OPIT Open Institute of Technology

Bachelor of Science in Modern Computer Science - Fact Sheet

Overall Programme Description

With input from employers, industry professionals and academics, the 180-ECTS OPIT online BSc degree in Computer Science was designed to provide students with a solid foundation in programming languages, software development, algorithms, database systems, and inference systems as well as with opportunities for career advancement in the field of Computer Science.

The program is divided into 6 terms distributed over 3 years, with a flexible schedule and an interdisciplinary approach that will allow students to better define their path and goals along the course.

The program includes 4 tech courses and 1 business course per term (from Term 1 through Term 4), a wide selection of industry-focused electives for students to choose from (Term 5) along with real-world challenges with peers and tutors as well as extracurricular activities, masterclasses and events with international partners and mentors.

All teaching and learning activities will take place online as the program is taught remotely through a dedicated platform and live interactions with professors, tutors, mentors, and company speakers.
Get-together events will be periodically organized online, where faculty members and students can interact in workshops and social events.

OPIT Open Institute of Technology

Target Audience				
	Ages 19 – 30	☒	Age 31 – 65	☒
	Age 65+	☒		

Target Group
The BSc in Computer Science is addressed to (a) those wishing to take up a career in the ICT industry, and (b) persons already employed in related sectors who wish to enhance their career prospects by acquiring a formal degree. The course prepares students either to find employment in the field or to undertake further studies in specific directions. Accordingly, the study plan has been designed with strong formative characteristics, sound theoretical foundations, and uses innovative pedagogies.

Entry Requirements

Application requests from prospective students are received by the Students Secretary Office, which will conduct an interview with the applicants.

Students will need to provide the following documents for admission:

1. Updated CV in English;
2. Copy of a valid ID (front and back);

Qualifications:

3. High secondary school certificate (MQF level 5); or
   University degrees (MQF level 6 or higher) in Computer Science, Computer Engineering, Economics, Physics, Mathematics, Statistics obtained at other institutions.

In addition, since the BSc will be taught in English, an IELTS 6 score or equivalent certification is required to non english-mothertongue participants to ensure a successful and full understanding of the training.

Students who do not hold the requested level must sit for the English Entry Test in order to certify the students' competences.

During the admission process of students wishing to enrol to the program, we will also ensure that such students have the required basic digital competence to successfully complete such a course. We will do so by administering to such students a standardised questionnaire that will cover aspects including, but not limited to: the availability of a PC with a webcam and speakers, the availability of an adequate internet connection, basic knowledge of operating systems and web browsers.

Learning Outcomes for Knowledge obtained at the end of the programme	The learner will be able to: a) Describe the inner workings of computing and information systems b) Define, describe, and use business processes in various fields c) Define, describe problems using formal mathematical tools d) Apply software engineering principles e) Define and describe the business-side of computer science projects f) Investigate the use of specialized algorithm design methodologies g) Analyse complex problems and propose technology based solution h) Acquire the ability to communicate technical concepts effectively i) Develop and use data science methods j) Develop and use cloud computing applications k) Explore different software frameworks for web, mobile and system programming
Learning Outcomes for Skills obtained at the end of the programme	The learner will be able to: a) Use different programming languages in systems, mobile and web environments b) Use different database technologies c) Apply business and project management strategies d) Apply algorithmic and mathematical reasoning to computation problems e) Map real-life processes and identify deficiencies f) Design software architectures g) Implement effective system and/or application software solutions h) Work effectively in groups i) Use current tools and methodologies in computing projects j) Analyse and compare alternative solutions to computing problems

Hours of Total Learning

1 ECTS is equivalent to 25 total hours of learning, inclusive of contact hours, supervised placement and practice hours, self-study hours and assessment hours.

Total Contact Hours [1] (Contact Hours are hours invested In learning new content under the Direction of a tutor/lecturer (e.g. lectures, participation in online forums, video-lectures)	1125	Supervised Placement and Practice Hours (During these hours the learner is supervised, coached, or mentored. Tutorial hours may be included here)	980
Self-Study Hours (Estimated workload of research and study)	2090	Assessment Hours (Examinations/ presentations/ group work/ projects, etc.)	305
Total Learning Hours	**4500 Hours**		

[1] *In the case of online learning, synchronous and asynchronous learning activities under the direction and control of an instructor are considered as contact hours.*

The Programme Structure

Module/ Unit Title	Compulsory (C) or Elective (E)	ECTS	MQF Level	Mode of Teaching	Mode of Assessment	Term
Technical English	Compulsory	6	6	Live lectures, asynchronous contents	Exercises, Tests	1
Computer Networks	Compulsory	6	6	Live lectures, asynchronous contents	Exercises, Tests	1
Programming Principles	Compulsory	6	6	Live lectures, asynchronous contents	Exercises, Tests	1
Computer Architectures	Compulsory	6	6	Live lectures, asynchronous contents	Exercises, Tests	1
Information Technology Fundamentals	Compulsory	6	6	Live lectures, asynchronous contents	Exercises, Tests	1
Foundational Mathematics	Compulsory	6	6	Live lectures, asynchronous contents	Exercises, Tests	2
Web Development	Compulsory	6	6	Live lectures, asynchronous contents	Exercises, Tests	2
Introduction to Operating Systems	Compulsory	6	6	Live lectures, asynchronous contents	Exercises, Tests	2
Data Structures and Algorithms	Compulsory	6	6	Live lectures, asynchronous contents	Exercises, Tests	2
Project Management and QA	Compulsory	6	6	Live lectures, asynchronous contents	Exercises, Tests	2
Introduction to Databases	Compulsory	6	6	Live lectures, asynchronous contents	Exercises, Tests	3
Cloud Computing Infrastructure	Compulsory	6	6	Live lectures, asynchronous contents	Exercises, Tests	3
Programming Paradigms	Compulsory	6	6	Live lectures, asynchronous contents	Exercises, Tests	3
Introduction to Artificial Intelligence	Compulsory	6	6	Live lectures, asynchronous contents	Exercises, Tests	3
Business Strategy	Compulsory	6	6	Live lectures, asynchronous contents	Exercises, Tests	3
Introduction to Software Engineering	Compulsory	6	6	Live lectures, asynchronous contents	Exercises, Tests	4
Introduction to Machine Learning	Compulsory	6	6	Live lectures, asynchronous contents	Exercises, Tests	4
Cloud Development	Compulsory	6	6	Live lectures, asynchronous contents	Exercises, Tests	4
Digital Marketing	Compulsory	6	6	Live lectures, asynchronous contents	Exercises, Tests	4
Introduction to Computer Security	Compulsory	6	6	Live lectures, asynchronous contents	Exercises, Tests	4

Cybersecurity	Elective	6	6	Live lectures, asynchronous contents	Exercises, Tests	5
Parallel and Distributed Computing	Elective	6	6	Live lectures, asynchronous contents	Exercises, Tests	5
Reinforcement Learning	Elective	6	6	Live lectures, asynchronous contents	Exercises, Tests	5
Machine Learning	Elective	6	6	Live lectures, asynchronous contents	Exercises, Tests	5
Complex Networks and their Applications	Elective	6	6	Live lectures, asynchronous contents	Exercises, Tests	5
Computer Vision	Elective	6	6	Live lectures, asynchronous contents	Exercises, Tests	5
Cloud Computing Automation and Ops	Elective	6	6	Live lectures, asynchronous contents	Exercises, Tests	5
Front-end Programming	Elective	6	6	Live lectures, asynchronous contents	Exercises, Tests	5
Mobile Programming	Elective	6	6	Live lectures, asynchronous contents	Exercises, Tests	5
Software Engineering	Elective	6	6	Live lectures, asynchronous contents	Exercises, Tests	5
Ethics of computer science and AI	Elective	6	6	Live lectures, asynchronous contents	Exercises, Tests	5
Agile Development and DevOps	Elective	6	6	Live lectures, asynchronous contents	Exercises, Tests	5
Game Development	Elective	6	6	Live lectures, asynchronous contents	Exercises, Tests	5
Thesis	Compulsory	30	6	-	Thesis	6
Total ECTS for Programme Completion		**180 ECTS**				

Technical English
Compulsory
6 ECTS
Term 1

Course description

English is the language used in most technical contexts and because of this, it is imperative that students get a good command of it. Throughout the course, students are presented with clear technical concepts and given information on how to present them in a way which is understandable by those present.

What's important is that the language is taught in context using real world examples thus helping the student understand the nuances of such a specialised use. The students will learn specialist terms alongside other basic English skills like reading, listening and communication. Practice is extremely important in any language because even though the basic constructs can be easily learnt, fluency and diction can only be acquired through active usage.

Since students might have a first language which is not English, grammatical rules will be regularly practised and summarised. This is done to reinforce the use of the English language within a work context and discourage code switching during such conversations. The topics presented will reflect the latest developments in technology taken from popular literature and which are relevant to students' applications.

The course will provide the basic linguistic competences necessary for the student to ultimately listen, read, write and critically think in English. A skill which is needed for academic and professional success.

The educational goals are defined in relation to the Common European Framework of Reference for languages (CEFR). Acquisition of the basic structures of the English language system (levels (B1/B2)) in order to develop both written and oral communicative competences.

Applying Knowledge and Understanding

At the end of the module/unit the learner will have acquired the following skills:

a) Utilize language skills necessary to succeed in a technical environment
b) Build basic information on technical issues for further studies
c) Deal with large quantities of technical readings and writings
d) Identify and make use of appropriate terms with particular reference to technical aspects
e) Choose phrases, oral and written, morphologically and syntactically correct and appropriate to the level of language proficiency required by the program
f) Translate and to interpret texts, oral and written, of adequate complexity and difficulty

Module-Specific Learner Skills At the end of the module/unit the learner will be able to a) Define, describe, and use both written and oral texts in Technical English b) Develop a good range of vocabulary in Technical English d) Construct English in situations of intercultural interaction/environments. e) Rewrite technical studies and business writing in English
Module-Specific Digital Skills and Competences At the end of the module/unit, the learner will be able to a) Use word processing software to write technical reports and business writing in English

Back to the Programme Structure

Computer Networks
Compulsory
6 ECTS
Term 1

Course Description

The Computer Networks module exposes students to the underlying workings of traditional networking ranging from a small office network to the internet. They will understand the basic network types and topologies. Furthermore, they will go through the process of creating packets of data, transmitting them, and routing them across the network. Throughout the process, they will understand how switching works and what happens in case of lost packages, especially when the network is congested.

