

## **Module Outline**

### **Year 2 Modules**

#### **Human Computer Interaction**

This module introduces the principles and concepts of Human-Computer Interaction (HCI) to enable the design of multimodal systems that effectively meet human needs. The course covers aspects of interaction design and human factors, offering theoretical grounding and practical skills for analysing, designing and evaluating the usability of interactive software systems. Topics covered include accessibility, auditory based interaction, and virtual and augmented reality.

On successful completion of the module, a student should be able to:

- Appreciate the importance and context of HCI and human factors in the software development lifecycle.
- Recognise the importance of identifying and involving users in the design of interactive systems.
- Apply task analysis and dialogue design methods to facilitate effective interaction design.
- Demonstrate awareness of design patterns for effective user interface design.
- Utilise usability guideline and principles in the design and evaluation of interface prototypes.
- Develop knowledge of tools and techniques for inclusive software design.
- Understand how multimodal user interfaces are designed and developed to overcome issues with screen-based interactive systems.
- Select and apply suitable usability evaluation methods for the evaluation of interactive software systems.
- Study essential empirical skills for conducting user studies to evaluate interface prototypes.
- Appreciate concepts and techniques for emerging interaction technologies such as virtual and augmented reality.

#### **Internet of Things**

In this module, students will develop an understanding of how to design and develop an Internet of Things (IoT) system. This module aims to cover the end-to-end IoT systems design and development process which includes aspects such as architecture, sensing and actuation, computation and analytics, human factors and interactions, privacy and security issues. The students will develop an understanding of how different design decisions may lead to different trade-offs. Further, students will also learn practical aspects such as device programming and debugging, cloud integration, and deployments.

On successful completion of the module, a student should be able to:

- Critically dissect and analyse existing IoT systems in terms of the design choices.
- Analyse and apply common design decisions, their trade-offs, and consequences.
- Identify and formulate a problem in the area of IoT.
- Design, develop and deploy an IoT application and justify their design choices.
- Create reproducible projects with well-documented codebases.
- Disseminate project outcomes to a broader audience through digital means.
- Awareness of relevant professional, ethical, legal and social issues that arise in the implementation of existing and future computer systems

### **Database Systems**

This module aims to provide students with a basic understanding of database system technology in general, and a theoretical and practical knowledge of relational database systems in particular. It seeks to equip the students with the skills to design and implement a database system.

On successful completion of the module, a student should be able to:

- Illustrate basic database concepts and systems architecture
- Understand the representational and processing power of the relational DBMS
- Explain the process of and the issues involved in database design and implementation
- Show a basic understanding of database transaction management
- Design a relational database system logically and physically
- Develop relational database queries using Structured Query Language (SQL)

### **Enhancing your Employability**

This module is intended to enhance students' employability by providing them with the skills and competencies needed to secure employment and support ongoing professional development. It provides an opportunity for exploring the concept of professionalism as it applies to computing professionals; the role and importance of continuous professional and personal development; the fundamental skills required from Computer Science students and graduates; and knowledge about and appreciation of relevant legal, professional and ethical concepts and standards that should be appropriately applied throughout their studies and career. In this module, students can gain access and develop their networks with professionals, employers and entrepreneurs, while developing their knowledge linked to social, legal and ethical issues in the modern workplace.

On successful completion of the module, a student should be able to:

- Create a professional presence using a variety of skills, techniques and tools
- Demonstrate an understanding of professional social, ethical and legal requirements relating to employment and technology
- Demonstrate industrial and commercial awareness of the technology sector and the role of technology in other industries
- Develop and maintain a portfolio of work including reflection on their continuing professional development

## **Secure Communication Networks**

The module will introduce the fundamental principles and emerging areas in communication networks, covering the underlying technologies and protocols, implementation, and issues of network security, dependability, and performance.

On successful completion of the module, a student should be able to:

- Evaluate issues involved in deploying communication networks and their potential security, performance, and dependability implications and trade-offs.
- Describe the fundamental principles and protocols of wired and wireless communication networks.
- Demonstrate an understanding of performance and dependability evaluation approaches for secure communication networks.
- Demonstrate an understanding of the principles of secure communication, including network vulnerabilities and security controls.
- Use software tools used to analyse network traffic

## **Group Project**

This module gives students an understanding of a professional approach towards developing integrated computational systems. As group members, students will initially define a problem, followed by identifying requirements, designing, implementing, testing, and evaluating a prototype system to address the problem they've identified. Students will be introduced to a range of development methodologies, covering techniques for gathering requirements, software design, implementation, and testing, including waterfall, iterative, and agile methods. Students will practice tackling a large, challenging project with a team of people, working as a member of a team, planning and managing a team project, completing a project within a set time scale, and producing prototype systems, reports and presentations.

