STUDIO

Paper Code	**ID00000**
Lecturer(s)	**Various**
Semester	**Semester 1 and 2**

Summary

Studio is a thread through the whole degree made up of 15 credits each semester. In Studio, learners will be gradually introduced to the tools and workflows used in the IT industry.

Learners will practice their technical skills in a workplace-like environment by undertaking team-based, curated project work that has been specially designed to target specific skill areas at the right level for each year.

Studio also comprises several taught workshops each semester, starting with more taught material in first year, designed to teach the skills required to stick everything together.

The Studio course follows a loose theme:

- **Studio 1:** Fundamentals and teamwork
- **Studio 2:** User-centric workflows
- **Studio 3:** Agile and Scrum
- **Studio 4:** Quality & SRE
- **Studio 5 and 6:** Capstone projects

PROGRAMMING 1

Paper Code	ID501001
Lecturer	Krissi Wood
Semester	Semester 1 and 2

Summary

The aim of this course is to introduce students to the concepts of program design and programming fundamentals.

We begin by creating programs using basic programming constructs and simple data structures. Then we move on to implementing routines as functions and as procedures and apply common methods to access files within an application.

This course has no prerequisites and does not require any previous programming experience. We will also cover the following:

- Program Design
- Algorithms
- Structured diagrams UML
- If statements, Nested Ifs, Switch statements
- Loops
- Arrays
- Data Types
- Reading Files of Records
- Version control
- Good programming practices that are independent of the language or model used.

PROGRAMMING 2

Paper Code	**ID511001**
Lecturer	**Grayson Orr**
Semester	**Semester 1 and 2**

Summary

The aim of this course is to build simple Object-Oriented applications.

We begin by building interactive, event-driven GUI applications using pre- built components. We then declare and implement our own classes. There are two assignments where learners build their own versions of classic games.

This course requires learners to have passed ID511001 Programming 1. We will also cover the following:

- Problem analysis and decomposition.
- Principles of good class design.
- Logic of basic algorithms.
- Use of core complex data structures.
- Good programming practices that are independent of the language or model used.

FUNDAMENTALS OF WEB DEVELOPMENT

Paper Code	**ID512001**
Lecturer	
Semester	**Semester 1 & 2**

Summary

In this paper, students will be introduced to industrially relevant client and server- side technologies for web-based development.

You will use fundamental components of web pages and basic client/server communication; develop simple web-based applications using industry relevant client/server-side programming languages; and use industry relevant tools and workflows in the development of web-based applications.

Things you will learn in this course:

- HTML/CSS/JavaScript
- Asynchronous programming (AJAX)
- Modern responsive web design principles
- Server-side programming using Node.js
- Web deployment

There are two assessments in this course:

- Skills Based Assessment: 30%
- Web Project: 70%

INTRODUCTION TO NETWORKS

Paper Code	**ID515001**
Lecturer(s)	**Michael Holtz**
Semester	**Semester 1 and 2**

Summary

Introduction to Networks introduces learners to fundamental networking concepts and technologies. The course covers the basics of network theory and skills needed to implement a simple network.

At the successful completion of this course, learners will be able to:

- Design and build a simple local area network using device addressing schemes and basic network configurations.
- Configure and troubleshoot end-to-end connectivity between local and remote networks using security best practices.

Topics include:

- OSI model
- Types of networks
- Application layer functionality
- Transport layer protocols (TCP/UDP)
- Network Layer protocols (IPv4)
- Layer 3 Addressing and subnetting
- Data link layer concepts and addressing
- Network cabling
- Configuring and testing a network

Assessment comprises weekly chapter exams delivered via the Cisco Network Academy, a subnetting skills assessment, a practical skills-based test, and a final theory test.

As a Cisco Network Academy course this paper is an important step towards gaining the experience and knowledge required to achieve the industry recognised and desired Cisco CCNA certification.

PLATFORMS AND DEVICES

Paper Code	ID520003
Lecturer(s)	Paul Admiraal
Semester	Semester 1 and 2

Summary

The aim of this course is to enable students to use a range of devices, platforms and concepts utilised within the IT industry.