To gain in-depth knowledge, students will be introduced to the Open Systems Interconnection (OSI) Model, which provides a conceptual model for systems interconnection. This is further split into seven layers of abstraction; Physical, Data Link, Network, Transport, Session, Presentation, and Application layer.

They will also learn to use the TCP/IP protocol suite, the primary standard in Computer Networks. This protocol is used to send and route simple packets of data and for real-time multimedia services such as video streaming.They will also understand the use of other available protocols (such as UDP).

Furthermore, they will be made to understand Network Operations (Ops) policies, plans and procedures; with an introduction to business continuity, high availability and disaster recovery.

Finally, students will have a practical component that teaches them to reflect on design and implementation considerations. These issues arise when installing and configuring switches and routers in simple networks using local or wide-area networks (LANs or WANs). They will also be exposed to essential tasks encountered during the setup of such networks, including troubleshooting, network optimization, and security.

Applying Knowledge and Understanding

At the end of the module/unit the learner will have acquired the following skills:

a) Design a simple computer network
b) Make effective use of the TCP/IP protocol
c) Effectively troubleshoot existing networks
d) Analyse and propose changes to current network configurations
e) Learn to use networking tools
f) Describe how packets travel across networks
g) Test existing configurations to identify inefficiencies

Module-Specific Learner Skills At the end of the module/unit the learner will be able to a) Explain how networks work and how they are instrumental to transmit data b) Define and describe the different constituents of a network c) Design and create simple architectures
Module-Specific Digital Skills and Competences At the end of the module/unit, the learner will be able to a) Make use of networking management tools

Back to the Programme Structure

Programming Principles
Compulsory
6 ECTS
Term 1

Course Description

The Programming Principles module is intended as the first introductory course to programming. It takes students without experience through a step-by-step process towards creating their first programs. This is achieved via a hands-on approach that highlights the fundamental concepts of most programming languages. The module will use C/C++ with the GCC compiler.

The module will first introduce the students to the inner workings of a computer. In particular, they will be exposed to variables and how they are used to access main memory. Following that, students will be introduced to more complex concepts such as data types and abstract types. With this knowledge, they can then proceed toward using fundamental programming concepts such as:

- Assigning and manipulating variables,
- Using conditional statements,
- Looping and iterations,
- Using more complex data types and data structures,
- Creating simple functions

Every program needs some debugging, and because of this, students will be guided toward solving issues that might arise in their code. In particular, they will be shown how to identify and solve syntactical or logical problems.

To create complex code, programmers do not just rely on their code but make use of other code available online for free. Thus, we will introduce the notion of libraries and explain how programs can use them. The idea of an Application Programming Interface will be defined, and students will be urged to make use of other functions in their code. The unit will also cover techniques to create efficient and effective code.

Applying Knowledge and Understanding

At the end of the module/unit the learner will have acquired the following skills:

a) Create efficient programs
b) Write and maintain reliable code
c) Effectively debug a computer program
d) Analyse and manipulate an existent program
e) Learn to use effective programming constructs
f) Describe the compilation and execution process of code

Module-Specific Learner Skills

At the end of the module/unit the learner will be able to

a) Explain how computer programming works
b) Define, describe, and use the different constituents of a program
c) Design and create a program capable of solving a simple problem

Module-Specific Digital Skills and Competences

At the end of the module/unit, the learner will be able to

a) Use C/C++ compilers

b) Test the program to ensure that it meets the intended requirements

c) Search and use external libraries

Back to the Programme Structure

Computer Architectures
Compulsory
6 ECTS
Term 1

Course Description

The Computer Architectures module first introduces the students to how a basic computer works. It starts by delving through the developments of traditional Von-Neumann Architecture and how it has evolved till today.

Students will then explore the movement of data within a machine, from secondary storage to primary storage. This is achieved via the data bus available on the motherboard. The link to external peripherals and interrupts is also explored at this stage. The module then focuses on the brains of the computer, the Central Processing Unit (CPU), and its various constituents, such as the arithmetic logic unit, the control unit, caches, multi-core architectures, etc.

Once the high-level concepts are dealt with, the focus will move to a granular lever toward low-level program execution. Students will then examine the fetch-decode-execute cycle and the peculiarities found across different families of processors (such as Reduced Instruction Set Computer RISK or Complex Instruction Set Computer architectures).

The storage of data as bits, bytes, words, etc., is also examined at this stage which leads to a discussion on information representation and the number system within a computer. Students can practise using different representations such as binary, decimal, octal, hexadecimal, ASCII character codes, etc. They will learn how to convert from one representation to another and what this means in terms of memory storage. This will help them understand more abstract data types such as integers, floating point variables, etc. Students will also learn about Hamming codes, how to use them, and their limitations within the context of error correction systems.

Finally, they will explore the various components of computer logic, including
Boolean algebra and expressions, basic logic gates, sequential and combinational logic circuits, simplification, design of Karnaugh maps, flip flops, shift registers, decoders, memory organization, full adders, etc.

Applying Knowledge and Understanding

At the end of the module/unit the learner will have acquired the following skills:

a) Design simple logic circuits
b) Perform conversions between different number systems
c) Effectively troubleshoot simple logic circuits
d) Analyze and propose changes to current computer architectures
e) Learn to use computer logic tools
f) Describe the Von Neumann Architecture and modern improvements to it
g) Test existing system and identify problems

Module-Specific Learner Skills At the end of the module/unit the learner will be able to a) Explain the inner workings of a computer b) Define, describe, and use the different components, what is their role and how they work together c) Design simple architectures
Module-Specific Digital Skills and Competences At the end of the module/unit, the learner will be able to a) Test existing digital systems and identify problems

Back to the Programme Structure

Information Technology Fundamentals
Compulsory
6 ECTS
Term 1

Course Description

The Information Communications Technology (ICT) module is an ideal foundation course for any student aspiring to deepen his knowledge in computing. It is designed to help them understand the basic principles of problem-solving, digital devices, and software development. After the modules, the students should have all the knowledge required to tackle various science, engineering and ICT areas. The skills learnt are also transferable to other fields and can be used in various situations encountered daily.

Through this module, students will take a hands-on approach to computing since it is designed to help them reflect on the technology they use daily and encourage them to ponder on how it works. They will also be encouraged to question how they can improve their problem-solving skills using technology as an enabler. Students will be able to focus on real-world problem-solving, mainly how this can be achieved through simple programming.

To achieve all this, the topics covered in this module include:

- Computational thinking
- Introduction to algorithms
- Introduction to data structures
- Language translators, interpreters and compilers
- IT project management methodologies
- Introduction to computer networks
- Introduction to cloud computing
- Principles of cybersecurity
- Introduction to data and databases
- Artificial intelligence and machine learning
- Future directions in Computer Science and Information Technology

The ICT fundamentals will provide students with the basic skills required to further their studies, equipping them for the ever-increasing open vacancies in the tech market.

Applying Knowledge and Understanding

At the end of the module/unit the learner will have acquired the following skills:

a) Demonstrate how the components, the architecture and the organisation inside a computer works
b) Practice the problem-solving process using various scenarios
c) Demonstrate the characteristics of operating systems and their applications
d) Develop a range of cognitive skills, including critical thinking skills

Module-Specific Learner Skills At the end of the module/unit the learner will be able to a) Explain the different components of a computer system and how it can connect to other machines b) Define, describe, and use computing techniques to address specific problems c) Design simple algorithms and solve basic problems
Module-Specific Digital Skills and Competences At the end of the module/unit, the learner will be able to a) Show the skills acquired to create algorithms which solve problems b) Solve computer-based problems using simple programs c) Use computer networks and show how they can be used to connect computers together

Back to the Programme Structure

Foundational Mathematics
Compulsory
6 ECTS
Term 2

Course Description

This Foundational Mathematics module aims to give students the basic mathematical knowledge required to understand fundamental concepts in various computer science courses. The module tries to develop the necessary, yet basic analytical skills of the students, which they can then apply to various computational problems. Students will be exposed to differential calculus for single and multivariable functions. Then, the module will give an overview of simple combinatorics notions (permutations, counting problems) and elements of probability and statistics (density functions, expectation, covariance and correlation).

Students will then discuss the theories of vectors and matrices, which are extensively used in modern algorithms. For instance, they form the basis of computer vision and graphics processing since they can be easily computed using a Graphical Processing Unit (GPU). In fact, GPUs (which were initially designed for gaming) are being extensively used to speed up complex algorithms. The module will then explore the use of linear algebra and transformations. Students will also get an introduction to vector spaces, eigensystems, etc.

In brief, the topics covered in this study unit include:

- Elements of differential calculus
- Elements of combinatorics
- Elements of probability theory
- Elements of vector geometry
- Matrices and determinants

Applying Knowledge and Understanding

At the end of the module/unit the learner will have acquired the following skills:

a) Apply various techniques for sequence and series analysis
b) Evaluate different theorems and their results
c) Practice various mathematical problems
d) Demonstrate the use of functions and variables
e) Show how to solve complex problems covered in the course
f) Apply different theorems to various computing problems

Module-Specific Learner Skills

At the end of the module/unit the learner will be able to

a) Explain the various uses of mathematics to solve computing problems
b) Describe how the mathematical toolkit can be used to address a specific challenge

Module-Specific Digital Skills and Competences

At the end of the module/unit, the learner will be able to

a) Design and propose simple algorithms

b) Use relevant tests to analyze the output from the different functions

↑ Back to the Programme Structure

Web Development
Compulsory
6 ECTS
Term 2

Course Description

The Web Development module is intended to give students a basic and generic understanding of how the web works by explaining the various components of the internet and how the world wide web operates upon that infrastructure. It starts by introducing core web programming principles while presenting the opportunities and challenges of these web-based technologies. The unit begins by introducing the fundamentals of the Hypertext Transfer Protocol (HTTP) while delving into issues and practices surrounding such a protocol. This will be presented through an architectural perspective, thus allowing students to understand how the web operates under the bonnet and helping them take full advantage of this complex distributed platform.

The unit then shifts the attention toward the browser, its anatomy, and its role in client-side architecture. At this stage, students will be introduced to markup languages; their use, and how they evolved. In particular, they will be introduced to and given the possibility to get their hands dirty with the Hypertext Markup Language (HTML), which is considered to be the language of the web, Cascading Style Sheets (CSS), and JavaScript. Successively, they will be introduced to server-side scripting, and exposed to technologies used within this context (e.g., JSON, XML, relational database), and the use of client-side techniques for document tree manipulation using the Document Object Model (DOM). Finally, they will also be exposed to Web debugging techniques and tools thus helping them troubleshoot both client and server-side code.

