

School of Lifelong Learning & Education

Department of Lifelong Learning

Higher Diploma in Science *in* Computer Science

Programme Handbook 2017–18

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 | 1 |
| 2 | Programme Learning Outcomes (PLO) | 1 |
| 3 | Entry Requirements | 2 |
| 4 | Overview of Programme | 3 |
| 5 | Programme Schedule | 4 |
| 6 | Module Descriptors | 6 |
| 6.1 | Database and Analytics | 7 |
| 6.2 | Forensics and Security | 10 |
| 6.3 | Networks and Cloud | 14 |
| 6.4 | Professional Skills | 21 |
| 6.5 | Software and Web Development | 24 |

1 Programme Aim

Programme Aim

The aim of the HDip in Science in Computer Science is to equip graduates in cognate disciplines with the knowledge & skills in Computer Science to a level commensurate with an ability to function as a programmer in a modern software development environment.

2 Programme Learning Outcomes (PLO)

On successful completion of this programme, all graduates will have achieved the Programme Learning Outcomes that are set out in Table 2.1. The Programme Learning Outcomes have been devised by the programme team in accordance with the QQI Computing Standards.

Table 2.1: Program level outcomes (PLO) for HDip in Science in Computer Science.

| On successful completion of the programme the learner should be able to: | |
|--|---|
| PLO 1 | Demonstrate knowledge and understanding of the theory, core skills, and current best practice in the field of computer programming, web development, databases, computer systems & network. |
| PLO 2 | Apply competence in computer programming and in associated discipline of developer operations. Critically relate these fields to database fundamentals, computer systems & networks. |
| PLO 3 | Acquire and assess a detailed knowledge of core module competencies and 2 chosen specialisms from Mobile Application Development, Enterprise Service Development and Security & Forensics. |
| PLO 4 | Integrate mastery of the theory, associated tool-chains and suitable real-world ICT problems, to enable the design, test and implementation of appropriate software application solutions. |
| PLO 5 | Apply programming and software development knowledge and skills in the design, development and implementation of computing and information systems. |
| PLO 6 | Initiate, lead and manage multi-disciplinary projects. Work effectively as an autonomous individual or as part of a team in a professional software development role. |
| PLO 7 | Demonstrate the ability to conduct professional research, critically review performance and delivery and proactively manage the one's on-going professional development. |
| PLO 8 | Develop, maintain and exercise an informed appreciation of ethical standards and professional judgement. |

3 Entry Requirements

The entry requirement for the HDip in Science in Computer Science is a level 7 computing degree.

4 Overview of Programme

The programme is comprised of an intensive taught course of core programming and computer science modules followed by a 6-month work placement. It includes a combination of fundamental concepts and practical exposure to problem solving, programming, and the broader software development ecosystem (increasingly known as “DevOps”), including system administration and security. As stated above it is designed to take candidates from non-computing backgrounds, who have the aptitude to tackle an intensive computer science course, and allow them to build the skills and experience to become employable as junior programmers within an 18-month window.

The layout of the modules are displayed by subject group in Figure 4.1.

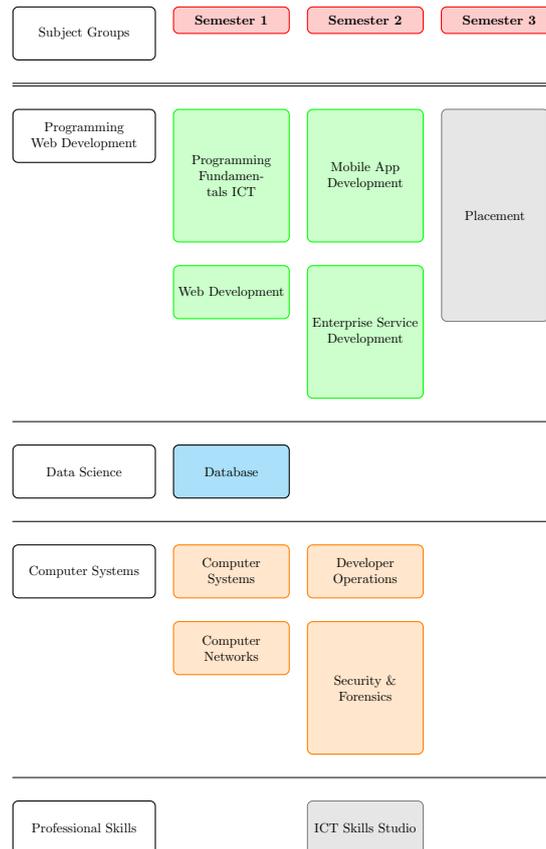


Figure 4.1: Modules organised by subject group.

