Anglia Ruskin University

Course Information Sheet

BSc (Hons) Artifical Intelligence

Mode and course length – Full-Time (4 years)

Location – ARU Cambridge Campus

Awarding Body – Anglia Ruskin University. As a registered Higher Education provider Anglia Ruskin University is regulated by the Office for Students.

Overview

Explore the subject of Artificial Intelligence and learn to employ AI methodologies in order to define problems, design solutions and develop AI-based software tools for enhanced technologies and decision making across a large number of areas.

Artificial Intelligence, the computer emulation of human thinking, has been identified by the UK Government as the first of its four Grand Challenges to keep the UK at the forefront of technological innovation and industrial development.

AI is an ever-evolving subject area that requires you to remain aware of constant changes and developments, as well as current issues that AI faces.

Our brand new degree course has been designed with employers and industry in mind, ensuring you are learning the professional skills you will need once you graduate. You will be provided with the knowledge and skills to develop as a specialist in the areas of data analysis, machine learning, neural networks and artificial intelligence algorithm design. You'll also have access to Intel's AI Academy, an online data processing resource. You will learn the history of Artificial Intelligence, systems needed to underpin AI and the advanced algorithms it employs, using up-to-date technology including software packages and hardware infrastructure.

Through this course you will gain the knowledge needed to display good professional judgement and responsibility in addressing commercial, environmental and ethical factors found in the various areas where AI is used.

If you enjoy mathematics and want to apply these fundamentals in a practical way – combining them with programming, computer systems and software design to create practical solutions to everyday technological issues – Artificial Intelligence is the subject for you.

Course Delivery

Our courses are delivered through teaching and learning methods which provide students with the widest possible exposure to a modern and innovative higher education experience.

These methods vary and could include attendance at lectures and seminars, undertaking laboratory exercises or work-based activities, practical work, performances, presentations, field trips, other relevant visits and e-learning through Canvas, our online learning management system.

Each course is divided into a number of 'modules' which focus on particular areas, each of which has a specific approach to its delivery. This information is published to students for each module they take via the Module Definition Form (MDF) and Canvas.

Assessment

Throughout the course, we'll use a range of assessment methods to measure your progress. You'll complete exams, essays, research reports, oral presentations, and a dissertation on a subject of your choice.

Fees

Information about your course fee including any annual fee increases or deposits (if required) can be found in your offer letter.

Modules

Core Modules

Year 1: Foundation in Engineering, Computing and Technology

This module will provide students with the necessary skills to begin studying at level 4 in Engineering, Computer Science and related courses.

Students will be introduced to the core skills necessary to succeed in higher education, including thinking critically, researching and referencing appropriately, demonstrating appropriate numeracy and ICT skills, and communicating effectively verbally and in writing.

In addition to these fundamental skills, Students will cover the subjects underpinning the technological disciplines. Fundamental mathematical skills will be covered, alongside pre-calculus, followed by an introduction to calculus and vector and matrix arithmetic. Students will also be introduced to Classical mechanics, and its application to real-world scenarios. Students will be introduced to the fundamentals of computer science, learning about the principles behind programming and applying them through a series of practical coding exercises. Students will undertake a multi-disciplinary group project as they learn about the collaborative nature of engineering, and design from a broader perspective of business.

The module is made up of the following 8 constituent elements:

- Interactive Learning Skills and Communication (ILSC)
- Information Communication Technology (ICT)
- Critical Thinking
- Maths for Scientists
- Maths for Engineers
- Physics for Engineers
- Fundamentals of Computing
- Engineering Design

Year 2: Introduction to Programming

This module provides an introduction to high level programming, requiring no prior programming experience. The student will use industry-standard tools and techniques to design, implement, test and document simple programs using a current programming language such as C#, Java or C++.

The module will enable students to understand the principal components of a high-level program, laying the foundation for subsequent modules requiring structured programming ability. It will emphasise the principles of good programming practice and introduce the techniques required to develop software which:

is robust and efficient;

satisfies the needs of the customer;

has a usable and aesthetic interface;

consists of elegant, easy to read code;

is resilient within the cybersecurity context.

Summative assessment will address the student’s knowledge of programming theory, syntax and best practice. Formative exercises will be set at intervals through the module for peer review and feedback. By the end of the module, students should have sufficient mastery of a high-level programming language to allow them to design, implement and test simple programs. The skills taught within the module are intended to be directly transferable to the workplace and to provide a suitable foundation for students who will be expected to apply programming skills in their later studies and future career.