Most of the technologies mentioned in this module are based upon official standards issued by the World Wide Web (W3C) consortium. Through the analysis of the various technologies, students will also understand the paradigm shift required to design and build web systems as opposed to traditional applications. This unit also offers the basics for various other modules throughout the whole programme.

Topics discussed in the module include:

- Web development environment (servers, domains)
- Web development cycle
- Website prototyping tools
- Principles of website prototyping
- W3C standards
- UX & Web usability
- HTML, CSS
- JavaScript
- PHP
- Web development frameworks
- Principles of web testing
- Cross-platform compatibility
- Web Accessibility

Applying Knowledge and Understanding At the end of the module/unit the learner will have acquired the following skills: a) Write and maintain reliable web pages b) Manage secure client-server connections c) Analyse and manipulate page elements using client-side scripts d) Request and use remote data using server-side logic e) Carry out domain registrations f) Prepare domain hosting comparison reports
Module-Specific Learner Skills At the end of the module/unit the learner will be able to a) Explain how the internet and the world wide web works b) Define and describe the functions of the various components in web applications c) Design and propose a basic website
Module-Specific Digital Skills and Competences At the end of the module/unit, the learner will be able to a) Create efficient web applications b) Generate simple HTML pages

Back to the Programme Structure

Introduction to Operating Systems
Compulsory
6 ECTS
Term 2

Course Description

In this Module/Unit, students will be introduced to Operating Systems (OS) and to some system programming aspects at the OS level. System programming refers to low-level programming used to communicate directly with the underlying machine. It deals with various aspects, such as the OS core, the firmware, and the entire development environment.

An OS is probably the most critical software in any machine, allowing it to perform the various essential functions one would expect. We find OSs in all devices, being a computer, a tablet, a smartphone, or even integrated within embedded systems. Because of this, students will first gain exposure to the anatomy of an OS, the essential functions, and the different layers one would expect in modern OSs. They need to understand how to use the various functionalities available in an effective way. By understanding the limitations of the underlying hardware, they will not only be in a better position to cater to those limitations but also manage to go around them using clever software workarounds, thus exploiting these limitations during program execution.

To illustrate the various low-level concepts, the study unit will use basic concepts found in imperative programming languages, such as variables, functions and program flows, to control the different operations of the OS. This will enable students to dirty their hands with real examples so they can understand the machine's internal workings.

The primary teaching environment used is Linux. Because of this, a thorough introduction to this OS will be provided. Linux was chosen because it is available on most devices and provides a high-security efficient environment. Furthermore, major OSs such as Android and Mac OS all share a common ancestor with Linux since they all emerged from UNIX. Notwithstanding, nuances of other OSs, such as Windows, will also be explored. The C programming language will be used to perform the various experiments since it facilitates low-level programming. Through this module, students can access low-level file input/output, manipulate memory and perform signal handling. Towards the end of the unit, students will not only have an excellent understanding of C, but since they will appreciate the OS's underlying workings, their coding level will also improve drastically.

Applying Knowledge and Understanding At the end of the module/unit the learner will have acquired the following skills: a) Show competence in low-level systems programming b) Design, implement and explain low-level programs c) Explain the concepts of low-level constructs such as pointers, buffers, etc. d) Explain the use of caching and virtual memory, and demonstrate their impact on performance e) Discuss concurrency and the race condition which might arise f) Build well-structured systems programs using available libraries
Module-Specific Learner Skills At the end of the module/unit the learner will be able to a) Explain how OSs work b) Define, describe, and use systems programming routines c) Design and create low-level programs capable of solving issues close to the machine
Module-Specific Digital Skills and Competences At the end of the module/unit, the learner will be able to a) Make use of the C programming language b) Show a deep understanding of OSs

Back to the Programme Structure

Data Structures and Algorithms
Compulsory
6 ECTS
Term 2

Course Description

Any programmer must use Data Structures and Algorithms (DS&A) in his day-to-day job. They need to understand their properties, distinguish between them, and learn how to use them effectively. Because of this, the module is divided into two parts, the first looks at elements of Discrete Mathematics to give the user a solid foundation into the theoretical aspects behind DS&A, while the second focuses more on the applied aspect thus providing students with an opportunity to see DS&A in action.

Initially, the module covers the essential mathematical tools to describe the computational complexity of algorithms using the big-O notation. Then, the module will focus on various data structures (queues, lists, linked lists, trees, hash tables, etc.) and the computational complexity involved in performing operations on such structures. Successively, students will apply what they learned about mathematical structures in real life. They will be taught to program constructs such as arrays, stacks, queues, trees, graphs, linked lists, heaps, and hash tables in real-world applications using real programming languages (such as C++) and pseudo-code.

Students will also explore different well-known algorithms, including those for sorting, traversing a tree, and simple graph algorithms. In the end, the unit will cover algorithmic analysis for efficient code design.

The DS&A is important not only because the structures learnt are extremely useful and used in most software projects. But through this unit, students will learn to create efficient algorithms. Because it is useless coding a system which is unusable. Code efficiency should be an important element in the repertoire of every software developer. Thus, through this unit, we will ensure that students create efficient code which reaches its intended objectives.

Applying Knowledge and Understanding

At the end of the module/unit the learner will have acquired the following skills:

a) Define and describe the time and space complexity of algorithms and explain how the Big O notation is used to determine both
b) List, describe, and compare the main abstract data types and also describe the various sorting, tree, and graph algorithms covered in class

Module-Specific Learner Skills

At the end of the module/unit the learner will be able to

a) Apply time and space complexity analysis for code segments
b) Comprehend why algorithms with exponential time complexity are unfeasible
c) Implement the abstract data structures
d) Design algorithmic solutions to real-world problems
e) Create code that manipulates data structures using recursion
f) Apply basic algorithmic techniques to different design tasks
g) Define, describe, and use basic data structures such as lists, stacks, queues, trees, hash tables, and graphs
h) Define, describe, and use basic algorithms used to manipulate the data structures covered in the module
i) Use mathematical methods to evaluate the algorithmic performance

Module-Specific Digital Skills and Competences

At the end of the module/unit, the learner will be able to

a) Explain how algorithmic complexity can have a big impact on the development of efficient algorithms
b) Define and describe the different types of data structures
c) Design and create new data structures capable of solving different problems

Back to the Programme Structure

Project Management and QA
Compulsory
6 ECTS
Term 2

Course Description

The Project Management and Quality Assurance module starts by giving students an overview of different projects and their backgrounds. This is meant to showcase a hands-on approach to project management. The unit will then delve into identifying the various project components and categorize them into types, followed by a discussion on the organization's culture and how this affects the successful completion or otherwise of a project. In the end, every project manager seeks to complete a project successfully, and to do so; the module will investigate common risks or pitfalls that projects encounter. To help manage the project, we will also explore the use of different methodologies (such as but not limited to PRINCE2).

Once the overarching objectives of Project Management are explained, the student will delve into more granular details. They will have to identify the project boundaries, ensure the necessary resources are available and organize them for efficient implementation. They must clearly define the project's scope and produce a work breakdown to do so. The project's structure has to be clear by defining a governance and organizational set-up that will see the project through the various hurdles. Project reporting should be in place throughout the project based on the expected milestones. To ensure that project costs do not spiral out of control, resources should be adequately estimated, and a responsibility chart should be drawn to ensure everyone is aligned with the project's objectives.

Quality Assurance is a very important aspect of project management, so students must manage quality by preparing quality plans. They need to ensure that costs are under control at all stages by using tools such as control cubes, s-curves, etc. Time too needs to be managed using bar charts, networks, and other planning methodologies to ensure that the execution of the project conforms to the original plan.

Finally, the crux of all the planning lies in the execution, which has to be carried out effectively and efficiently. Students need to set up robust processes to make sure that they reach these goals, such as work approvals, allocation of resources, team management, progress measurement, forecasting, and project completion amongst others. Almost all projects exhibit some sort of variance, and because of this, they need to engineer recovery actions directly in the plan thus guaranteeing that the project goals are reached. The students will also be exposed to project management information systems which will help them manage very large and international projects.

Applying Knowledge and Understanding At the end of the module/unit the learner will have acquired the following skills: a) Apply the theoretical elements of project management to real-life scenarios b) Use project management techniques to ensure an effective governance and organizational set-up c) Ascertain the principles of accountability and competency through the use of rigorous project management methodologies d) Create detailed project management plans taking into consideration every aspect e) Compose calculated recovery action plans in order to tackle variances which might arise f) Demonstrate the use of Quality Assurance in all aspects of the project
Module-Specific Learner Skills At the end of the module/unit the learner will be able to a) Explain how project management works and apply it in different contexts b) Define, describe, and use complex functions of project management and quality assurance methodologies c) Design basic project management plans
Module-Specific Digital Skills and Competences At the end of the module/unit, the learner will be able to a) Comprehend how to use project management tools

Back to the Programme Structure

Open Institute of Technology

Introduction to Databases
Compulsory
6 ECTS
Term 3

Course Description

Data is one of the most precious assets in today's world, and it fuels most of the operations around us, from healthcare to education and commerce. However, it needs to be organized and processed in such a way that it can aid decision-support systems.

To achieve this goal, at the heart of most modern applications, one finds a database with ample data storage that allows for quick and efficient data retrieval. This unit's first part aims to introduce the students to the core concepts behind Databases (DBs) and database management systems (DBMSs). Students will be exposed to the background behind traditional DBs, their architecture, and fundamental concepts, together with design (using the Entity-Relationship model) and implementation. An essential element of most databases is querying, and in the relational DB model, this is achieved via the Structured Query Language (SQL). This is intended to give students the fundamentals required to explore large data repositories and allow them to further examine recent advancements in DBMS.

The second part of the unit deals with modern advances in databases such as object-oriented, semi-structured, multimedia, blockchains, and temporal database models. Students will also be exposed to the extensible Markup Language (XML) and how the various technologies which emerged from it are changing how we handle data. The students will also be familiarized with the concept of Triple Stores, Query optimization, Online Transaction Processing, NoSQL, Data Warehousing, and other essential technologies.

Finally, the student will learn how to design and implement an Information System using various data models. They will understand the pros and cons of using different approaches, especially when dealing with un/semi/structured data sources. The students will also learn the importance of using a DBMS in managerial decision-making. All of this will be supplemented with relevant case studies, which will present real-world applications of Databases to students so they can learn how to make organizations efficient and effective by using the stash of data they already possess.