Programme Schedule

Stage 1

| WD_KCOSC_G / Semester 1 | | Level | Status | Credits | Contact Hours | | | | | Assessment (%) | | |
|-------------------------|------------------------------------|-------|--------|---------|---------------|-----|-------|-------|----|----------------|------------|--|
| Module Title | Lect | | | | Prac | Tut | Other | Total | CA | Proj | Final Exam | |
| A13487 COMP-0519 | Computer Networks (ICT Skills) | Adv | M | 5 | 12 | 36 | | | 48 | 50 | 50 | |
| A13622 COMP-0518 | Computer Systems (ICT Skills) | Adv | M | 5 | 24 | 24 | | | 48 | 50 | 50 | |
| A13705 COMP-0079 | 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 | | |
| A13422 | Website Development 1 | Intro | M | 5 | 12 | 36 | | | 48 | 100 | | |

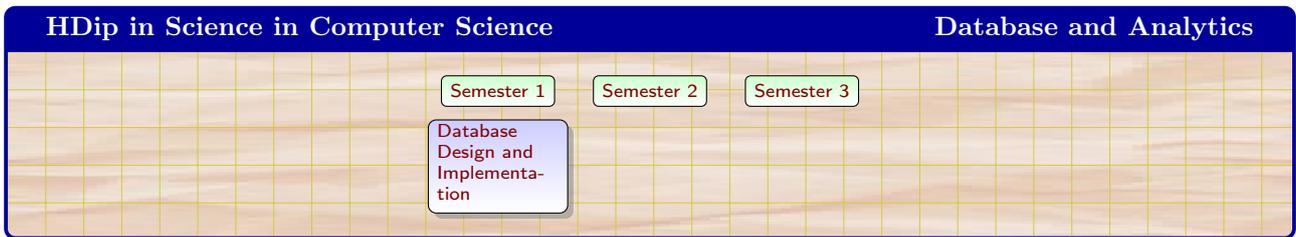
| WD_KCOSC_G / Semester 2 | | Level | Status | Credits | Contact Hours | | | | | Assessment (%) | | |
|-------------------------|--------------------------------------|-------|--------|---------|---------------|-----|-------|-------|----|----------------|------------|--|
| Module Title | Lect | | | | Prac | Tut | Other | Total | CA | Proj | Final Exam | |
| A11302 COMP-0559 | Developer Operations | Adv | M | 5 | 12 | 36 | | | 48 | 100 | | |
| A14173 COMP-0521 | ICT Skills Studio | Inter | M | 5 | 24 | 24 | | | 48 | 100 | | |
| A13744 COMP-0525 | 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 | 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 | 7 |
| Forensics and Security | 10 |
| Networks and Cloud | 14 |
| Professional Skills | 21 |
| Software and Web Development | 24 |

Database and Analytics



Database Design and Implementation 8

Database Design and Implementation (A13705)

Short Title: DB Design and Implementation

Department: Computing and Mathematics

Credits: 5

Level: Advanced

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 Business Systems Analysis (WD_KBUSY_G) | 4 / 1 / M |
| COMP-0079 Higher Diploma in Science in Computer Science (WD_KCOSC_G) | 4 / 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.

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.

Learning Modes

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

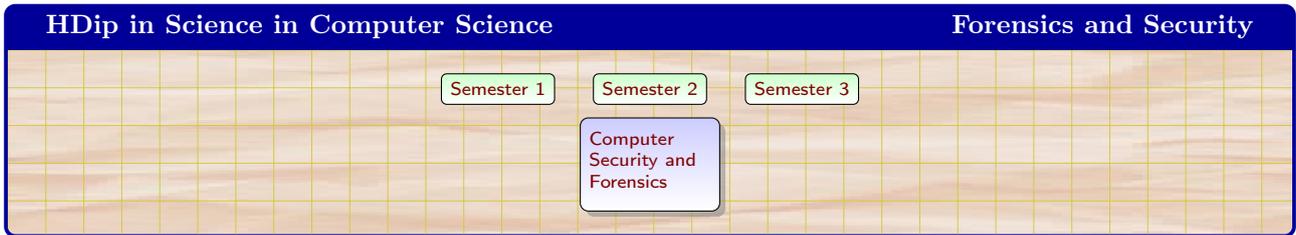
Supplementary Material(s)

- 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 Persistence*. NY: Addison-Wesley Professional, 2012.