The curriculum introduces the computer hardware and software skills needed to help meet the growing demand for entry-level information and communication technology (ICT) professionals. The curriculum covers the fundamentals of PC technology, networking, and security, and introduces advanced concepts.

We will cover the following:

- Research a topic that has a significant impact on the software testing industry, society and or the software development life cycle
- Installing, configuring, and selecting PC hardware components
- Operating systems installation and maintenance (systems tools)
- Overview of operating systems (mobile, desktop, service, etc.)
- Use a VM & develop basic understanding of virtualisation
- Basic use of transmission protocols (e.g., FTP, SSH)
- File systems
- Backup and RAID systems
- Troubleshooting hardware and software
- Connecting and configuring devices (Bluetooth, Wi-Fi, printers, etc.)/Mounting drives
- Command line proficiency
- Basic network configuration
- 'Embedded' (Raspberry Pi, Arduino, Development platforms)
- Environmental impact of IT
- Identify sustainability issues involved in purchasing, using, and disposing of devices.

MATHS FOR IT

Paper Code	**ID521001**
Lecturer(s)	**Vaughn Malkin**
Semester	**Semester 1 and 2**

Summary

The aim of this course is to introduce learners to a range of mathematical tools, skills and concepts that underpin computer systems.

This course is presented as an entry level course, requires no specific mathematical prior knowledge and seeks to build confidence.

We will cover the following within an IT context:

- Logic circuits
- Decimal and binary number systems
- Vectors and matrices
- Randomness and probability
- Factorials, permutations and combinations
- Check digits
- Random number generation
- Encryption
- Algorithms, iteration, recursion
- Hash tables and hash functions
- Computational complexity

DATABASES 2

Paper Code	ID605001
Lecturer(s)	Krissi Wood
Semester	Semester 1 and 2

Summary

No matter which area of IT you see yourself in, a working knowledge of Databases is essential for the well-rounded IT professional.

This paper aims to introduce students to enterprise-scale database principles and methodologies with emphasis on relational database management systems. Working primarily with MariaDB (a branch of MySQL) and SQLite.

Students can expect to:

- Apply formal relational database theory in the design, development, deployment and use of real database systems.
- Construct appropriate data and database (ERD) model for a specified problem and build the corresponding database.
- Construct and apply syntactically correct database queries using an appropriate query language.
- Identify the need for security controls and implement basic data checking and validation.

ID605 has prerequisites of **ID505** - Studio (or Introduction to Databases) and **ID511** - Programming 2. You must talk to the lecturer if you do not meet these requirements.

INTRODUCTORY APPLICATION DEVELOPMENT CONCEPTS

Paper Code	**ID607001**
Lecturer(s)	**Grayson Orr**
Semester	**Semester 1 and 2**

Summary

In this paper, you will design and build usable, secure & attractive applications with dynamic database functionality following an appropriate software development methodology. This course is taught using Node.js, Express, React and PostgreSQL.

Things you will learn in this course:

- How to create a Node.js REST API application.
- How to create a React application that consumes your Node.js REST API application.
- Interact with a PostgreSQL database using an ORM.
- How to decouple your React application from your Node.js REST API application.
- Deploy your applications to a lightweight PAAS.
- Development workflows such as formatting and Git commit message conventions.
- Use GitHub to manage your applications.

It is recommended that you enrol in this course if you are also enrolling in Studio 3.

This course is a pre-requisite for ID608001: Intermediate Application Development Concepts and ID721001: Mobile Application Development.

There are three assessments in this course:

- **Project 1:** Node.js REST API application (30%)
- **Project 2:** React CRUD application (30%)
- **Practical:** Skills-Based (20%)

INTERMEDIATE APPLICATION DEVELOPMENT CONCEPTS

Paper Code	**ID608001**
Lecturer(s)	**Grayson Orr**
Semester	**Semester 1 and 2**

Summary

In this paper, you extend on ID607001 Introductory Application Development Concepts. This course emphasises writing well-organised and tested code. This course is taught using Node.js, Express, Next.js and PostgreSQL.