On successful completion of the module, a student should be able to:

- Understand how to apply appropriate requirements gathering approaches to determine appropriate functional and non- functional requirements in a constrained project scenario.
- Understand common approaches towards software development from waterfall to iterative and agile methods, and use agile project management techniques within a project.
- Show awareness of legal, social, commercial, ethical and professional issues in software development.
- Assess benefits, risks and relevant quality factors appropriate to specifying, designing, and implementing a system.
- Use common software development tools aimed at different development approaches and stages.
- Demonstrate an appreciation of the problems involved in tackling a challenging project as part of a team.
- Design a software system to meet given requirements.
- Implement and test a software system based on a design.
- Evaluate how well a software system meets given requirements.

- Reflect on their experience of working in a team and their individual contributions to the project.

### **Object Orientation, Algorithms and Data Structures**

This module aims to teach the principles of good Object-Oriented Analysis and Design (OOAD) practices using design patterns, and techniques for the design and analysis of algorithms using efficient data structures. It focuses on ways in which computing concepts can be realised in an object-oriented fashion, and on the development and application of re-usable code and designs. It introduces the idea of classifying data according to its abstract behaviour, as distinct from its representation. It provides an understanding of the basic skills needed in the design of algorithms, emphasising the interactions between algorithms and data structures in creating efficient code. It introduces the concept of concurrency and multi-threading in object-oriented programs.

On successful completion of the module, a student should be able to:

- Appreciate the main features that are needed in a programming language in order to support the development of reliable, portable software and how key Object Oriented (OO) principles relate to those features.
- Apply principles of good OO software design to the creation of robust, elegant, maintainable code.
- Understand the design of various fundamental data structures and know their advantages and disadvantages.
- Explain the difference between various fundamental algorithmic techniques and employ these for simple algorithm design.
- Understand and analyse the resource requirements of various algorithms and data structures.
- Identify the most appropriate data structure and algorithm to solve a particular problem.
- Explain and apply a range of design patterns. Demonstrate understanding of object-oriented abstractions for concurrency.

### **Data Processing and Visualisation**

The aim of this module is to develop skills needed to process information and provide an understanding of statistical methods to analyse the resulting data. The techniques studied will enable data collection from a range of sources, including files and the web. The module will cover Python modules that can easily manipulate and convert information to extract data. Statistics to describe collections of data will be studied, along with basic methods to derive correlations and test simple hypotheses.

On successful completion of the module, a student should be able to:

- Use Python to extract, manipulate, store and analyse information from a range of sources.
- Understand statistical methods to apply to data.
- Understand static visualisations of data
- Create static visualisations of data

## **Informatics**

The aim of this module is to provide the student with an understanding of the role data mining and data quality techniques play in our lives. The students will develop a basic toolbox allowing them to use methods for learning and evaluation information. This will be paired with a consideration of the ethical implications surrounding gathering and using information in an automated manner.

On successful completion of the module, a student should be able to:

- Execute and evaluate various techniques in knowledge discovery and data mining.
- Analyse and critically evaluate methods for assuring quality of information.
- Appraise the ethical implications and societal risks associated with data mining and data quality assurance.

## **Introduction to the Theory of Computation**

The theory of computing deals with questions such as: what is computation? What problems can be computed? What is the inherent difficulty of a given problem? To answer such questions in a satisfactory manner, we need a formal definition of computation, which does not rely on any specific hardware or programming language. It is perhaps surprising that some real-world problems can be mathematically proven to be undecidable, i.e. regardless of any future advances in hardware, such problems are provably impossible to solve on a computer. In this module we will discuss the main models of computability and provide several examples of undecidable problems. While questions about computability have initially been studied from a theoretical perspective, the resulting theories have had many important practical applications. Among others, the methods discussed in this module are used for implementing compilers (in particular for parsing), for defining functional programming languages, and for formally verifying the correctness of software and hardware.

On successful completion of the module, a student should be able to:

- Use standard methods for proving mathematical properties.
- Understand the basic rules of classical logic.
- Demonstrate an understanding of formal languages and automata.
- Explain the notion of undecidability and its importance for computer science.
- Explain the basic concepts from complexity theory.

## **Year 3 Modules**

### **Emerging Technologies**

The aim of this module is to provide an opportunity to study current research topics within Emerging Technologies. Students will be presented with emerging research taking place within areas of Computer Science and the impact such technologies are likely to have on society and business. Students will hone their research and interpersonal skills by interacting with members of the School's research themes. The precise aims for each element of this module will be documented and disseminated to the student ahead of the module. They will include both the development of knowledge of aspects of that topic, and the development of associated practical skills.

At the end of this module students will have gained knowledge on a number of advanced and contemporary areas in Computer Science, Software Engineering and Emerging Technologies.