Requested Resources

- Room Type: Computer Lab

Forensics and Security



Computer Security and Forensics 11

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.

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.

Learning Modes

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

Essential Material(s)

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

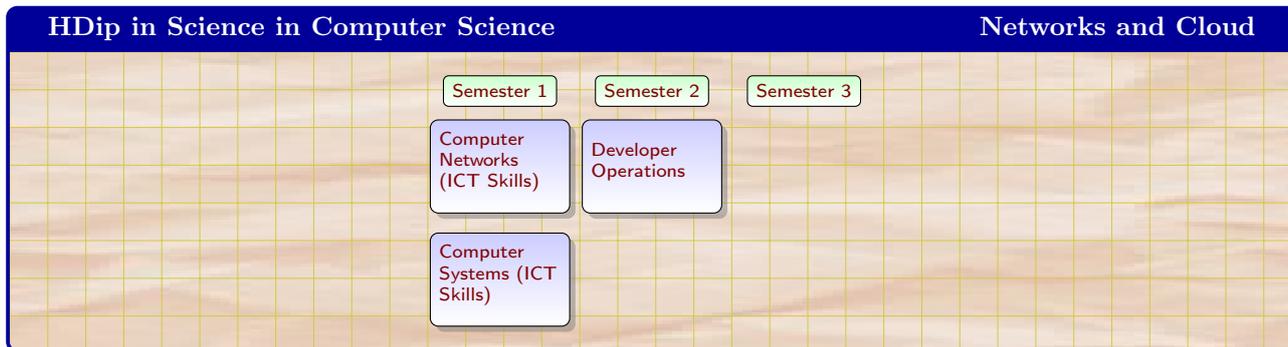
Supplementary Material(s)

- "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



| | |
|--------------------------------------|----|
| Computer Networks (ICT Skills) | 15 |
| Computer Systems (ICT Skills) | 17 |
| Developer Operations | 19 |

Computer Networks (ICT Skills) (A13487)

Short Title: Computer Networks (ICT Skills)

Department: Computing and Mathematics

Credits: 5

Level: Advanced

Description of Module / Aims

This module introduces data communications terminology and concepts, network protocols and models. Students will use protocol analysis software to explore various network protocol operations. An examination of TCP/IP, IP addressing and Ethernet is presented as well as a brief introduction to Routing and Wireless LANs. Practical skills are an essential part of this module.

Programmes

| | stage/semester/status |
|--|-----------------------|
| COMP-0519 Higher Diploma in Science in Computer Science (WD_KCOSC_G) | 4 / 1 / M |

Indicative Content

- Introduction to Computer Networks and Protocols
- OSI and TCP/IP models
- Ethernet
- Network Layer Protocols and Functionality
- IP Addressing and subnetting
- Routing
- Transport Layer Protocols and Functionality
- Application Layer Protocols and Functionality e.g. HTTP, FTP, DNS, SMTP
- Wireless LANs

Learning Outcomes

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

1. Compare network protocol models to explain the layers of communications in data networks.
2. Evaluate the major components, operation and functionality of a computer network and commonly used protocols and services.
3. Design, calculate and apply subnet masks and addresses.
4. Build a simple Ethernet network using routers and switches.
5. Configure routers and switches using command line interface.
6. Determine the operations and features of network protocols and services using protocol inspection software.
7. Implement a basic wireless network.

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.
- The continuous assessment will require the student to apply the theory and practical knowledge to a business solution.

Assessment Methods

| | Weighting | Outcomes Assessed |
|---------------------------|-----------|-------------------|
| Final Written Examination | 50% | 1,2,3 |
| Continuous Assessment | 50% | |
| In-Class Assessment | 50% | 3,4,5,6,7 |

Assessment Criteria

<40%: Unable to describe the major functions and operation of a Computer Network. Unable to describe and compare the OSI and TCP/IP models. Poor understanding of role of communications protocols in computer networks.