Year 2: Computer Systems

This module introduces students to the components present in modern computer systems and networks. On completing this module, students will be able to specify, construct and maintain networked computer systems, and gain an in depth understanding of common computer and network architectures, their function and confidently solve their problems. Theoretical topics include CPU architectures; bus systems; types of memory and data; program execution; number systems; peripheral and network architectures, models and protocols; the importance of standards; network devices and cabling. The module will be delivered in weekly lectures, which should be consolidated through significant self-study. Laboratory sessions will enable students to gain the practical skills needed to construct, maintain and solve problems on networked computer systems. Underpinning the theory students will learn (among other practical skills) how to safely assemble PCs (taking account of health and safety issues); construct (and test) network cables; examine low-level Internet traffic and set up and configure local area networks.

Year 2: Computer Modelling and Simulation

This module introduces the use of computer tools to solve engineering problems. It is intended to provide a sound understanding of the principles of generating a computer model, simulation or solution from a defined specification. The MATLAB software package is introduced, which allows mathematical expressions and algorithms to be implemented using various command functions and simple software statements. Basic ideas of producing plots are presented; relevant toolboxes are also discussed, e.g. the Statistics and Machine Learning Toolbox. Students are also introduced to the basics of Python language. Fundamental issues like variables, strings, tuples, loops, control flow, plotting, file input/output, functions will be discussed. Students will learn the basic features of NumPy (Numerical Python) and will create data visualisations with Matplotlib. This module will also introduce the Pandas library for data analysis. Finally, students will learn how to solve problems by incorporating all the elements reviewed in the module and apply them in different scenarios. The emphasis in learning computer languages will be on a structured approach. The applications targeted will be in the area of modelling and solving technological problems relevant to students' course.

Year 2: Fundamentals of Artificial Intelligence

The UK Government has identified ‘Artificial Intelligence and Data’ as first of its four Grand Challenges in order to keep the UK at the forefront of technological innovation and industrial development. Artificial Intelligence can be defined as the computer emulation of the human thinking process.

This module provides the foundations of understanding Artificial Intelligence (AI), its historical development, main application directions and how it has become one of today’s essential technologies.

The module will explore the history of AI, essential concepts like supervised/unsupervised learning, neural networks, fuzzy logic, as well as the enabling role that large scale data collection and fast computing have in establishing AI as a main tool for development and automation in a growing number of industries. This module will also provide a fundamental understanding of key concepts of Artificial Neural Networks (ANN) and Fuzzy Logic (FL), as core elements of Artificial Intelligence (AI). ANNs are information processing structures that emulate the architecture and operation mode of the biological nervous tissue. In ANN, several basic entities named ‘neurons’ are interconnected and operate in parallel, transmitting signals to each other in order to perform a certain task. They form networks which have the special capability to simulate the learning process as an outstanding feature. Fuzzy Logic extends the principles of Boolean logic to handle the ‘partial true’ truth values between ‘completely true’ and ‘completely false’ truth values. Fuzzy Logic has emerged as an effective tool to control complex industrial processes and systems for which an exact mathematical model cannot be easily developed. The importance of data to both training of neural networks and the steps in a data science workflow will also be emphasised. Current hardware and software packages will also be introduced.

Students undertaking this module will develop essential skills related to the understanding of some of the core principles of artificial intelligence alongside their key applications into the real world. Such skills are very much sought by employers as they underpin a wide range of modern industrial developments with a forecast for rapid growth.

Year 2: Introduction to Mathematical Techniques for Artificial Intelligence

Artificial intelligence aims to discover, analyse and classify patterns in data sets in a manner that lends itself to optimised automated or even autonomous operation. It is therefore a field that uses techniques that deal with multiple numerical features of data, the relative variation of these features, as well as the variability of the data that may determine a pattern. This module is essential for the student who needs a solid background in mathematical techniques and analysis in order to pursue a degree

programme in artificial intelligence. The module will help students to assess their existing numerical and analytical skills and enable them to expand these into core mathematical skills, knowledge and techniques needed in order to tackle scientific and engineering problems in the field of pattern analysis and machine intelligence.
The module will provide an introduction to linear algebra, with an emphasis on the practical use of vectors, matrices and linear systems as representations of multi-dimensional data and processes. The techniques of calculus, differentiation and integration, are also discussed, with application to optimisation problems. Fundamental methods to obtain numerical solutions to analytical problems, as well as an introduction to modelling numerical data are also included. The module will consider applications of the fundamental mathematical concepts into the context of artificial intelligence.