Things you will learn in this course:

- How to create a complex Node.js RESTful API application. This application will include authentication, authorisation/access control, caching, optimisation, API testing and API integration.
- How to create a complex Next.js application. This application will include server-side rendering techniques, state management, component testing.
- Extend your knowledge of development workflows such as linting and Git hooks.
- Web security

This course is a pre-requisite for ID730001: Advanced Application Development Concepts and ID711002: Advanced Algorithms (non-games pathway learners only).

There are three assessments in this course:

- **Project 1:** Node.js RESTful API application (45%)
- **Project 2:** Next.js application (35%)
- **Practical:** Skills-Based (20%)

OPERATIONS ENGINEERING 1

Paper Code	**ID609001**
Lecturer	**Dr Olayinka Adeleye**
Semester	**Semester 1 and 2**

Summary

This course will introduce system administration using different cloud platforms. It starts with an introduction to PowerShell and gradually delves into managing basic system operations in Azure/AWS with scripting. It helps students to solidify the tools and techniques learned in the previous ID616 Operating Systems Concepts and helps them transition into the next level Operations Engineering, Security, and Virtualization courses.

It will be difficult to have a holistic view of system operations/services in different infrastructures without taking this course. It will also significantly strengthen the participants to have a smooth start in 3rd year DevOps Security projects and beyond. Since most organizations are running their services/ infrastructure in the cloud, this course will be the first step to learning about major cloud services and resources.

Selected topics:

- Configuring and deploying cloud services and resources
- Advanced user/group management and permissions
- Advanced scripting for system administration
- Backup and disaster recovery
- Directory services in different OSs
- Configuring and deploying cloud services and resources
- Hardening operating systems and networks
- Containers

What is assessed?

The paper includes in-class quizzes, lab-based assessments, and assignments. The networking and scripting concepts learned in the previous IN616 Operating System courses will be further expanded for assessment. In addition, assessments will be carried on for new concepts like containers, cloud, and directory services.

Why would you want to take this course?

This paper will help students to solidify the tools and techniques learned in the previous ID616 Operating Systems Concepts and help them transition into the next level of Operations Engineering, Security, and Virtualization courses.

Note: This paper is a prerequisite for the Virtualization (720), Security (724), and Operations Engineering (734) papers offered in the 3rd year of the degree.

SWITCHING, ROUTING AND WIRELESS ESSENTIALS

Paper Code	**ID615008**
Lecturer(s)	**Michael Holtz**
Semester	**Semester 1 and 2**

Summary

In Routing and Switching Essentials we study the architecture and operation of routers and switches in a small network. This is a more applied course than first year (ID515) with a greater number of labs involving device configuration.

By the end of this course, you will be able to deploy services such as DHCP and NAT, configure and troubleshoot routers and switches and resolve common issues with routing protocols, virtual LANs, and inter-VLAN routing in both IPv4 and IPv6 networks.

Topics include:

- Switch security
- Virtual Local Area Networks (VLANs)
- Inter-Vlan routing
- Static Routing
- Dynamic Routing Protocols (RIP)
- Access Control Lists (ACL)
- Dynamic Host Configuration Protocol (DHCP)
- Network Address Translation (NAT)

Assessment comprises weekly chapter exams delivered via the Cisco Network Academy, an assignment to plan and implement a small organisation network, a skills based assessment and a final theory test.

As a Cisco Network Academy course this paper is an important step towards gaining the experience and knowledge required to achieve the industry recognised and desired Cisco CCNA certification.

OPERATING SYSTEMS CONCEPTS

Paper Code	**ID616001**
Lecturer(s)	**Rob Broadley**
Semester	**Semester 1 and 2**

Summary

This course will introduce you to the fundamentals of Operating Systems and Systems Administration using Linux. The idea of this course is to take you from a newbie to an advanced user and put you on a pathway for the advanced courses on systems administration, virtualization and security and lead you to the DevOps/Security/Cloud pathways.

We will explore the core system components, and learn how to create users, groups and manage permissions. You will understand how to deal with processes, install software, and automate tasks. In the second term we will look at networking, including basic network configuration, setting up network services as well as exploring how we can use Linux to provide basic network security.