40%–49%: Can describe and compare the OSI and TCP/IP models. Can provide overview of main computer network components and protocols.

50%–59%: All of the above. Can describe in detail the data encapsulation process. Demonstrate an understanding of basic LAN implementation.

60%–69%: In addition, be able to recommend a network solution given an organisations' requirements.

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

Learning Modes

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

Supplementary Material(s)

- "Association for Computing Machinery." <http://www.acm.org>
- "Cisco." <http://www.cisco.com/web/learning/netacad/index.html>
- "IEEE Communications Society." <http://www.comsoc.org>
- "IEEE Computer Society." <http://www.computer.org>
- Dye, M., R. McDonald and A. Rufi. *Network Fundamentals: CCNA Exploration Companion Guide (Cisco Networking Academy)*. New York: Cisco Press, 2011.
- Tanenbaum, A. *Computer Networks*. 4th Ed.. New York: Prentice Hall, 2002.

Requested Resources

- Room Type: Computer Lab

Computer Systems (ICT Skills) (A13622)

Short Title: Computer Systems (ICT Skills)

Department: Computing and Mathematics

Credits: 5

Level: Advanced

Description of Module / Aims

The focus of this module will be on five components: computer architecture, memory management, process management, file system management and virtualisation. The student will be given hands-on experience in installing and configuring contemporary operating systems and application services with an emphasis on the Linux environment, including use of the command line interface and writing scripts.

Programmes

| | | stage/semester/status |
|-----------|--|-----------------------|
| COMP-0518 | Higher Diploma in Science in Computer Science (WD_KCOSC_G) | 4 / 1 / M |

Indicative Content

- Translation and calculation in the following number bases: Binary, octal, decimal, hexadecimal. Mod arithmetic
- Boolean Algebra: Basic logic & truth tables
- Computer System Architecture – CPU, memory, storage
- Data Representation
- Operating system structure: Components, services and utilities
- Process Management: Data structures, concurrency, threads, scheduling, synchronization
- File management: File-system utilities, file system implementation, case studies
- Memory management: Paging, segmentation, virtual memory
- Scripting
- Virtualisation

Learning Outcomes

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

1. Calculate operations in binary, octal, decimal and hexadecimal number bases.
2. Apply the basics of Boolean Logic.
3. Describe how the components of a computer system operate together.
4. Demonstrate the relationship between high-level software, low-level programming and hardware.
5. Evaluate the memory, process and file management components of a modern operating system with regard to: Operation principles, data structure requirements and algorithms used.
6. Explain the concepts and theory of virtualisation and in particular how this relates to operating systems management and development.
7. Setup and configure some contemporary operating systems (within a virtual PC environment), and configure the services necessary to support basic applications.
8. Demonstrate competency with a limited set of the utilities (e.g. file management) provided by a contemporary operating system.

Learning and Teaching Methods

- This module will be delivered using a combination of lectures and practical laboratory work.
- The lectures will be used to introduce new topics and their related concepts.
- The practicals will focus on developing the practical skills of the student using simulation exercises.

Assessment Methods

| | Weighting | Outcomes Assessed |
|---------------------------|-----------|-------------------|
| Final Written Examination | 50% | 1,2,3,5,6 |
| Continuous Assessment | 50% | |
| Practical | 25% | 4,7,8 |
| Practical | 25% | 4,7,8 |

Assessment Criteria

<40%: Cannot represent the conceptual design of the system components presented in class or explain operation principles at a basic level.

40%–49%: 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.

50%–59%: 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.

60%–69%: Can demonstrate a comprehensive understanding of the material covered in the lectures.

70%–100%: Excellent understanding of the presented material and displays value added knowledge as a result of independent learning.

Learning Modes

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

Supplementary Material(s)

- "Association for Computing Machinery." <http://www.acm.org>
- "Institute of Electrical and Electronics Engineers." <http://www.ieee.org>
- Dale, N. and J. Lewis. *Computer Science Illuminated*. 6th ed. MA. USA: Jones & Bartlett Learning, 2016.
- Garrido, J.M., R. Schlesinger and K. Hoganson. *Principles of Modern Operating Systems*. 2nd ed. MA. USA: Jones & Bartlett Learning, 2013.
- Silberschatz, A., P.B. Galvin and G. Gagne. *Operating System Concepts with Java*. 8th ed. NJ. USA: John Wiley & Sons, 2010.
- Stallings, W. *Operating Systems: Internals and Design Principles*. 8th ed. NY. USA: Pearson, 2014.