Year 3: Advanced Analytical Techniques for Artificial Intelligence

Building on the mathematical methods acquired in Introduction to mathematical methods for artificial intelligence, this module provides further mathematical knowledge and techniques for the study of pattern analysis and machine intelligence. Substantial parts of the machine learning discipline are concerned with analysing large amounts of data in order to discover and label similarities between data points. These inferences are made based on probabilistic modelling and statistical measures of reliability. This module will develop the basic theory of probability, covering discrete and continuous random variables and problems like conditioning and independence, Bayesian inference and applications. It will equip students with knowledge and skills that will allow them to design simple probability models for prediction, to make basic statistical analyses of data, and to assess and interpret such analyses. The module will also expand the knowledge on linear algebra acquired at level 4 with topics like decompositions of symmetric or arbitrary matrices, sparse computing and applications to machine learning. The interconnection between linear-algebraic methods and statistical data analysis will be explored.

Year 3: Database Design and Implementation

This module guides students through the fundamentals of database design using the software lifecycle as a basis. This grounding enables students to construct industrial quality databases. Students work in small groups emulating real world development teams. They learn the skills of constructing excellent documentation, working in draft, making revisions and delivering work to a deadline. Implicitly they learn the skill of managing a group environment. The module begins with the development of an acceptable approach to industrial clients and their problems. Working within the specification given by an Enterprise, the group learns how to extract data from interviews and documents. Students progress to designing and building a database, querying the database to provide the reports (including statistics) that the Enterprise needs. During this process the current industrial choice database language (SQL) is learned. Specialist resources include availability of Microsoft AccessTM, MySQL, or similar applications.

Year 3: Software Engineering

Ian Sommerville states that Software Engineering is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software. The aim of this module is to give to students a real-world experience in software engineering. The number, size, and application domains of computer applications have grown to the extent that many organisations and individuals depend on the effectiveness of software development. Therefore software products have to be efficient, of very good quality and to help us to be more efficient and productive. This module will provide students with the intellectual and practical tools required to be able to design, implement and test software systems. The module is built on material covered in Fundamentals of Design and Introduction to Programming, and will cover all the phases of the life cycle by taking case studies and building real software applications based on them. As part of this at least two types of project management, waterfall and agile methodologies will be discussed together with the cost drivers that can influence projects. The students will use computer-aided software engineering (CASE) tools to study topics including analysis and design in UML and managing the OO software development process. Both the automated and manual testing are discussed and the students have to demonstrate the ability to use both of them.

Year 3: Digital Security

'Digital Security' is essentially about giving individuals the freedom to embrace the digital lifestyle, confidently engaging in everyday interactions across all digital devices with a certainty that the accessibility and integrity of the data is ensured. Digital security affects all aspects of the digital lifestyle, which, among others, comprises computers and the internet, telecommunications, financial transactions, transportation, healthcare, secure access and the IT professional's legal and moral obligations when practicing their trade. This module essentially covers these broad topic areas: - Computer Security Principles covers security objectives such as authentication, authorization, access control, confidentiality, data integrity, and non-repudiation. The module also introduces fundamental software design principles such as that of least privilege, fail-safe stance, and defense-in-depth. - Introduction to Cryptography covers both symmetric encryption and public-key cryptography, discussing how they are used to achieve security goals and build PKI (Public-Key Infrastructure) systems. Technologies covered are typically DES, 3DES, AES, RC4, RSA, ECC, MD5, SHA-1, X.509, digital signatures, and all cryptographic primitives necessary to understand PKI. Diffie-Hellman key exchange and man-in-the-middle attacks will also be discussed. - Secure Programming Techniques discusses the threats that worms and hackers present to software and the programming techniques that developers can use to defend against software vulnerabilities such as buffer overflows, SQL injection, and off-line dictionary attacks.