Applying Knowledge and Understanding At the end of the module/unit the learner will have acquired the following skills: a) Define, describe, and explain problems requiring a database b) Explain how these challenges can be solved using a Database Management System c) Distinguish between the different database technologies and learn when they should be used d) Make use of different techniques to find a solution based upon the data available e) Name the main parts of a Database Management System f) List the resources required by such systems g) Design using the various modelling techniques h) Choose a design technique after the analysis of the various requirements i) Prepare table specifications, populate them and manipulate the data store in them j) Comprehend database queries k) Write basic triggers
Module-Specific Learner Skills At the end of the module/unit the learner will be able to a) Use the Database Management System covered during the course b) Compare the different techniques available c) Represent facts using different knowledge representations d) Comprehend why Database Management Systems are becoming increasingly relevant in today's world
Module-Specific Digital Skills and Competences At the end of the module/unit, the learner will be able to a) Comprehend concurrency and parallelism b) Learn to store and index data in an effective way c) Define and describe modern approaches in parallel computing d) Explain different methods suited for different kinds of Databases

↑ Back to the Programme Structure

Cloud Computing Infrastructure
Compulsory
6 ECTS
Term 3

Course Description

The Cloud Computing Infrastructure module aims to help students understand the growing needs in computing resources and how they can be addressed using remote computing architectures. The module will highlight the fundamental concepts behind clouds (such as virtualization, Infrastructure as a Service, Platform as a Service, Hybrid Models, and others) in relation to alternate distributed computing paradigms (such as Peer to Peer networks, Grid computing, and others). This will ensure that students decide on the best computing paradigm they should adopt when solving real-world applications.

The systems available for cloud computing are various and change according to the context. Because of this, the module will introduce different technologies developed by the major players, such as Microsoft - Azure, Amazon - AWS and Google - GCP. Through this practical approach, they will understand the core principles and concepts surrounding the various architectures (including but not limited to Hybrid Multicloud, Serverless, Microservices, Cloud Natives, etc.). They will then explore the myriad of supporting software that exists, thus allowing them to manage a cloud subscription effectively and deploy efficient architectures while keeping costs in check.

The module also delves into the practicalities of setting up a real cloud environment using the leading cloud service providers (such as Microsoft Azure) as a case study. It will expose students to create and provision virtual machines, manage storage solutions and apply lifecycle policies, use the Map Reduce paradigm, thus helping them understand how to simplify, transform and analyze large datasets. They will also be exposed to various cloud components for networking, storage (blob/file/queues), databases (SQL/NoSQL), web/mobile services, IOT and finally managing and orchestrating cloud solutions via monitoring and metrics. Other industry essential topics including technology utilization, security, encryption, authentication, scalability, and performance will also be covered. All of these components will allow students to design a robust cloud system that is also secure.

Finally, an overview of the business implications of using a cloud system will be covered, thus allowing students to gain a macro perspective on using the cloud within larger organizations. Since the dependency on the cloud is constantly increasing, this skill set will be essential in the coming years.

Applying Knowledge and Understanding At the end of the module/unit the learner will have acquired the following skills: a) Plan and setup analytical tasks that can be used with Cloud technologies b) Design a system which uses the Map-Reduce Paradigm c) Prepare financial budgets (pertaining to cloud expenses) that allow business owners to make smart decisions d) Compose security, performance and scalability plans using the latest Cloud technologies
Module-Specific Learner Skills At the end of the module/unit the learner will be able to a) Explain how an cloud system works b) Define and describe the functions of the various components in cloud architectures c) Design and propose a basic cloud system
Module-Specific Digital Skills and Competences At the end of the module/unit, the learner will be able to a) Apply the techniques learnt to design a cloud system capable of handling big data b) Setup industry standard platforms such as Google Cloud, Microsoft Azure or Amazon Web Services

Back to the Programme Structure

Programming Paradigms
Compulsory
6 ECTS
Term 3

Course Description

Students will have to face different programming languages and paradigms throughout their careers as ICT professionals. It is probable that every few years, the programming language of choice changes in line with the technological advancements of the time. To help them make informed decisions, this module aims to expose students to different programming paradigms. They must understand the programming languages' underlying principles and their various features.

The module will start by giving them a historical perspective to help them understand how the earliest programming languages have evolved into the mainstream ones we use today. Through this historic walkthrough, they will appreciate the various paradigms, their strengths, weaknesses, and differences. It will also help them realize how paradigm shifts helped programmers create better and more complex code.

The student will be exposed to the following programming paradigms:

- Imperative. The course will quickly go over C as a prototypical language to explain the imperative programming paradigm.
- Object-Oriented programming, which organizes software design around data and objects. The module will focus on this paradigm, since it plays a major role in modern software design methodologies. This provides various advantages over imperative programming, such as modularity, reusability, security, and flexibility. Students will learn about classes, objects, and interfaces in this part. They will then delve into more advanced topics such as abstraction, encapsulation, inheritance, polymorphism, and input/output. The languages of choice will be Java and Python.
- Functional programming. The module will briefly introduce students to functional programming using Python/Java. Advanced functionalities are not covered in this module.
- Brief overview of concurrent (i.e. parallel and multi-thread) programming, time permitting. Here, the module will quickly expose students to concurrency control and other aspects of concurrent programming.

Through this module, we aim to consolidate student's knowledge of object oriented programming, and at the same time we aim to give students a broader overview of different paradigms, thus helping them create large and more complex software projects.

Applying Knowledge and Understanding At the end of the module/unit the learner will have acquired the following skills: a) Apply OO software design methodologies to implement robust applications b) Use event handling to illustrate OO concepts c) Develop classes which exhibit inheritance and allow for re-use d) Create custom libraries e) Use various data structures to store complex objects in secondary memory f) Develop safe multithreaded programming g) Use one of the different Integrated Developers Environments (IDE) available
Module-Specific Learner Skills At the end of the module/unit the learner will be able to a) Explain the difference between different programming paradigms b) Design programs by following the object oriented paradigm c) Describe basic concurrent programs
Module-Specific Digital Skills and Competences At the end of the module/unit, the learner will be able to a) Install and use one of the many IDEs available b) Install software frameworks required to run Java and Python programs

Back to the Programme Structure

Introduction to Artificial Intelligence
Compulsory
6 ECTS
Term 3

Course Description

Artificial Intelligence (AI) is the field of study which aims to get machines to do things that humans do best. It is probably the most important technology of the decade and because of this, students need to understand what it is and how to use it. This module introduces them to the fundamental concepts of AI, however it does so by focusing on a wide range of topics so as to give the students a taste of theoretical aspects but ultimately focus on the practical viewpoints.

The module covers a range of topics including but not limited to:

- Introduction to the Python programming language
- The difference between classical and modern AI
- Main AI applications
- Knowledge representation & Reasoning
- Problem Solving
- Search Algorithms
- Decision Making
- Natural Language Processing
- Intelligent Agents
- Robotics
- The challenges and limitations of AI
- AI ethics
- The future of AI

By the end of the module, students should be in a better position to understand when and how to use AI.

Applying Knowledge and Understanding

At the end of the module/unit the learner will have acquired the following skills:

a) Evaluate different approaches and choose the best one
b) Represent real-world information using different knowledge representation techniques
c) Use any AI technique covered in the module to solve simple challenges
d) Plan and execute a project which uses various aspects of AI
e) Identify ethical issues which might arise from AI projects
f) Prepare conceptual solutions for different AI applications

Module-Specific Learner Skills

At the end of the module/unit the learner will be able to

a) Explain how AI approaches work
b) Comprehend the difference between AI and other subfields of computing

Module-Specific Digital Skills and Competences

At the end of the module/unit, the learner will be able to

a) Design and propose a simple AI system using available technologies

Back to the Programme Structure

Business Strategy
Compulsory
6 ECTS
Term 3

Course Description

The Business Strategy module is designed to provide students with a comprehensive understanding of the critical concepts and frameworks for developing effective business strategies in the digital age. The course will cover various topics, including market analysis, competitive analysis, and identifying opportunities and threats in the digital landscape.

Students will learn to conduct market research and analyse data to identify market trends and customer needs. They will also learn how to evaluate the competitive environment and develop a competitive strategy to gain a competitive advantage. In addition, students will learn how to identify and assess opportunities and threats in the digital marketplace and develop a strategic plan to capitalise on these opportunities and mitigate the risks posed by these threats.

The course will also cover the use of digital tools and techniques to gain a competitive advantage. Students will learn about social media marketing, search engine optimisation, data analysis, and how to use these tools to reach and engage with customers in the digital world. They will also learn the importance of developing a digital business model and implementing a digital marketing plan to drive customer acquisition and retention. In addition to lectures and discussions, the course will include hands-on activities and case studies to help students apply their knowledge to real-world scenarios.

By the end of the course, students will be equipped with the skills and knowledge necessary to develop and implement effective business strategies in the digital world. In addition to the core concepts and frameworks covered in the course, students will also have the opportunity to learn about the latest developments and trends in the digital landscape. This will include topics such as the impact of emerging technologies, the rise of digital platforms, and the growing importance of data and analytics in business.

Overall, the Business Strategy module is designed to provide students with a comprehensive understanding of the key concepts and frameworks for developing effective business strategies in the digital age. Through a combination of lectures, discussions, hands-on activities, and case studies, students will gain the knowledge and skills necessary to develop and implement successful business strategies in the dynamic and rapidly-changing digital marketplace.

Applying Knowledge and Understanding At the end of the module/unit the learner will have acquired the following skills: a) Apply market research and analysis techniques to identify market trends and customer needs b) Evaluate the competitive environment and develop a competitive strategy c) Assess opportunities and threats in the digital marketplace d) Develop a digital business model e) Implement a digital marketing plan f) Applying skills learnt to real-world scenarios through hands-on activities and case studies g) Develop and implementing effective business strategies h) Keep up-to-date with the latest developments and trends in the digital landscape
Module-Specific Learner Skills At the end of the module/unit the learner will be able to a) Explain how a business strategy is essential to ensure a success digital business b) Define, describe, and use the various business strategy techniques covered in the module c) Monitor new technologies being adopted
Module-Specific Digital Skills and Competences At the end of the module/unit, the learner will be able to a) Use digital tools and techniques, such as social media marketing and search engine optimisation, to gain a competitive advantage

Back to the Programme Structure

Introduction to Software Engineering
Compulsory
6 ECTS
Term 4

Course Description

Introduction to Software Engineering covers the fundamental processes and techniques for creating robust and reliable software systems. The module starts by looking into the various elements of the system life cycle (while also exploring the different models), requirements analysis, and several software design methodologies. Throughout the unit, we will explore different software architectures and delve into software testing, ensuring that the software works well and does what it was planned to do.