On successful completion of the module, a student should be able to:

- Carry out independent research on a contemporary research topic within Computer Science.
- Assess the applicability of novel tools and techniques within a specified problem domain.
- Demonstrate an understanding of emerging research directions in computer science, and their impact on society and business.

### **One Semester Individual Project**

The aim of this module is to give each student the opportunity to show individual creativity and originality, to apply where appropriate knowledge and skills taught throughout the degree programme, to practise investigative problem-solving, communication, management and other transferable skills, and to demonstrate the ability to undertake an individual computer science project. The project will be executed independently by the student, under the guidance of a supervisor. It requires to conduct in-depth work on a substantial problem, which includes researching and analysing the problem, and finding and realising a solution. While projects may vary widely in the problem they address, the problem must be related to the students' degree programme.

On successful completion of the module, a student should be able to:

- Specify and plan a computer science project.
- Undertake a substantial computer science project in a professional manner.
- Understand the elements of a successful computer science project.
- Show an appreciation of best practice in solving computer science problems.
- Exhibit a sound knowledge in the subject area related to the project.
- Demonstrate an in-depth understanding of the technology and methodology used in the project.

## **Large-Scale Databases**

This module explores a range of database technologies that have been motivated by the demands of applications that create massive volumes of data with rapidly changing data types - structured, semi-structured and unstructured data. For example, management of location and geo-spatial information has resulted in extensions to conventional relational databases that can be supported by object-relational database systems. Access to massive quantities of social, scientific and commercial data on the web has resulted in more radical departures from the relational data model. The module introduces the modelling and management of large-scale datasets with a range of modern database technologies, including NoSQL document and graph databases.

On successful completion of the module, a student should be able to:

- Demonstrate an appreciation of applications of large-scale databases in a variety of commercial, scientific and professional contexts.
- Discuss how relational databases are extended with object-relational technologies to support management of spatial information.
- Understand the characteristics of and methods of processing geospatial information for purposes of storage and retrieval.
- Describe non-relational database approaches including document and graph databases to support access to large data sets.
- Be able to choose and develop a non-relational database solution suitable for the type of data and application considered.

## **Knowledge Management**

This module looks at how individuals and organisations generate, capture, transfer and exploit knowledge. We examine how the application of knowledge can make a significant difference to the success or failure of an enterprise, whether it be a business, service, or community. The course covers key questions such as, how can different kinds of knowledge be transferred? Can an organisation “learn”? And what roles can information and communication technologies play in knowledge transfer?

This module will analyse the role of knowledge as a resource in the learning organisation. It will consider the concepts of knowledge management alongside the organisational and individual factors which must be interpreted when implementing knowledge management systems within organisations to add organisational value.

This module requires a basic understanding of database and web technologies.

On successful completion of the module, a student should be able to:

- Explain and exemplify what organisational knowledge is.
- Illustrate and analyse the knowledge management lifecycle and activities.
- Judge and examine the capacity of technology to enable organisations to codify, share and use knowledge.
- Discuss how to measure the impacts of knowledge management activities on an organisation.
- Assess and recognise the potential of new and emerging technologies on knowledge management.

## **Security**

This module provides students with basic understanding of cryptographic tools and techniques that are used in modern systems to achieve security objectives, such as confidentiality, integrity, and authentication. Students are introduced to relevant practical aspects as well as to the mathematical foundations of these techniques. Finally, the module goes into real-world security protocols (such as SSL/TLS) and how they are composed as a combination of basic techniques.

Throughout the semester, students are given programming tasks (as formative assessment) to practice the principles and techniques learned in the lectures.

This module requires working knowledge of Probability, Discrete Mathematics (sets, permutations), Programming.

On successful completion of the module, a student should be able to:

- Describe how fundamental cryptographic algorithms and security protocols work.
- Design cryptographic protocols and attacks.
- Select and use appropriate cryptographic algorithms to achieve security objectives, such as confidentiality, integrity, and authentication.
- Analyse the role of the constituent cryptographic algorithms in a security protocol
- Evaluate cryptographic algorithms with respect to security and efficiency\

## **Forensics**

This module covers the principles, techniques, theory and applications central to computer forensics, The module focuses on computer file system fundamentals, detection, acquisition, analysis and report writing as well as coverage of legal and professional issues all of which focus on the practice of obtaining 'legally safe' evidence of criminal activity. Through a "hands-on" approach to learning forensic computing techniques using open-source and commercial forensic tools.