Selected topics:

- Operating System Basics
- Process and Threads
- Linux architecture, shells, file systems, devices and processes
- Bash commands and scripting
- Linux system administration
- Management of Linux task scheduling, packages and services
- Troubleshooting system and performance issues
- Customizing Linux Kernel
- Linux network configuration and troubleshooting
- Linux security

What is assessed?

Since the course primarily looks at developing skills, most assessments will be skills-based. This includes classical skills-based assessments, assigned labs, and exercises to increase your confidence with the command line and standard Linux tools you find on any Linux machine.

Further to this you will need to write scripts that utilize advanced system commands and build confidence with bash, a language you are likely to encounter in your future career. In addition, you will have the opportunity to explore topics outside the course curriculum and present those to the class. The final assessment is a written exam that builds on the many skills you have developed in the course.

Why would you want to take this course?

This course provides you with the skills to comfortable navigate and administer Linux systems, an aspect that is important for developers who want to work with open-source software or have to deploy their own code (an increasing shift from the traditional role separation of programmer and administrator).

Beyond the programming perspective, this course provides the foundation for backend systems administration roles, which makes it a natural prerequisite for the 2nd year 2nd semester (ID609) Operations Engineering 1, 3rd Year (ID719) Operations Engineering 2, ID720) Virtualization (ID720) and (ID724) security papers. This course provides the minimum level of exposure to Linux one should display when completing a BIT degree.

Prerequisite knowledge: An understanding of basic programming concepts at a “101” level. Familiarity with using command-line and/or terminal interfaces. Familiarity with using keyboard navigation to access and control a computer. Basic knowledge of computer hardware and components. Basic knowledge of networking. Familiarity with virtual machines.

EMBEDDED SYSTEMS

Paper Code	**ID620001**
Lecturer(s)	**Vaughn Malkin**
Semester	**Semester 1**

Summary

Embedded systems is a popular, practical subject in the second year of the BIT. It regards computing from the point of view of an intelligent microprocessor embedded in a device or gadget. Of course, we have to learn how to write programs that are burned into the microprocessor and then respond to real world events and signals like temperature, position and light as well as human actuated events like buttons, switches and joysticks.

The language we use is gcc C++ which is large, powerful and free. The hardware platform is the Ardunio which has a huge collection of online resources. We supply a basic kit for you to work on, and you may buy extra components if you wish.

Assessment is mostly practical, project based and a lot of class time is spent tinkering with real gadgets with negotiated functionality. **This course suits self-motivated independent people** who what to get to the bottom of the binary instructions that are the basis of all computing.

Prerequisite knowledge: Basic experience at programming in a C-family language, or an understanding of basic programming concepts at a “101” level in another language. No previous electronics or electrical knowledge required. Self-starter.

AUTOMATION AND ROBOTICS

Paper Code	**ID621001**
Lecturer(s)	**Vaughn Malkin**
Semester	**Semester 2**

Summary

This paper builds on the background students have developed in the Embedded Systems (ID620) paper and exposes you to the themes of task automation and robotics. The course is 100% project based.

We begin with an initial review of basic electronics and some simple electrical theory. We quickly move onto simple circuit fabrication, using multi-meters, prototyping software, and soldering before we look at fabrication technologies like 3D printing and some basic CAD. We will touch on mechanical and physical design considerations useful in robotics.

In the second part of the course we will look at the principles of robotics and the challenges of operating in a physical environment. You will build your own robot that navigates and interacts autonomously using its sensing and communication capabilities.

Selected topics

- Electrical theory and components (resistors, capacitors, relays, etc.)
- Sensors, Motors
- Soldering
- Behavior-based robotics

What is assessed?

You will be assessed on your ability to development and implement the hardware and software for a given robotics platform and include the functional assessment of the robot behaviour (ie: whether your robot does what it is supposed to do).

You will also be assessed on your ability to make practical use of software and hardware tools and techniques: 3D printing, CAD, soldering, circuit board creation, among other things.

Prerequisite knowledge: You have passed Embedded Systems OR you have previous experience working with simple electronics (Through-hole components, breadboards). You are familiar with the Arduino microcontroller eco-system, the use of simple sensors, reading and processing sensor data. You are familiar with the basics of C-family languages.