Apart from covering traditional techniques, we will also expose the students to modern concepts and methodologies in software development. Students will learn how to apply patterns, thus teaching them about tried and tested solutions for common problems. The emphasis behind this module is on high-quality software design, especially within a team where team dynamics might play a crucial factor in determining the success of a software project. Teamwork and project management within the context of software development will also be brushed upon in this module.

The tasks assigned throughout the module will ensure that the students gather the required skills from real-world examples to help them design and develop software systems in a team environment.

Topics covered will include (amongst others) :

- Classical Software Engineering principles (waterfall model, etc.)
- The System Life Cycle
- Object Oriented Design (with languages such as UML)
- Verification Techniques between different models
- Requirements Elicitation
- Software Testing
- Software Configuration Management
- Software Quality Standards

Applying Knowledge and Understanding

At the end of the module/unit the learner will have acquired the following skills:

a) Evaluate the use of various software engineering methodologies in different applications
b) Design robust and efficient systems
c) Identify common issues found in different platforms
d) Define and describe the requirements of different business models
e) Use different tools to specify the user requirements
f) Plan the successful execution of a software development project

Module-Specific Learner Skills At the end of the module/unit the learner will be able to a) Design a software system using rigorous software engineering methodologies b) Evaluate the different methodologies available and choose the most suitable one for the current project c) Outline the design of robust and efficient software components d) Implement complex systems as part of a team taking into consideration complicated team dynamics e) Define and describe different architectures, design patterns, testing methodologies
Module-Specific Digital Skills and Competences At the end of the module/unit, the learner will be able to a) Use different software engineering techniques and methodologies b) Create a software specification using appropriate notation c) Design a graphical user interface which adheres to important human computer interaction principles

↑ Back to the Programme Structure

Introduction to Machine Learning
Compulsory
6 ECTS
Term 4

Course Description

With the proliferation of various electronic devices (computers, tablets, smartwatches, smartphones, IoT, etc.), the world is suddenly producing vast amounts of data to such an extent that the absolute majority of information in existence today was created in the last few years. Today, data is the fuel that powers most of the systems we use, and because of this, it is crucial that we learn to harvest data, process, analyse and manipulate it in various forms to serve our ever-growing needs.

Machine learning allows us to do just that in a rigorous manner and gives us the confidence to answer the various questions arising from data analysis. The role of machine learning is multidisciplinary and can be used in many areas including but not limited to finance, physics, biology, and many others. With such data in hand, students can formulate novel hypotheses or confirm existing ones. Essentially, machine learning will allow them to build interesting models to predict behaviours and improve business decisions.

The Introduction to Machine Learning unit will help students gain a multi-disciplinary overview of the field using elements from computation, statistics, and machine learning. They will then go through the basic steps of a machine learning project, including the following:

- Introduction to the Python programing language for Machine Learning
- The Numpy library for manipulating multidimensional arrays
- The data-driven modelling paradigm
- Data preprocessing and visualization
- Model complexity and cross-validation
- Simple models for classification, regression, and clustering

The computational aspects will be developed by using the Python programming language and related open source libraries, such as numpy and scikit-learn. Python is the de-facto standard in Machine Learning and Data science, offering the possibility to use numerous open source resources and software environments. The module will stress the use of jupyter notebooks for developing code for machine learning projects and writing reports.

Applying Knowledge and Understanding

At the end of the module/unit the learner will have acquired the following skills:

a) Design simple machine learning models for specific applications
b) Create machine learning pipelines
c) Use Python to process and analyse different datasets
d) Use plotting libraries to programmatically create graphs describing the underlying datasets
e) Apply the lessons learnt to predict real-world phenomena.
f) Prepare data elements for further analysis

Module-Specific Learner Skills At the end of the module/unit the learner will be able to a) Explain how machine learning methods are able to extract interesting features from existing datasets b) Use the Python programming language and related functionalities c) Monitor new technologies being adopted as part of the machine learning toolkit
Module-Specific Digital Skills and Competences At the end of the module/unit, the learner will be able to a) Use the Python programming language effectively to analyse data b) Write Python programs to use simple machine learning models

Back to the Programme Structure

Cloud Development
Compulsory
6 ECTS
Term 4

Course Description

The Cloud Development module is intended to train developers wishing to create cloud solutions starting from requirements gathering up to the design and implementation. Cloud development is different from traditional software development because the applications running on the cloud need an internet connection to work, thus making the dynamics of handling data slightly more complex. Security, too, is a growing issue in cloud systems, and special considerations need to be taken. It also offers various advantages, such as automatic load balancing and real-time up/down scaling of the resources.

Students will be exposed to one of the major cloud platforms, such as Microsoft Azure. They will learn to implement complete solutions, deploy functions, utilise cloud storage, set up authentication/authorisation and manage web apps. They will also learn how to connect to and consume other cloud services, including third-party ones. Since they will be implementing an actual project, students will also have the opportunity to learn how to monitor these online services, troubleshoot them and optimise them based on the number of users using the system at any time.

To ensure a quicker learning curve, students will perform a substantial amount of work using one of the online sandboxes (such as the Microsoft Learn Sandbox). These environments will allow them to try out different programs safely without running the risk of affecting the cloud server. Through it, they can learn to manage online resources like integrating, transforming and consolidating data. They will then proceed to deploy cloud services, all within a controlled environment which replicates to functions of a real one.

Cloud is without doubt the future of most services. Thus it is imperative for any student to learn cloud development since it will most probably be an important part of most modern software development jobs.

Applying Knowledge and Understanding

At the end of the module/unit the learner will have acquired the following skills:

a) Select an appropriate scripting language for the development of cloud-based solutions
b) Compare and contrast scripting languages with other programming languages
c) Read programs in a range of scripting languages
d) Plan the deployment of cloud-based architectures
e) Operate cloud sandboxes
f) Demonstrate how to create web apps and services
g) Use cloud based tools for optimization and balancing
h) Apply best practices when creating cloud-based applications

Module-Specific Learner Skills At the end of the module/unit the learner will be able to a) Explain how a cloud application works b) Define and describe different components of cloud applications c) Design and propose a basic cloud-based application d) Use the various tools available
Module-Specific Digital Skills and Competences At the end of the module/unit, the learner will be able to a) Make use of a Cloud Sandbox to experiment with cloud architectures b) Create a project using real cloud application servers

Back to the Programme Structure

Digital Marketing
Compulsory
6 ECTS
Term 4

Course Description

This Digital Marketing module delves into today's digital media scenario by discussing how technology has changed marketing. Through it, students will learn to harness the potential of web marketing strategies and embrace the challenges in online environments.

The aim of this module is to provide students with a critical understanding of the key concepts, fundamentals, and challenges of online marketing. This is taken from a global perspective whereby social media is gaining importance, and the number of people shifting to mobile commerce is increasing. The module explores the impact of the internet on the traditional marketing mix. It examines the key ingredients of the online micro and macro environments. Moreover, it delves into analysing consumer behaviours and how they respond to social media strategies.

Students will then experience a typical online marketing strategy that addresses market segmentation, positioning, and communication. They will also experience the power of Customer Relationship Management applications and marketing approaches, especially within a dynamic online environment. Finally, they will dirty their hands by formulating an online marketing strategy and seek to implement it, keeping in mind the changing web environment.

Throughout the module, students will be exposed to successful case studies of web marketing. This will allow them to appreciate the various approaches used by these companies to integrate the Internet into their marketing strategy. They will also learn to use multiple best practices to facilitate electronic commerce using media channels such as affiliate marketing, search engine optimization, and web public relations.

Applying Knowledge and Understanding

At the end of the module/unit the learner will have acquired the following skills:

a) Define and describe the role of web analytics within digital marketing
b) Describe the strategic and operational aspects of web analytics tools and how they can influence to create new marketing levers
c) Apply creativity in the application of online analytics platforms to monitor and track web marketing activities
d) Compose marketing funnels to improve the outcomes of marketing strategies within the context of the modern business and its international dimension
e) Arrange existing strategies to make them more effective and in line with specific target groups

Module-Specific Learner Skills At the end of the module/unit the learner will be able to a) Explain how web marketing works b) Comprehend the importance of the marketing mix especially when applied to online commerce c) Design and propose a web marketing strategy
Module-Specific Digital Skills and Competences At the end of the module/unit, the learner will be able to a) Create web marking plans b) Use Customer Relationships Management tools to create highly targeted campaigns

Back to the Programme Structure

Introduction to Computer Security
Compulsory
6 ECTS
Term 4

Course Description

Most people today conduct a digital life where they spend most of their time connected to a screen and interacting with other humans or even bots. This is made even more accessible through the proliferation of digital devices such as laptops, tablets, smartphones, smartwatches, and other wearable devices. Because of this, it is estimated that most of the online information available today was generated in the past two years. Furthermore, with the propagation of Internet of Things (IoTs) devices and Artificial Intelligence (AI) applications, digital content production is set to increase drastically.

Thus, developing secure methods in the online world is becoming extremely important. Let's not forget that governments, entities, corporations and individuals all have different cybersecurity needs and their data has to be protected from malicious users. Considering the rising trend of digitising everything, it becomes more crucial to meet the needs of the different users and protect them from potential online dangers. With time, these perils are becoming more sophisticated since AI bots are also being used as digital accomplices in online crime.

Through this module, students will study the main issues surrounding cyber security. They will cover the core principles and analyse the implications and trade-offs of different methodologies. Finally, they will also look at current technologies and how they are evolving with the deployment of emerging technologies. The topics covered include:

- Cryptography and key management
- User authentication and identity management
- Secure Operating systems
- Network security
- Security on the Web and cloud systems
- Policies for risk management
- Privacy, anonymity, and censorship
- General Data Protection Rules (EU-GDPR)

Applying Knowledge and Understanding

At the end of the module/unit the learner will have acquired the following skills:

a) Demonstrate the security controls necessary to provide a required level of confidentiality, integrity, and availability in computer networks
b) Diagnose cybersecurity events
c) Investigate attacks on an organization's networks
d) Design incident response plans
e) Apply critical thinking and problem-solving skills to detect existent and future attacks on an organization's computer systems
f) Communicate effectively proposed information security solutions to technical and non-technical decision-makers
g) Apply business principles in an information security environment

Module-Specific Learner Skills At the end of the module/unit the learner will be able to a) Explain the intricate issues surrounding cyber security b) Define and describe different components of secure systems c) Design and propose secure systems
Module-Specific Digital Skills and Competences At the end of the module/unit, the learner will be able to a) Apply secure methodologies to protect and defend computer systems from cyber attacks

Back to the Programme Structure

Cybersecurity
Elective
6 ECTS
Term 5

Course Description

The Cybersecurity module builds upon the concepts learnt in the introductory module in computer security. It covers three main aspects; security, compliance, and identity fundamentals across cloud-based systems. The unit will use cloud services such as Microsoft Azure to ground the theoretical aspects into real-life applications thus illustrating the different concepts presented.

