to Covid-19, there may be some changes to the traditional delivery of this module including online delivery of some content and how the assessments methods are implemented. The school office, your programme leaders and lecturers will keep you informed of any changes.

Description of Module / Aims

Introduces the student to the fundamental building blocks of the visual aspect of the Web, with an emphasis on the basics of HTML, CSS and Java Script. These topics will be explored collectively to give the student the ability to design dynamic and responsive web micro-sites. Leverage this knowledge and apply it to Contact Management Systems (e.g. WordPress, Django) "mash-up" toolkits (e.g. Yahoo pipes, Zembly), to enable the construction of simple web applications and widgets (Facebook)

Programmes

		stage/semester/status
	BSc (Hons) in Computer Forensics and Security (WD_KCOFO_B)	1 / 1 /
	BSc in Software Systems Development (WD_KCOMC_D)	1 / 1 /
	BSc (Hons) in Applied Computing (WD_KCOMP_B)	1 / 1 /
COMP-0517	HDip in Science in Computer Science (WD_KCOSC_G)	1 / 1 / M
	BSc (Hons) in Entertainment Systems (WD_KENTS_B)	1 / 1 /

Indicative Content

- Basic Document Constructions: Paragraphs; Line breaks; Headings; Images; Hyperlinks; Lists; Tables; Forms; Frames.
- Cascading Style Sheets: Syntax and Rules.
- Basic Scripting: Simple JavaScript; Variables; Functions; Conditions; Loops and Repetition; Arrays
- The Document Object Model: Nodes, Manipulation
- The role and purpose of Content Management Systems.
- Widget/Mash-up development. The use of Web2.0 style toolkits and integration frameworks.

Learning Outcomes

On successful completion of this module, a student will be able to:

1. Describe the fundamentals of the HTML markup language.
2. Describe the role of Human Computer Interaction and manipulate CSS to present HTML content.
3. Read and adjust simple java script fragments and describe their impact on presentation.
4. Integrate HTML, CSS and Java script to structure simple micro sites and widgets.
5. Describe the role of a Content Management System and be able to integrate and manage simple CMS extensions.
6. Construct simple toolkit based web widgets/applications.

Learning and Teaching Methods

- This course will be delivered via a combination of lecture and studio style practicals.
- The lectures will be ensuring the student has an appreciation of the essential formalisms: mark-up languages and simple scripting.
- Photoshop or equivalent tools will be the focus of the initial programme. However, simple experiments in mash-up and widget development will involve exploration of representative environments and web based tools.
- Practicals will centre on putting these formalisms to work in the creation of micro sites.
- Attention will be paid to the effective use of appropriate tools.

Learning Modes

Learning Type	**F/T Hours**	**P/T Hours**
Lecture	24	
Practical	24	
Independent Learning	87	

Assessment Methods

	Weighting		**Outcomes Assessed**
Continuous Assessment	100%		
Assignment		50%	1,2,4
Assignment		50%	3,4,5,6

Assessment Criteria

<40%: Attention will be paid to the effective use of appropriate tools.

40%–49%: Able to combine HTLM, CSS and simple Java script fragments in a simple micro-site.

50%–59%: All of the above and in addition be able to implement a more sophisticated scripting capability.

60%–69%: Demonstrate an understanding of an encompassing platform, based around a simple CMS. Be able to introduce some simple new features into such a platform using the knowledge gained in the course. Attempt some simple mash-ups.

70%–100%: Be able to construct more ambitious mash-ups/widgets - incorporating HTML, CSS and JavaScript - with some aspect of server side processing.

Essential Material(s)

- Anderson & Anderson, A. *Assemble the Social Web with Zembly.* NY: Prentice-Hall, 2008.
- Crockford, M. *The Good Parts.* NY: Javascript, 2009.
- Robson & Freeman, M. *Head First HTML with CSS & XHTML.* NY: O'Reilly, 2005.

Requested Resources

- Room Type: Computer Lab