Requested Resources

- Computer Lab: BYOD Lab

Developer Operations (A11302)

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

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 Computer Forensics and Security (WD_KCOFO_B) | 3 / 5 / M |
| COMP-0559 | BSc (Hons) in Applied Computing (WD_KCOMP_B) | 3 / 5 / M |
| COMP-0559 | Higher Diploma in Science in Computer Science (WD_KCOSC_G) | 4 / 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.

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.

Learning Modes

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

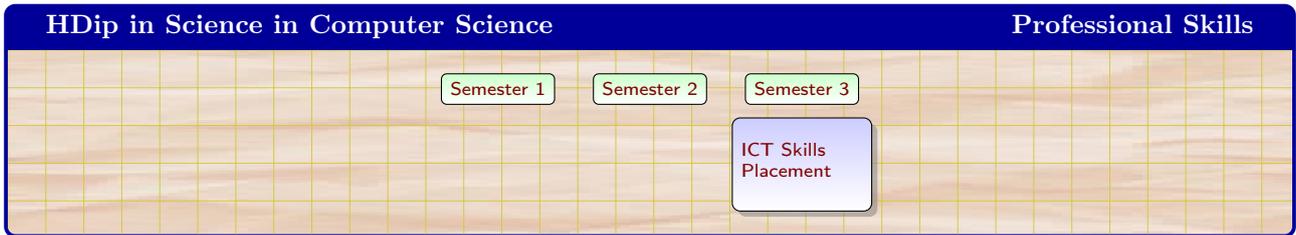
Supplementary Material(s)

- "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



ICT Skills Placement 22

ICT Skills Placement (A14887)

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

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 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) 4 / 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.

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.

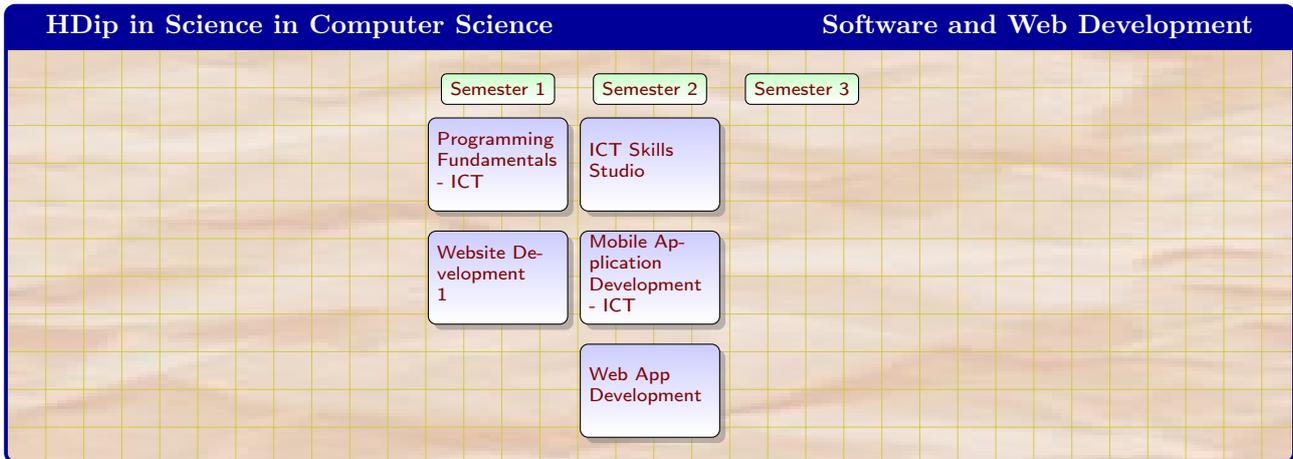
Learning Modes

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

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



| | |
|--|----|
| ICT Skills Studio | 25 |
| Mobile Application Development - ICT | 27 |
| Programming Fundamentals - ICT | 30 |
| Web App Development | 32 |
| Website Development 1 | 35 |

ICT Skills Studio (A14173)

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

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) | 4 / 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 development
- 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.