Year 3: Machine Learning

Machine learning is a form of Artificial Intelligence that enables a system to learn from data rather than through explicit programming. Machine learning has become one of the most important topics within development organisations that are looking for innovative ways to leverage data assets to help the business gain a new level of understanding.
In this course, students will learn about the most effective machine learning techniques, and gain practice implementing them and getting them to work. Students not only learn the theoretical underpinnings of learning, but also gain the practical know-how needed to quickly and powerfully apply these techniques to new problems.
By the end of this course, students will have practical knowledge of:
 Supervised learning algorithms
 Key concepts like under- and over-fitting, regularization, and cross-validation
 How to identify the type of problem to be solved, choose the right algorithm, tune parameters, and validate a model

Year 3: Artificial Neural Networks

This module will equip students with detailed understanding of the core principles and applications of artificial neural networks (ANNs). The biological basis upon which neural networks are loosely based will be briefly summarised (including evaluating the key differences between biological and artificial neural networks), and the historical development of ANNs will be reviewed. The strengths/weaknesses of the ANN approach in comparison to other artificial intelligence (AI) methods will be discussed, and example problem types for which ANNs are suitable will be outlined (such as pattern recognition, classification, medical diagnosis, financial/economic data analysis and prediction, and gaming). The fundamental graph-based data

structures and algorithms used in neural network architectures will be presented, and the basic topologies of different ANNs compared. A range of alternative neural network architectures will be presented, such as feedforward (including perceptron, autoencoder, and convolutional networks), radial basis function networks, Hopfield networks, Boltzmann machines, recurrent and spiking neural networks. Attention will be drawn to the advantages/disadvantages of each approach in specific problem domains. The merits of supervised and
unsupervised learning, different activation functions, learning rules, and training approaches will be discussed. The approaches whereby a trained ANN is evaluated for overfitting will be presented, including training/testing partitions, k-folds cross validation, and leave-one-out validation. In weekly practical sessions, students will construct ANNs to solve computational problems that would be difficult to solve using conventional computing methods. A suitable high-level language and ANN functions from standard libraries will be used, such as TensorFlow, Caffe, Theano, and Keras with Python or MATLAB with the Neural Network Toolbox.

Year 4: Final Project

The individual Final Project module allows students to engage in a substantial piece of individual research and / or product development work, focused on a topic relevant to their specific discipline. The topic may be drawn from a variety of sources including: Anglia Ruskin research groups, previous / current work experience, the company in which they are currently employed, an Anglia Ruskin lecturer suggested topic or a professional subject of their specific interest (if suitable supervision is available). The project topic will be assessed for suitability to ensure sufficient academic challenge and satisfactory supervision by an academic member of staff. The chosen topic will require the student to identify / formulate problems and issues, conduct literature reviews, evaluate information, investigate and adopt suitable development methodolgies, determine solutions, develop hardware, software and/or media artefacts as appropriate, process data, critically appraise and present their finding using a variety of media. Regular meetings with the project supervisor should take place, so that the project is closely monitored and steered in the right direction. The project developed in this module is the most substantial piece of work that the student is producing during their undergraduate studies. Thus, the choice of project topic and the quality of the work is likely to bear a great influence on the student's career / employability. Therefore, the module will also include aspects of Personal Development Plan and CV preparation. The students are strongly advised to allocate appropriate attention, time and effort to this module. The successful completion of the module will increase students' employability, as they will acquire skills directly applicable to real world projects.

Year 4: AI Techniques (Fuzzy Logic and Genetic Algorithms)

In artificial intelligence, an expert system is a computer system that emulates the decision-making ability of a human expert. Students are first introduced to the classic software architecture of an expert system as first developed in the 1980s, comprising a knowledge base and an inference engine and some well-known examples are described (e.g. Prospector, Mycin, Deep Blue). The theoretical basis for such rule-based systems is presented through covering some basic principles of propositional and predicate logic. Students will be expected to have a good working basic knowledge of one high level programming language (such as C#, Java, C++, Python) as they will be expected to be able to edit and configure small-scale source code examples of rule-based systems implemented in a high-level general purpose computer language. Students will also be introduced to a specialist logic computer programming language (e.g. Prolog, Lisp).

Fundamental to all of this will be an appreciation of a kind of learning where new knowledge is derived from conditional facts. Modifications to the 'crisp' rule-based approach will then be outlined including the principles of fuzzification and de-fuzzification and how this manifests itself in a more accurate and customized output response. The use of genetic algorithms in expert systems follows a recent general trend to combine different methods of machine learning to optimise a solution. The principles of genetic algorithms will be reviewed, including how the biology of genes, chromosomes and reproduction can be represented in a computational context. Issues of problem representation through binary expression of those problems are covered, and students will edit pre-existing source code examples to evaluate and understand reproductive strategies and assess issues of accuracy and error minimisation. The course culminates in seeing how the two approaches of rule-based inferred learning and learning through utilisation of a genetic algorithm can be combined, principally by demonstration of some case examples drawn from computer gaming, the Internet of Things and in cybersecurity. This 15-credit module is quite practical with an emphasis on interactivity in terms of code development, and assessed entirely by coursework.