Why would you want to take this course?

This course is for you if you have completed the Embedded systems course and want to continue developing your electronics skillset building meaningful applications that build on sensing and automation technology.

This paper will further introduce you to principles of robotics (i.e. acting in a physical environment) and associated challenges of interfacing hardware to software, functioning in the unpredictable environment of the "real world".

This paper forms a prerequisite for any student contemplating hardware projects in the capstone project of the 3rd Year.

INTRODUCTION TO ALGORITHMIC PROBLEM SOLVING

Paper Code	**ID630151**
Lecturer(s)	**Grayson Orr**
Semester	**Semester 1**

Summary

To allow learners to carry out semi-independent exploration into a specific information technology topic to a depth not supported by an existing Bachelor of Information Technology paper. In this paper, students will be introduced to common programming algorithms, and apply them to situations to solve problems. The context of this paper will be games programming and scripting, and the application of algorithmic problem solving in developing games.

Things you will learn in this course:

- C# scripting
- Game loops
- Pathfinding, procedural generation, trees, graphs
- Basic AI
- Functional programming (filtering, sorting)
- FSM
- Events

Prerequisite knowledge: A basic understanding of C# programming (variables, loops, conditionals) is expected. No specific Unity or game development knowledge required.

DEVELOPING FLEXIBLE IT COURSES

Paper Code	**ID703001**
Lecturer(s)	**Elise Allen / Vaughn Malkin**
Semester	**Semester 1**

Summary

This course aims to prepare students for the training role that is often performed by information technology professionals. Anyone considering a technical support or help desk role will almost certainly be called upon to train users either face to face or by writing training modules.

In this course we learn how to identify the training requirements associated with a new development. We learn to prepare, conduct and evaluate appropriate training sessions in a face to face and in an online setting.

You will be required to teach a small group of students face to face and create an online training module.

ID703 requires strong reading, writing and presentation skills.

DATABASES 3

Paper Code	ID705001
Lecturer(s)	Krissi Wood
Semester	Semester 2

Summary

Databases 3 continues the work begun in Databases 2, providing the opportunity to build enterprise level databases in Microsoft SQL Server. Students can expect to greatly increase their skills in complex Queries, Triggers and Procedures, primarily using Transact-SQL.

We will delve into the world of the DBA in a containerised environment looking at indexing, replication, backups, restore and query optimisation.

Students who complete ID705 should have the database skills necessary to contribute confidently to teams engaged in “full stack” development, the heart of which is the database.

Taken in conjunction with other 3rd year software development papers, ID705 prepares students to apply for internships or graduate positions in software development (particularly database application development) and sets up graduates for further in-work training as a development or production DBA.

Prerequisites: ID605

ADVANCED ALGORITHMS

Paper Code	**ID711002**
Lecturer(s)	**Dr David Rozado**
Semester	**Semester 2**

Summary

IN711 teaches the advanced computational concepts that every BIT student planning to become a competitive professional software developer should master. This is because BIT graduates working as software developers will often need to work alongside computer science graduates and communicate and collaborate efficiently with them.

OP BIT graduates will also compete with computer science graduates for promotions, contracts, etc. BIT graduates from OP tend to have stronger IT practical skills than computer science graduates from university. Computer science graduates however frequently have an edge on the theoretical foundations of computational thinking.

The goal of IN711 Advanced Algorithms is to bridge the computer science gap between BIT students and computer science students. After successfully completing the course, BIT graduates will be able to use the language and conceptual tools of computational theory to communicate and work efficiently with computer science graduates in a level playing field.

Topics covered include: Computational modelling, Genetic algorithms, AI, cryptocurrencies, recommender systems, industrial control, numerical optimization, searching, sorting, compression, Monte Carlo simulations, constraint satisfaction, stochastic algorithms, trees, hashes, graphs, cryptography, Bayesian inference, recursion, runtime analysis, dynamic programming, Fourier transform, signal processing, Blockchain, Brain-computer interfaces, smart contracts and other advanced techniques.