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.

Learning Modes

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

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

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) | 4 / 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.

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 a 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 debug 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).

Learning Modes

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

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'Reilly, 2015.
- Phillips, B. *Android Programming: The Big Nerd Ranch Guide*. New York: Pearson, 2015.

Requested Resources

- Computer Lab: BYOD Lab

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.

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.

Learning Modes

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

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

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 |
|---|
| Higher Diploma in Science in Computer Science (WD_KCOSC_G) 4 / 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.

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.

Learning Modes

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

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

Website Development 1 (A13422)

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

Description of Module / Aims

This module introduces the student to the creation and development of web based applications. The student will become skilled in HyperText Markup Language(HTML) and Cascading Style Sheets(CSS), to enable the creation of a well structured aesthetically pleasing static website, while meeting accessibility compliance standards. The student will be able to enhance the layout of websites, using a CSS Framework.

Programmes

| | stage/semester/status |
|--|-----------------------|
| Diploma in Computing with Security and Forensics (WD_BCSEC_SP) | 3 / 1 / M |
| BSc (Hons) in Computer Forensics and Security (WD_KCOFO_B) | 1 / 1 / M |
| BSc in Software Systems Development (WD_KCOMC_D) | 1 / 1 / M |
| BSc (Hons) in Applied Computing (WD_KCOMP_B) | 1 / 1 / M |
| Higher Diploma in Science in Computer Science (WD_KCOSC_G) | 4 / 1 / M |
| BSc (Hons) in Entertainment Systems (WD_KENTS_B) | 1 / 1 / M |
| BSc in Information Technology (WD_KINFT_D) | 1 / 1 / M |
| BSc (Hons) in the Internet of Things (WD_KINTT_B) | 1 / 1 / M |
| BSc in Multimedia Applications Development (WD_KMULA_D) | 1 / 2 / M |
| BSc (Hons) in Creative Computing (WD_SR16CC_1) | 1 / 2 / M |
| BSc in Computing (ACCS) (WD_SR16IT_1) | 1 / 1 / M |
| BSc (Hons) in Software Systems Development (WD_SR16SD_1) | 1 / 1 / M |
| BSc (Hons) in Software Systems Development (International) (WD_SR16SD_2) | 1 / 1 / M |

Indicative Content

- Basic Document Construction(HTML features)
- Styling with CSS
- Use of a CSS Framework
- User Experience(UX) and website design
- Templating
- Deployment of a static website

Learning Outcomes

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

1. Demonstrate the ability to create a well-structured static website using HTML.
2. Demonstrate the ability to present and control the format of web pages using CSS.
3. Develop a standards compliant accessible website.
4. Demonstrate the ability to enhance the layout of a website using a CSS Framework.
5. Deploy an aesthetically pleasing website.

Learning and Teaching Methods

- The module will be delivered in one lecture and in three hours of computer-based practicals each week.

Assessment Methods

| | Weighting | Outcomes Assessed |
|-----------------------|-----------|-------------------|
| Continuous Assessment | 100% | |
| In-Class Assessment | 30% | 1,2 |
| Portfolio | 70% | 1,2,3,4,5 |

Assessment Criteria

<40%: Unable to interpret and describe key concepts of website development.

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

50%–59%: Ability to discuss key concepts of website development and ability to discover and integrate related knowledge in other knowledge domains.

60%–69%: Be able to solve problems within website development by experimenting with the appropriate skills and tools.

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

Learning Modes

| Learning Type | F/T Hours | P/T Hours |
|----------------------|-----------|-----------|
| Lecture | 12 | 6 |
| Practical | 36 | 18 |
| Independent Learning | 87 | 111 |

Essential Material(s)

- "W3Schools." <http://www.w3schools.com/>

Supplementary Material(s)

- Castro, E. and B. Hyslop. *HTML and CSS: Visual Quickstart*. 8th ed. Berkeley: Peachpit Press, 2014.
- Felke-Morris, T. *Web Development and Design Foundations with HTML5*. 7th ed. London: Pearson, 2015.

Requested Resources

- Computer Lab: BYOD Lab