Year 4: Professional Issues: Computing and Society

It is essential to ensure that a professional engineer has an in depth understanding of professional ethics, law and the impact of what they do on society. This module aims to provide an understanding of the issues, opportunities and problems which have arisen as a result of the computerisation of wide areas of human activity. It is designed to enhance advanced computer reflective thinking in both computer science specialists and others, and is a key part of the programme of professional development for computer scientists and others seeking to embody professional values and approaches in the IT and computing fields. The course covers relevant and current topics in Computer Law (e.g. Data Protection; Intellectual Property Law; Computer Misuse) and other social, ethical and legal topics such as considering the causes and effects of systems failures (including but not limited to computer systems failure). Other aspects such as the ethical and professional responsibilities of graduates - particularly those from IT and Computing disciplines - will be critically appraised. Topics may also cover the technical development and social effects of computer technology from c1936 to the present day, as the basis for an informed discussion of the issues and whether lessons have been learnt from the past. A high level of in class student participation is expected; non attendance/non-contribution may be penalised.

Year 4: Deep Learning

This module will equip students with detailed understanding of the core principles and applications of Deep Learning. Neural network principles upon which deep learning is based will be revised (including what is a neuron and its similarity to a biological neuron). Basic architecture of a feed-forward neural network, activation functions and weights will be reviewed.

Fundamental neural network architectures like feed-forward networks, Boltzmann machines, convolutional neural networks (CNN), and recurrent neural networks (RNN) will be introduced. The concept of how a neural network computes the output given an input in a single forward pass, and how to use this network to train a model will be covered. Students will learn how to calculate the loss and adjust weights using a technique called backpropagation. Different types of activation functions will also be introduced, as well as techniques to use to improve training speed, accuracy, and to prevent overfitting (regularization). The method of how to build a CNN as well as techniques, terminology, mathematics of deep learning will be discussed. How to appropriately build and train neural network models will be examined and compared. In addition, how classification topologies (such as AlexNet, VGG-16 and VGG-19, Inception, and ResNet) are designed and the usage scenarios for each will explored. A range of various suitable deep learning applications (e.g. recurrent neural networks (RNN) and their application to natural language processing (NLP)) will be presented and reviewed in detail. Furthermore how to use pre-trained models for best results will be outlined and discussed (e.g. how to make the most of the available labelled data using data augmentation).

Using a blend of theoretical discussion, laboratory sessions and remote access to class servers, this module will cover the necessary skills to understand and evaluate, and to appropriately build and train models in various deep learning applications to achieve best results.

The module is delivered by a mixture of classroom-based lectures and practical sessions. During and outside of scheduled class times students have remote access to a class server where they will be able to access notes, participate in discussions, store their documents and experiment with some of the class material.

Year 4: Distributed Programming

Distributed Programming is the development of software applications that utilise the distributed functionality of an intranet or the internet. These applications are vital to the banking sector, commercial organisations and governmental institutions as they involve the fundamental technologies underpinning Cloud Computing and On-Line Multi-Player Gaming environments.

The module covers the key principles of low-level distributed programming to manage the communication of data between computers. The language of implementation will be one whose libraries support Socket programming, such as Java, C# or C++. Students will learn how to develop applications that share out, or 'farm' large computing operations to smaller interconnected nodes thus implementing a kind of virtual parallel processing.

A variety of practical exercises will illustrate these programming techniques and components in an Intranet environment. Examples of programming language support for some of the more common application and communication network protocols will be covered. Threads and multi-threading is introduced as a technique to manage concurrency and the marshalling of data between processes.

Optional Modules

(Subject to availability)

Year 3: Object Oriented C++

C++ (and its language precursor, C) is arguably the most common programming language in industry, and graduates who are good C/C++ programmers are often much sought after in the IT sector (systems programming, embedded software, graphics and games programming). The reason for the popularity of C++ is partly historical, partly because the programmer can produce fast, memory-efficient programs, and partly because of its flexibility to support different programming styles. This module provides an introduction to C++ for those already with some programming experience in another language such as Java or C#. Following a procedural introduction students will learn an object-oriented style of programming including some design considerations. Code will be written using an appropriate development environment (such as VisualC++, DevC++, or C++Builder) and mainly confined to ANSI/ISO C++ and use of the standard library so as to promote source code portability to other platforms. Students will learn how explicit types of memory allocation can be used to manipulate data and how this can influence computer resources, and thus will gain an understanding of the underlying architecture behind how other high level programming languages manage their data.