After completing IN711, a student should be able to communicate, collaborate and compete efficiently with computer science graduates and confidently apply for graduate software development positions. Prerequisites: Familiarity with object-oriented programming and discrete mathematics (i.e. ID521 Maths for IT).

ADMINISTERING A VIRTUAL INFRASTRUCTURE

Paper Code	**ID720001**
Lecturer(s)	**Dr Olayinka Adeleye**
Semester	**Semester 2**

Summary

This course deep dives into the building blocks of modern virtualized infrastructures (i.e cloud). Students will be able to appreciate how virtual machines are constructed using different virtualization technics (i.e hypervisors). It also includes exercises that show how to deploy modern application infrastructure using OpenStack and AWS/Azure platforms. Building upon the skills learnt through Operations Engineering 1 and 2 papers, this course goes further into container orchestrations using Kubernetes and deploying microservices in cloud.

Selected Topics

- Install and configure a virtual environment
- Deploy virtual machines and virtual applications in cloud platforms
- Management techniques and key performance metrics used to identify CPU, network, memory and application performance bottlenecks in a virtualized environment.
- Container orchestration using Kubernetes
- Troubleshooting hypervisor problems
- Establish, monitor and troubleshoot service levels for enterprise cloud infrastructure
- Critically analyse key performance factors in virtualized systems

This paper comprises three sections:

Containers

Containers are a lightweight form of virtualization that has been adopted rapidly in the last few years. Now they are commonly used for both development and deployment of services in a wide range of settings. We will see what containers are, how to create and use them, and how to orchestrate the management of collections of containers to deliver a service.

Public Clouds

We can think of *clouds* as unified platforms to create and manage virtual machines. The most common way that IT organisations use clouds is through hosted cloud service providers, or *Public Clouds*. In this section we will see

how public cloud services are used. Much of the focus of this section will be on using public cloud services through a remote API.

Self-Hosted Clouds

Although public cloud services are more widely used, there remain use cases for *self-hosted clouds*. In addition, working with them provides an opportunity to explore some key topics in virtualization. This is a huge and complicated topic, but in this section we will set up and operate our own cloud systems.

There is an assessed activity in each for the three sections above. There is also a final theory examination.

MOBILE APPLICATION DEVELOPMENT

Paper Code	**ID721001**
Lecturer(s)	**Grayson Orr**
Semester	**Semester 2**

Summary

In this paper, you will learn the specifics of mobile application design and development.

This course has three pathways:

- Native mobile development using Kotlin and Android Studio
- Cross-Platform development using TypeScript and Expo
- Game development using C# and Unity

You will be able to develop and publish mobile applications or a mobile game to Google Play Store.

Things you will learn in this course:

- Core developer topics
- View system
- Model-View-ViewModel architectural pattern
- Accessibility
- Dependency injection
- Performance
- Security
- Policies and procedures

ADVANCED NETWORKING

Paper Code	**ID723001**
Lecturer(s)	**Michael Holtz**
Semester	**Semester 1**

Summary

In Advanced Networking we look at designing and building redundant, scalable and private networks on a larger scale. Importantly we study how these networks can connect to each other using policy to control information exchange and path preference. Lastly, we look at how service providers provide customer separation and the options for customer connectivity.

While we will primarily be using Cisco platforms including Cisco's Virtual Internet Routing Lab (VIRL) it is important to note that the majority of technologies covered in this course are standards based and are implemented similarly across most network vendors.

Topics include

- Border gateway protocol (BGP)
- Virtual route forwarder (VRF)
- Link aggregation
- Hot standby routing protocol (HSRP), gateway load balancing protocol (GLBP)
- Multi area open shortest path first (OSPF)
- IPSec, GRE virtual private networks (VPN)
- Multi protocol label switching (MPLS) virtual private networks (VPN)

This paper is not based on a Cisco Netacad course. Assessment comprises a two-phase project a research assignment a final SBA and a final written test.

Students who complete the course will understand how to evaluate and apply several advanced networking protocols, services and concepts to the design, deployment and interconnection of variable scale networks. The technologies covered are important requirements for industry recognised and desired Cisco certifications such as CCNA and CCNP.