Students will first explore the various security methodologies, concepts and compliance principles. This should give them a good understanding of what they are and why they are essential. Within this context, they will be presented with real-life cases to help them understand the daily threats that system administrators face. It is quite alarming that the number of threats has been on the rise in recent years. Furthermore, these cybercriminals are becoming increasingly more sophisticated with the use of Artificial Intelligence technologies to help them find a breach in the cloud infrastructure. Hence why it is extremely important for students to get a good understanding of cybersecurity. They will then proceed further towards identity principles, concepts and types. With this information, they can try out authentication and access management procedures. This will allow them to assess whether such systems have adequate identity protection and governance capabilities.

They will then analyse the security aspects of various systems by looking into the security management capabilities of cloud platforms. This will include using industry-standard software which facilitates this function, such as Azure Sentinel, Microsoft 365 Defender, and InTune.

The final item is compliance, whereby students will examine information protection risks and data lifecycle management capabilities in cloud deployments. They will look at both internal and external risks, analysing them in detail, while keeping in mind the resource governance capabilities which cloud platforms provide.

In the end, students would have gained a good knowledge of security, compliance, and identity across cloud-based platforms. They would also have tried them in real-life thus giving them the much sought-after experiences that employers require.

Applying Knowledge and Understanding At the end of the module/unit the learner will have acquired the following skills: a) Apply secure methodologies to protect and defend computer systems from cyber attacks b) Demonstrate the security controls necessary to provide a required level of confidentiality, integrity, and availability in computer networks c) Diagnose cybersecurity issues and take action against them d) Investigate attacks on an organization's networks e) Analyse security, compliance and identity issues f) Apply critical thinking and problem-solving skills to detect existent and future attacks on an organization's computer systems g) Communicate effectively proposed information security solutions to technical and non-technical decision-makers h) Apply business principles in an information security environment
Module-Specific Learner Skills At the end of the module/unit the learner will be able to a) Explain complex issues surrounding cyber security b) Comprehend security, compliance and identity fundamentals c) Design and propose secure systems
Module-Specific Digital Skills and Competences At the end of the module/unit, the learner will be able to a) Gain experience in securing cloud systems like Microsoft Azure

Back to the Programme Structure

Parallel and Distributed Computing
Elective
6 ECTS
Term 5

Course Description

The vast volume of data available for analysis provides interesting avenues of development, but also hides significant challenges in terms of technical feasibility. To this end, parallel and distributed computing systems opened the way to advanced applications in data analysis and beyond.
The module discusses challenges faced in constructing parallel and distributed applications, including testing, debugging and performance evaluation.

Topics discussed in the module include:

- Multithreaded programming (pthreads, MPI, etc.)
- Asynchronous & Synchronous Computation/Communication,
- Concurrency control,
- Parallel & Distributed Algorithms,
- Load balancing & Scheduling,
- Fault tolerance,
- Scalability & Performance Analysis and Tuning,
- Distributed Data Storage/Computing

Different programming languages will be used, depending on the specific application scenario. Programming languages include C/C++, Java, and Python.

Applying Knowledge and Understanding

At the end of the module/unit the learner will have acquired the following skills:

a) Design and implement simple parallel algorithmic solutions
b) Design and implement simple distributed algorithmic solutions

Module-Specific Learner Skills

At the end of the module/unit the learner will be able to

a) Check the consistency of parallel/distributed implementations

Module-Specific Digital Skills and Competences

At the end of the module/unit, the learner will be able to

a) Write and compile parallel algorithms in C++ and Java
b) Write and run Python scripts
c) Link to different third-party libraries

Back to the Programme Structure

Reinforcement Learning
Elective
6 ECTS
Term 5

Course Description

Autonomous systems must acquire the ability to make sensible decisions in order for AI to reach its full potential. Systems can be trained in decision-making using the effective paradigm of reinforcement learning (RL). Robotics, gaming, consumer modelling, autonomous driving vehicles, healthcare, and other fields can all benefit from the use of RL algorithms.

Students will receive an introduction to RL while gaining knowledge of the fundamental concepts, algorithms and open problems in the field. The module content will be delivered through a combination of theoretical lectures and practical coding assignments, including examples of modern deep RL architectures.

A tentative list of topics includes:

- Finding the best strategies in an unknown environment
- Framework for RL: Markov decision processes
- Temporal Difference (TD) RL algorithms: Q-learning, SARSA
- Policy-gradient based RL algorithms: REINFORCE, actor-critic methods
- RL algorithms for deep neural networks:
- Deep Q-networks
- Advantage Actor-Critic
- Trust Region Policy Optimization and Proximal Policy Optimization
- Deep RL successes for Game-Playing AI: AlphaGo (Chess, Go) and AlphaStar (StarCraft II)
- Sparse Reward Problem in RL:
- Reward Shaping
- Credit Assignment Problem
- Learning from human experts:
- Behavior Cloning, Imitation Learning and GAIL
- Inverse Reinforcement Learning
- RL and human psychology:
- Intrinsic rewards
- Advanced topics in RL:
- Adversarial RL
- Explainable RL
- Safety and Ethics in RL

Applying Knowledge and Understanding

At the end of the module/unit the learner will have acquired the following skills:

a) Evaluate different RL approaches and choose the best one
b) Use any RL technique covered in the module to solve simple challenges
c) Plan and execute a project which uses various aspects of RL
d) Prepare conceptual solutions for different AI applications

Module-Specific Learner Skills At the end of the module/unit the learner will be able to: a) Explain how RL approaches work b) Describe the difference between RL and other subfields of AI for decision-making
Module-Specific Digital Skills and Competences At the end of the module/unit, the learner will be able to a) Design and propose an RL-based system using available technologies in Python language

Back to the Programme Structure

Machine Learning
Elective
6 ECTS
Term 5

Course Description

The Machine Learning module builds upon different modules and in particular on the Introduction to Machine Learning module. The Machine Learning module gives students advanced knowledge on different data-driven methodologies. Notably, in this module students will look at different machine learning models and related learning algorithms that deal with classification, regression, clustering, and visualization tasks. They will learn the core principles and how to use standard frameworks to train, evaluate, and use them with real-world examples.

Students will explore different methodologies of increasing complexity, starting from relatively simple models like random forests, and concluding with modern deep learning models. The conceptual and theoretical notions will be accompanied by practical examples using scikit-learn and keras, two well-known Python frameworks.

The topics covered in the module include, but are not limited to,

- Data pre-processing and validation
- Training paradigms in machine learning
- Support vector machines
- Dimensionality reduction and manifold learning
- Clustering
- Multilayer perceptrons
- Convolutional neural networks
- Recurrent neural networks
- Generative models

The module will stress the use of jupyter notebooks for developing code for Machine Learning projects and writing reports.

Applying Knowledge and Understanding

At the end of the module/unit the learner will have acquired the following skills:

a) Design complex machine learning solutions to deal with specific applications
b) Create advanced pipelines using machine learning algorithms
c) Use Python to program and analyse different datasets
d) Use machine learning libraries to analyse the data and extract insights describing the underlying datasets
e) Use scikit-learn and keras
f) Apply the lessons learnt to predict real-world phenomena
g) Prepare data elements for further analysis

Module-Specific Learner Skills At the end of the module/unit the learner will be able to a) Explain how data science manage to extract interesting features from existing datasets b) Describe the organization and main functionalities of scikit-learn and keras c) Monitor new technologies being adopted as part of the data science domain
Module-Specific Digital Skills and Competences At the end of the module/unit, the learner will be able to a) Use scikit-learn and keras b) Import and make predictions using datasets of various kind

Back to the Programme Structure

Complex Networks and their Applications
Elective
6 ECTS
Term 5

Course Description

Complex networks are ubiquitous in the world around us: biological and chemical networks, communication networks, transportation networks, social networks, and power networks, to name a few.
In this module, students will first be introduced to the formalization of network representations. Successively, we will delve into the characterization and analysis of properties of complex networks, together with their interpretations for some of the aforementioned real systems. The theoretical and methodological notions introduced in the module will be supported by practical examples with the Python programming language and the networkx library.

Students who complete this course will receive a broad introduction to recent work in this area, understand the strengths and weaknesses of modeling the network, and may apply networks and their analysis in a variety of configurations.

A non-exhaustive list of topics covered in the module follows:

- Mathematics of networks and their representation
- Weighted and directed networks
- The adjacency, Laplacian, and incidence matrices
- Degree, paths, components
- Independent paths, connectivity, and cut sets
- Degree centrality, eigenvector centrality, katz centrality, PageRank
- Hubs and authorities, closeness centrality, betweenness centrality
- Transitivity, reciprocity, similarity, assortative mixing
- Components
- The small-world effect: Six degrees of separation
- Degree distributions. Power laws and scale-free networks
- Distributions of other centrality measures
- Random graphs
- Models of network formation

Applying Knowledge and Understanding

At the end of the module/unit the learner will have acquired the following skills:

a) Evaluate different approaches and choose the best one
b) Represent real-world information using the complex networks framework
c) Use any complex network methods covered in the module to solve simple challenges
d) Plan and execute a project which uses various aspects of complex networks
e) Prepare conceptual solutions for different applications

Module-Specific Learner Skills At the end of the module/unit the learner will be able to a) Explain how complex network approaches work b) Define and describe the difference between the complex networks framework and other fields in Data Science
Module-Specific Digital Skills and Competences At the end of the module/unit, the learner will be able to a) Design and develop Python programs to analyze datasets of complex networks

Back to the Programme Structure

Computer Vision
Elective
6 ECTS
Term 5

Course Description

Computer Vision is a subfield of Artificial Intelligence that deals with the processing and understanding of visual data from the world around us. It involves the use of algorithms and deep learning techniques to interpret and analyse images and video, in order to extract useful information and make decisions based on that data.

In this practical introductory course, students will learn the fundamental concepts and techniques of this field. They will begin by exploring the basics of image processing, including filtering, image warping, feature detection/extraction, segmentation, and stereo/multi-view reconstructions.

Next, students will learn about the use of deep learning algorithms in computer vision. This will include an introduction to the basics of deep learning, including the use of neural networks and convolutional neural networks (CNNs) for image recognition and classification. Students will also learn about the use of other deep learning techniques, such as generative adversarial networks (GANs) and autoencoders, in computer vision applications.