On successful completion of the module, a student should be able to:

- Evaluate the principles of computer forensic analysis and appreciate where and how these principles should be applied.
- Critically discuss the nature of digital evidence and the interpretations of that evidence obtained from computer forensics investigations.
- Evaluate the legal and procedural issues and be aware of the documentary and evidentiary standards expected in presenting investigative findings in a court of law.
- Analyse and evaluate the professional requirements of a computer forensics practitioner, and to critically discuss the challenges facing the computer forensics practitioner.
- Demonstrate knowledge and understanding of file structures both in a Linux environment and Windows, disk structures and use of a range of forensic tools and techniques.
- Understand the methods of data extraction from mobile phones.
- Understand methods of security logging and pattern matching for detection.
- Explain the link between technology and business processes in the context of gathering evidence.
- Describe the investigator's duty to the courts and explain the rules of evidence.



## **Artificial Intelligence**

Artificial intelligence (AI) is the study of computational methods that are capable of automating aspects of human intelligence. This module offers a general introduction to AI, focusing on how real-world problems can be represented in a suitable way, methods for reasoning about such representations, and effective algorithms for solving problems by intelligent exploration of the solution space.

On successful completion of the module, a student should be able to:

- Choose and apply an appropriate method for solving a given problem
- Describe and apply techniques to represent and reason with knowledge
- Understand concepts of fuzzy logic and approximate reasoning
- Explain and apply constraint satisfaction techniques.
- Critically discuss the ethical, societal, and technological limits and risks associated with the development and deployment of AI systems.
- Apply libraries to implement AI solutions.

## **Design Thinking and Prototyping for User Experience**

This module aims to provide the students with in-depth and hands-on understanding of user experience (UX) design in the context of software (from apps to large software), device platforms (from smartphones to ubiquitous computing and robots), and a wide range of interaction modalities (e.g., input and output), from voice and audio to gesture and tactile. The module will help to understand users and their needs better, and ethical, fair, accessible, sustainable, inclusive, usable, and effective user designs and theories. Additionally, students will learn what user experience is about and how to design and prototype digital tools that offer fulfilling user experiences. Moreover, students will learn about the number of factors that influence user experience; the theories that underlie good interaction design; and the methods and techniques designers use to create effective interactive products.

On successful completion of the module, a student should be able to:

- Understand UX design and relevant tools and methods.
- Gather user requirements and validate them in the context of use.
- Understand and apply design thinking approach.
- Understand and apply visual, audio, and/or tactile modalities in designing prototypes.
- Conduct evaluation studies and collect, analyse and validate data from the studies and present them (visually and verbally).

## **Introduction to Computational Robotics**

This module aims to provide a general introduction to the computational and algorithmic aspects of robotics, with a particular emphasis on how robots plan actions (i.e., make decisions) in navigation and manipulation tasks. Such tasks represent some of the current challenges to deploy robots in situations in which they must interact and collaborate with humans, e.g., when navigating social spaces (i.e., environments shared with people), or when handing over objects/tools to human teammates. The students will study robots as

(semi-)autonomous systems that continuously sense the surroundings, plan actions, and act. The module will provide a general overview of different aspects of the robot sense- plan-act loop, but it will mostly focus on how robots plan actions. The students will learn about the most common algorithms that solve different variations of the robot planning problem, while also getting a practical understanding of other correlated problems, such as robot localisation and mapping.

The students will understand the aforementioned challenges and learn about some available algorithmic solutions via instructor-led and hands-on/tutorials sessions, in which the students will also familiarise themselves with computational libraries (e.g., the Open Motion Planning Library) and middleware (e.g., the Robot Operating System) that are commonly used in robotics. Such theoretical and practical sessions will be complemented with guest lectures from both industry and academia. The students will also learn about the potential opportunities as computer scientists and robotics software engineers in the growing and in- high-demand market of autonomous and robotic systems.

On successful completion of the module, a student should be able to:

- Explain and describe robots as (semi-)autonomous systems that operate under the sense-plan-act loop.
- Identify the main applications of autonomous and robotic systems.
- Understand the most relevant computational and algorithmic aspects in robotics.
- Recognise the importance of computational robotics in human-robot interaction/collaboration (HRI/HRC).
- Propose algorithmic solutions to some of the common robot planning problems.
- Implement algorithmic solutions for the robot planning problem by using existing computational libraries and middleware for robotics.

### **Managing Change**

Software systems and services are key enablers for change in many organisations. However, software-enabled change is more complex in larger organisations as it needs to comply with established standards and processes for managing change, to deliver value to a wide range of stakeholders. This module aims to give students an appreciation of the difficulties inherent in managing change in organisations where technology is a key enabler of change. Students will investigate the difficulties of managing change in organisations through a range of case studies and will evaluate how these difficulties can be addressed through management standards, processes and techniques.

On successful completion of the module, a student should be able to:

- Develop a business case to assess options, costs, benefits, risks, timescales and evidence for a software project.
- Appreciate the difficulties inherent in delivering change in organisations.
- Critically evaluate the effectiveness of management standards, processes and techniques in addressing these difficulties.
- Discuss how legal, ethical and social issues affect managing change and provide justified recommendations/actions for addressing these issues.