Year 3: Microprocessor Systems Design

In this module the student will develop an in-depth understanding of microprocessor system and its relation to the design of modern digital systems. Hands-on programming and simulation of the operation of a commercial microprocessor will be an important part of this module. The module covers different microprocessor architectures, and core elements like ALU, CU, BIU, memories, caches, pipelines, superscalar architectures, RISC and CISC. Real time and non real time hardware and software requirements for embedded microcontroller systems are presented and the relationship between system performance and hardware and software interface is covered. The module delivery strategy combines complex theoretical aspects and case studies presented during lectures, with practical skills - hands on supervised and unsupervised laboratory work, using state-of-the art industry standard CAD tools. Students are encouraged to take responsibility for their assignments and to work in their own time as well as during the timetabled classes. The successful completion of this module will increase students' employability, who will acquire industry standard skills, directly applicable to real world projects.

Year 4: Image Processing

Image processing is a rapidly growing area of computer science, with applications as diverse as entertainment, manufacturing/robotics, forensics, security, broadcasting, cognitive psychology, computer graphics, and medicine. The range of underpinning techniques relevant to the image processing discipline is similarly diverse, and include image acquisition and digital representation, visual perception, image statistics, transformation, enhancement, restoration, compression, and higher level analyses such as feature recognition and object tracking. The input to an image processing operation is typically a two dimensional signal (i.e., an image matrix), and the output may be a modified signal (for instance, an enhanced image), or a set of descriptors or interpretations, produced by an image processing operation. This module will introduce students to a set of core image processing operations in weekly lectures in which their theoretical and mathematical foundations will be emphasised. Students will be expected to implement a range of image processing algorithms using real datasets in structured weekly laboratory sessions.

Year 4: Digital Signal Processing

Fundamental to the understanding of digital signal processing is a sound working knowledge of the mathematical principles which underpin the subject. Also a good understanding of the algorithms which are available for implementing digital signal processing techniques which include digital filtering and spectral analysis methods. The module will therefore provide the student with a working maths framework to enable them to understand how digital signal processing techniques can be implemented in commercial digital systems.

Year 4: Internet Services, Data Analytics and the Cloud

The Internet and the emerging cloud-computing paradigm provide an opportunity to design and implement a wide range of effective analytical and distributed applications that can be accessed via various types of devices. The success of such applications involve skilful use of data science and several programming techniques, requiring professionals in the field to confidently deliver solutions in a fast-paced and time-constrained environment. An in-depth knowledge of prototyping, through coding, testing and deployment, is the key to delivering such applications. This module is specifically designed to provide the knowledge and skills to enable students to confidently implement effective analytics applications using technologies that underpin the Internet and the Cloud. Cloud Computing security is also discussed and explored.

A significant proportion of the module will involve writing and testing code using current industry standard programming and scripting languages. Prior coding and programming experience will be assumed: students taking this module are expected to quickly pick up the programming languages introduced. An important part of developing effective cloud/web-based distributed analytics applications is the understanding of current database management systems. Prior knowledge of and experience with simple database design and implementation is therefore a pre-requisite for this module and will be assumed.

Using a blend of theoretical discussion, laboratory sessions and remote access to class servers, this module will cover the necessary skills to understand, evaluate, implement and apply good practice in prototyping effective cloud/web based distributed applications.

Year 4: Ethical Hacking and Countermeasures

The aim of this module is to give students a rounded introduction to the principles of ethical hacking from theoretical and technical perspectives and to provide a contextual setting for ethical hacking by an examination of the issues associated with systems security, computer crime and the criminal justice system i.e. Computer Misuse Act. Students will be introduced to the basic principles of ethical hacking and the role ethical hacking plays in providing more secure and robust information to support computer systems and networks (including wireless networks). Students will be exposed to, and use, the basic tools and techniques of ethical hacking, particularly in regard to penetration testing and systems security. Students will be provided with opportunities to develop academic skills in report writing and reflective practice presentations. By research and application students will develop the skills to manage the particular legal, ethical and professional challenges, facing the Information Security practitioner with particular reference to the criminal justice system in England and Wales and the Computer Misuse Act.