Throughout the course, students will have the opportunity to apply the concepts and techniques they have learned through hands-on exercises and software projects. This will involve the use of popular computer vision libraries and frameworks, such as OpenCV and Keras, to build and train their own computer vision pipelines.

In addition to the technical aspects of computer vision, the course will also cover the ethical and societal implications of this technology. This will include discussions on the potential uses and abuses of computer vision (such as using DeepFakes), as well as the ethical considerations that need to be taken into account when working with visual data.

By the end of the course, students will have a solid understanding of the basics of computer vision and be able to apply some conventional computer vision and more recent deep learning algorithms to real-world problems. They will have the skills and knowledge needed to continue learning and exploring the field on their own and be well-prepared to pursue further study or careers in computer vision and related fields.

Applying Knowledge and Understanding

At the end of the module/unit, the learner will have acquired the following skills:

a) Define the fundamental concepts and techniques of computer vision, including image processing and the use of deep learning algorithms

b) Manipulate and transform digital images using appropriate algorithms

c) Identify the mathematical foundations of computer vision, including linear algebra and statistics

d) Describe the use of deep learning algorithms, including neural networks and convolutional neural networks, in image recognition and classification

e) List other deep learning techniques, such as GANs and autoencoders, and their applications in computer vision

f) Write computer vision techniques and algorithms through practical exercises and projects

g) List the ethical and societal implications of computer vision technology, and the ability to consider these considerations in the development and use of visual data

Module-Specific Learner Skills

At the end of the module/unit the learner will be able to

a) Apply the basic algorithms of computer vision in various applications

b) Design computer vision applications

c) Manipulate and transform digital images using algorithms

d) Use deep learning algorithms to perform image recognition and classification tasks

e) Apply other deep learning techniques, such as GANs and autoencoders, to computer vision problems

f) Implement and train deep learning models for computer vision tasks using popular libraries and frameworks such as OpenCV and TensorFlow

g) Consider the ethical and societal implications of computer vision technology in the development and use of visual data.

Module-Specific Digital Skills and Competences

At the end of the module/unit, the learner will be able to

a) Learn to use OpenCV

b) Gain experience with Tensorflow and Keras

Back to the Programme Structure

Cloud Computing Automation and Ops
Elective
6 ECTS
Term 5

Course Description

Cloud computing is increasing in popularity, and computer scientists need to handle the application development aspects with networking, monitoring, storage, and security. They must ensure that the business processes they create are secure, scalable and reliable. Because of this, the Cloud Computing Automation and Ops module covers advanced operations such as virtualisation, nosiness continuity, disaster recovery and setting of data platforms. To understand and appreciate the value of these technologies, students will cover real-life examples and use extensive case studies.

Students will learn the key concepts and principles of DevOps, understand the benefits derived by business. They will understand the fundamentals of Continuous Integration and Continuous Delivery (CI/CD), lean and agile methodologies, service delivery process, automation infrastructure.

They will have hands-on experience of developing a sample cloud application using the actual cloud infrastructure framework (such as Microsoft Azure). They will understand how to build automation pipelines, using code repositories (like GitHub), orchestrating the deployment via orchestration language (such as Ansible). Hence exposing them through the software development iterations followed in the industries using agile methodologies.

In the end you will have necessary knowledge of working as DevOps engineer who have subject matter expertise in working with people, processes, and products to enable continuous delivery of value in organisations.

Applying Knowledge and Understanding

At the end of the module/unit the learner will have acquired the following skills:

a) Plan and setup secure data and authentication applications
b) Apply the techniques learnt to design a cloud system capable of handling complex systems
c) Practice strong analytical and technical problem-solving skills
d) Setup industry standard platforms such as Google Cloud, Microsoft Azure or Amazon Web Services
e) Examine different architectural solutions and make recommendations
f) Compose implementation plans using the latest Cloud technologies
g) Ensure that architectural standards are adhered to
h) Provide guidance to engineers working on deploying cloud solutions

Module-Specific Learner Skills At the end of the module/unit the learner will be able to a) Explain how an complex cloud system and big data applications work b) Define and describe the functions of the various components in a cloud architecture especially in relation to security, resilience and robustness c) Design and propose a complex cloud system which satisfies the business requirements of an organisation
Module-Specific Digital Skills and Competences At the end of the module/unit, the learner will be able to a) Microsoft Azure tools b) AWS tools

Back to the Programme Structure

Front-end Programming
Elective
6 ECTS
Term 5

Course Description

The Front-end Programming module is designed to give students an advanced overview of modern web development frameworks. This is important because most of today's applications are created directly as web applications. This comes with its challenges because these applications will have to deal with limited or unstable internet connectivity, browser restrictions, different screen sizes, and various other limitations.

This course provides a good start for anyone wanting to further his career in web development. Students will learn about the daily responsibilities of a web developer and get a general understanding of how the world wide web works. They will then create real-world web applications that work fine and are easy to maintain. The module will initially refresh basic web concepts encountered in previous units such as HTML, CSS, DOM tree, JSON, etc. They will then shift their focus towards JavaScript and major User Interface frameworks like Bootstrap or React.

These frameworks are important because they increase the app development speed by significantly reducing the amount of code that needs to be written and tested. The underlying libraries are very stable and its data binding approach facilitates software testing. Deployment is greatly facilitated requiring no advanced knowledge and the web pages produced are SEO-friendly. Furthermore, they also provide advanced debugging tools which allow programmers to easily fix software issues. Finally, since major systems like Netflix, PayPal, Airbnb and eBay already use such frameworks, user-generated content is widely available online making it easy for students to explore other advanced features.

As part of their assessment, students will be asked to create a web application using the main technologies covered during the module.

Applying Knowledge and Understanding

At the end of the module/unit the learner will have acquired the following skills:

a) Use reusable components to render pages where data changes over time
b) Create more scalable, responsive and maintainable web applications
c) Create dynamic and interactive web applications
d) Use forms to allow users to input data in web page
e) Build an application using the framework covered in the module
f) Write and maintain reliable web applications
g) Manage secure connections with the server
h) Analyse and manipulate page elements using client-side scripts
i) Generate complex HTML pages

Module-Specific Learner Skills At the end of the module/unit the learner will be able to a) Explain how Web Frameworks work b) Define and describe the functions of the various components in web applications c) Design and propose a complex web application
Module-Specific Digital Skills and Competences At the end of the module/unit, the learner will be able to a) Learn to use an industry standard web framework

Back to the Programme Structure

Mobile Programming
Elective
6 ECTS
Term 5

Course Description

The Mobile Programming module is intended to give students a thorough understanding of app design, implementation and deployment using the major mobile platforms. Apps will be created using native platforms and will allow developers to focus on mobile-first applications.

The module will focus on Android and IOS development since they are the two major platforms dominating the market for the past decade. Throughout the course, students will learn to use Flutter and Swift, the main languages for Android and IOS, respectively. Kotlin will also be mentioned since it is an upcoming language based on Java which has been gaining traction on the Android platform.

The best way for students to learn the language in the programming modules is to try various examples. Because of this, coursework and classroom exercises will increase the student's knowledge of the area. They will have the opportunity to create multiple fully functional mobile examples, thus highlighting different concepts. Students will gain exposure to the latest technologies while learning the fundamental building blocks of application development, such as understanding the basic architectures of various platforms, the development life-cycle, elements of the Graphical User Interface and the model-view-controller design pattern. A discussion will also ensue on the User Experience since mobile programmers must ensure that the screen space available is fully utilised. They have to guarantee that the navigation through the app is smooth and intuitive, thus allowing users to reach their objectives in the shortest time possible.

Developing basic Android, iOS, or hybrid cross-platform applications during the module will give students valuable hands-on experience and help them enhance their portfolios. A combination of theory and practical programming experience will best prepare them for a career which is very much in demand. The course will also expose individuals to the management of development teams who want to execute mobile application projects efficiently. At the end of the unit, students will understand the fundamentals of building cutting-edge mobile applications and how to publish them on the different app stores.

Applying Knowledge and Understanding At the end of the module/unit the learner will have acquired the following skills: a) Demonstrate proficiency in using Flutter and Swift as well as an understanding of the Kotlin language b) Practice and gain hands-on experience creating fully functional mobile apps using native platforms c) Optimize screen space and navigation in mobile applications d) Assemble diverse teams capable of creating mobile app projects e) Develop and publishing mobile apps on the Android and iOS app stores f) Create fully functional mobile apps that highlight different concepts learnt during the course
Module-Specific Learner Skills At the end of the module/unit the learner will be able to a) Explain the different components used in mobile programming b) Define and describe how natives apps are constructed for different platforms c) Design complex mobile apps on the major platforms
Module-Specific Digital Skills and Competences At the end of the module/unit, the learner will be able to a) Create an enhanced digital portfolio of mobile apps

Back to the Programme Structure

Software Engineering
Elective
6 ECTS
Term 5

Course Description

Software engineering is the application of engineering principles to the design, development, and maintenance of software systems. This course provides an introduction to the field of software engineering, covering fundamental concepts and techniques used in the construction of large, complex software systems.

Through lectures, discussions, and hands-on activities, students will learn about the software development process, including requirements gathering, design, implementation, testing, and maintenance. They will also explore various software engineering tools and methodologies, such as version control, issue tracking, and agile development.

In addition to these technical skills, the course also touches on the non-technical aspects of software engineering, such as project management, collaboration, and communication. Students will learn how to work effectively in teams, manage deadlines and budgets, and communicate their ideas clearly and effectively.

By the end of the course, students will have a strong foundation in the principles and practices of software engineering, and will be prepared to start building their own software projects. Some of the topics covered in this course include:

- The software development process, including requirements gathering, design, implementation, testing, and maintenance
- Project management and planning
- Testing and debugging techniques
- Version control and issue tracking tools, such as Git and JIRA
- Collaboration and communication in software engineering teams
- Mention of agile development methodologies, such as Scrum and Kanban

This course is designed to provide students with hands-on experience in the field of software engineering, covering both the technical and non-technical aspects of the discipline. Students who complete the course will have a strong foundation in the principles and practices of software engineering, and will be well-prepared to work in teams on real-world problems.

Applying Knowledge and Understanding At the end of the module/unit the learner will have acquired the following skills: a) Apply software design principles to any project b) Practice the different techniques learnt during the course c) Plan extensively taking into consideration the resources available and the teams involved d) Design detailed diagrams using graphical notations learnt during the course e) Use different techniques to collect detailed specifications from users f) Prepare all the necessary groundwork to ensure a successful project
Module-Specific Learner Skills At the end of the module/unit the learner will be able to: a) Explain the different software engineering techniques used in the course b) Define and describe the various software engineering methodologies c) Design a detailed development plan which can be used to guide the different teams involved in the project
Module-Specific Digital Skills and Competences At the end of the module/unit, the learner will be able to a) Demonstrate the use of software engineering methodologies in real-life applications

Back to the Programme Structure

Ethics of computer science and AI
Elective
6 ECTS
Term 5

Course Description

The Ethics of Computer Science (CS) and Artificial Intelligence (AI) module explores the various implications of the technologies developed by practitioners. This course will introduce students to the fundamental ethical theories and frameworks used to analyse and evaluate the ethical implications of relevant CS and AI technologies. It will provide opportunities for students to apply these theories and frameworks to real-world scenarios involving CS and AI.

Students can explore various ethical issues throughout the module, including privacy, bias, fairness, transparency, accountability, and autonomy. They will learn how to identify and analyse ethical dilemmas that may arise in developing these technologies and the potential solutions available.

In addition to discussions and lectures, this course will include various hands-on activities and case studies that will allow students to apply their knowledge of ethical theories and frameworks to real-world scenarios. For example, students may work in teams to analyse and evaluate the ethical implications of a new AI system being developed by a tech company. They may also explore the potential consequences of a proposed policy related to using personal data by technology firms.
By the end of the course, students will have a deep understanding of the ethical and societal implications of CS and AI. They will be prepared to apply their knowledge to various real-world scenarios while having the skills and confidence to engage in informed and productive discussions about the ethical issues surrounding these technologies.

Some of the specific topics that may be covered in this course include:

- The role of ethics in CS and AI
- Fundamental ethical theories and frameworks, such as utilitarianism, deontology, and virtue ethics
- Ethical issues related to privacy, bias, fairness, transparency, accountability, and autonomy
- Approaches to identifying and analysing ethical dilemmas in CS and AI
- Strategies for addressing ethical dilemmas and mitigating potential negative consequences
- Case studies and hands-on activities involving real-world scenarios involving CS and AI

This course is designed to provide students with a comprehensive introduction to CS and AI ethics. Through a combination of lectures, discussions, and hands-on activities, students will gain a deep understanding of the ethical and societal implications of these technologies. They will be prepared to apply their knowledge to various real-world scenarios.

Applying Knowledge and Understanding

At the end of the module/unit the learner will have acquired the following skills:

a) Practice ethics audits on CS and AI case studies while considering the ethical implications of the technologies involved

b) Demonstrate the ethical dilemmas that may arise in the development and use of these technologies

c) Apply ethical theories and frameworks to real-world scenarios involving privacy, bias, fairness, transparency, accountability, and autonomy

d) Develop strategies for addressing ethical dilemmas and mitigating potential negative consequences of software systems

e) Engage in informed and productive discussions about the ethical issues surrounding software projects

f) Apply knowledge and skills learned in this course to real-world scenarios

g) Conduct AI audits on existent projects

h) Create a plan of action to ensure that software projects take into consideration the ethical implications

Module-Specific Learner Skills

At the end of the module/unit the learner will be able to

a) Explain the different ethical issues that might arise in CS and AI systems

b) Comprehend how to track ethical issues and mitigate them

c) Design structure plans to ensure that all the ethical implications are taken into consideration

Module-Specific Digital Skills and Competences

At the end of the module/unit, the learner will be able to

a) N/A

↑ Back to the Programme Structure

Agile Development and DevOps
Elective
6 ECTS
Term 5

Course Description

Agile methodologies have been in use for several years, and practices such as automated building & testing, together with many others, led to the development of DevOps. That is why this module marries together Agile methodologies and aspects of DevOps. Both are important in successfully implementing software projects and complement each other.

The module first starts with a run-through of Agile practices. Students will then learn about the critical elements of scrum and how to conduct team functions. Following that, they will delve into writing good user stories to help them keep track of their team's progress using a Kanban board. Finally, they will be introduced to a product backlog where the team and customers discuss priorities in a flexible and blameless environment. It is a known fact that such an approach tends to lead to high levels of efficiency, stabilise requirements across sprints and lead to better overall success. The students will be taught to collaborate with others, organise themselves in teams under the guidance of a scrum master, and finalise projects. To help them out, various real-world scenarios will be presented, thus placing the students in actual situations encountered by development teams.

The second part of the module focuses on the amalgamation of development and operations, generally referred to as DevOps. This is one of the fastest-growing areas in the workforce, so students need to get a good grounding in it.
The module explores essential topics such as developing an instrumentation strategy, site reliability engineering, security and compliance plans, source control management, enabling continuous integration, delivery, and release strategies. Students will learn to look beyond the silos found in organisations and create cross-functional teams with solid communication channels. By building a culture of shared responsibilities and transparency, developers can create better products for customers.

Applying Knowledge and Understanding At the end of the module/unit the learner will have acquired the following skills: a) Use DevOps and Agile methodologies in existent projects b) Create effective project management plans c) Use the different techniques learnt during the course d) Compose new processes that allow developers to benefit from agile methodologies e) Demonstrate the efficacy of DevOps and convince team members to adopt them f) Manage diverse teams in an effective and transparent way g) Analyse the progress of various projects and make appropriate adjustments h) Generate detailed execution plans with the support of all team members
Module-Specific Learner Skills At the end of the module/unit the learner will be able to a) Explain the benefits of DevOps and Agile methodologies b) Define and describe how the various techniques are being used within the context of complex software development c) Design and propose complex plans
Module-Specific Digital Skills and Competences At the end of the module/unit, the learner will be able to a) Gain knowledge in the use of a real platform such as Microsoft Azure and AWS

Back to the Programme Structure

Game Development
Elective
6 ECTS
Term 5

Course Description

Game development is a multi-disciplinary field of study based on themes from computer science, artificial intelligence, mathematics, creative writing, art, audio, and others. It deals with the entire life cycle of a game, starting from conception, idealisation, design, programming, testing and up to the final release.

A digital games developer is slightly different from a normal software developer because they have to contribute to the game design and production and must keep in mind that games function in a totally different way. Video games tend to have a game loop dictated by the performance of the platform being used, and in that loop, a game programmer needs to apportion various calculations, including AI algorithms. Most of the time, they do not have the luxury to finish those calculations in one cycle and thus, they have to be spread over various. This is done to ensure that the game response occurs in real-time, a requirement which is not always vital in normal software development. Furthermore, since the development occurs for multiple platforms simultaneously (PC, console, mobile, etc.), the level of care required is much bigger.

In the past years, irrespective of global instabilities, game development experienced massive growth. So the module will look at the sector both from an independent game development perspective and also from the viewpoint of a massive AAA game company. It will teach students how to make games using one of the leading industry-standard game engines (e.g. Unity). We will start by covering basic game development principles, gaining familiarity with the game engine editors, and delving into developing meaningful user experiences. The module will guide the student towards developing a prototype, how to test it and polish it in preparation for the final release.

Throughout the course, students will explore the following topics:

- Create 3D models using Autodesk Maya
- Mapping textures
- Integrate 3D models in Unity
- Game design in Unity: Interactions and user interfaces
- Special effects
- Visual scripting for interactive content
- Explore different game mechanics
- Use Artificial Intelligence in games
- Handle game physics
- Animate characters

Applying Knowledge and Understanding

At the end of the module/unit, the learner will have acquired the following skills:

a) Practice game development using an industry-standard game engine, specifically on:

a. The game engine's interface
b. Use of Game objects
c. Scene creation
d. Managing assets
e. Scripting
f. The Physics Engine
g. Lighting
h. Materials, textures and shaders
i. Audio manipulation
j. Animations
k. Artificial Intelligence
l. Optimisation
m. Deployment

b) Design the game mechanics of a game

Module-Specific Learner Skills

At the end of the module/unit the learner will be able to

a) Proficiently use one of the world-leading game development engines to create production-quality games
b) Create scripts for different purposes such as game loops, controlling non-player characters, networking, etc.
c) Demonstrate knowledge of game development terminology
d) Plan, assemble and distribute work
e) Apply the feedback received to improve the game under development
f) Create an indie game and deploy it to multiple platforms
g) Apply the knowledge gained to other industries (such as the entertainment sector, simulations, serious games)
h) Create a game design document, playtesting plans and quality assurance documentation

Module-Specific Digital Skills and Competences

At the end of the module/unit, the learner will be able to

a) Learn to use one of the commercial game engines

Back to the Programme Structure

Thesis
Compulsory
30 ECTS
Term 6

Course Description

The final-year project is the most significant single work assigned to students throughout their degree program. It is intended to consolidate the skills gained during the degree program and prepare students to undertake and participate in projects following graduation. Each student, together with an assigned supervisor, will have discussed and developed the title, focus, and objectives of a thesis to be undertaken before proceeding with the work and study required for its completion.

In most cases, the thesis is the longest and most challenging assignment given to a student, requiring an entire term of preparation and hard work. The supervisor's role is to guide the student since most of the thesis should be independent work. The quality of the work expected should provide enough depth within a particular topic and answers typically a specific research question which is agreed upon a priori with the supervisor. Although at this level, one does not expect novelty, innovation distinguishes between an excellent and good thesis. At the end of the process, the student would have learnt to conduct independent research, problem-solving, numerical mastery, project management, time management, and self-discipline, amongst others.

Students will have the opportunity to conduct internships with industrial partners as a way to work and complete their thesis.

The thesis will be presented to a committee, composed of an internal examiner, an external examiner, and the supervisor, where the student will be expected to defend the work done, and the results presented. This happens typically via an oral examination called a viva, where the student presents their work and answers questions from the committee.

Applying Knowledge and Understanding

At the end of the module/unit the learner will have acquired the following skills:

a) Learn how to construct a comprehensive literature review
b) Provide a solution to a particular problem, putting it in a particular context of study
c) Articulate the criteria that describe the adequacy of a solution, design and develop a solution
d) Assess the solution with respect to the identified criteria
e) Produce a well-structured report describing and scientifically substantiating the approaches, concepts and techniques used and the results obtained
f) Practice the use of core soft-skills such as time-management, project-management, presentation skills, etc.

Module-Specific Learner Skills At the end of the module/unit the learner will be able to a) Apply to major situational contexts formal and informal knowledge b) Apply the knowledge and methods acquired during the course of studies c) Make use of different modes of communication, according to the different contexts in which students are exposed
Module-Specific Digital Skills and Competences At the end of the module/unit, the learner will be able to a) Perform extensive online searches through various digital repositories b) Analyse data using statistical packages or other advanced mathematical tools c) Use digital presentation tools such as Powerpoint

Back to the Programme Structure