SECURITY

Paper Code	ID724001
Lecturer(s)	Dr Olayinka Adeleye
Semester	Semester 2

Summary

This course offers a systematic introduction to the topic of information security. Many papers refer to security as an important aspect, but in this paper, you will get a better understanding of the different ways in which systems are exploited – and how to prevent it from happening.

We will determine potential vulnerabilities (weaknesses) and learn how to mitigate these threats in a variety of scenarios. In the final stages of the paper, we will also investigate computer forensics – methods to determine and reconstruct digital events. Throughout the paper, we adopt an attack and defend scenario. Usually, the first session of the week involves analyzing vulnerabilities and exploiting systems by thinking like an attacker. As a result of this course, you will have a broad understanding of basic information security best practices.

Specific topics

Basic information security principles

- Confidentiality, data integrity, authentication
- Protecting information and communication using cryptography (encryption, hashing, digital signatures)

Network Security

- Information gathering (e.g., banner grabbing, port scanning)
- Vulnerability scanning and exploitation
- Protecting systems using robust firewall protection
- Server hardening (e.g., securing services)

Web application security

- Cross-site scripting (XSS)
- SQL Injection
- Secure password storage
- Password hashing and password cracking
- Security audits/pen tests
- Ethical and legal considerations

What is assessed?

Since the topic of security is moving rapidly, the assessments are primarily based on lab work in which you explore different types of security problems hands on. This includes lab worksheets for every session. You will also be tasked with two assessments:

1. Securing a vulnerable web application by fixing a variety of common web application vulnerabilities.

2. A full security audit of a vulnerable server. You will also sit a final exam to assess your overall knowledge developed throughout the course.

Why would you want to take this course?

If you want to be a coder and write better code and know the weak links of your system and produce reliable systems, this course will provide you the unique opportunity in the curriculum.

Security is equally (if not more) important if you want to work as a system administrator. Here your focus would lie on testing your systems for security holes, keeping up to date with the latest developments and introduce reliable maintenance strategies to keep your systems online.

The paper further provides you with a basic foundation if you plan to concentrate on consulting in the area of security.

Disclaimer

We could fill an entire curriculum with security topics, so this course provides you with an entry-level understanding. Becoming a security expert would require further, more specialised training.

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

Paper Code	**ID727001**
Lecturer(s)	**Dr David Rozado**
Semester	**Semester 1**

Summary

Artificial intelligence is a branch of computer science trying to mimic human Intelligence in machines. Artificial Intelligence (AI) technology is increasingly prevalent in our everyday lives, albeit often invisibly, as advances in computational power and theoretical breakthroughs allow for the emergence of new applications.

This course aims to provide a broad introduction to Machine Intelligence and Data Science with an emphasis on the intuition and the applications behind the concepts. Students will be able to analyse a machine intelligence problem and based on a reasoned argument choose and deploy the appropriate machine learning tool to solve the problem. The end result should be useful/actionable information from the data or an autonomous system that shows “intelligent” behaviour.

Topics covered in ID727 include:

- Computer vision, Natural language processing
- Data Science and Big Data
- Artificial neural networks/Deep learning
- Dimensionality reduction
- Transfer learning
- Generative models/Synthetic Content Generation (Language, images, etc.)
- Anomaly detection
- Transformers
- ChatGPT

Possible applications of course content are: object recognition in digital images, autonomous navigation (e.g., self-driving cars), object tracking, speech recognition, weather forecast, automatic medical diagnosis, natural language understanding, content categorization, text generation, chatbots, document summarization, synthetic image generation, machine translation, biometrics, sentiment analysis and a myriad of others.

After completing ID727, a student should be able to confidently work on machine learning projects in industry. Prerequisites: Familiarity with object-oriented programming and discrete mathematics (i.e. ID521 Maths for IT).

UX ENGINEERING

Paper Code	**ID729002**
Lecturer(s)	**Elise Allen**
Semester	**Semester 2**

Summary

UXE is about building the “front of the front-end” and requires an understanding of both client-side programming and user experience concepts.

In this course students will be introduced to a broad range of UX concepts including:

- User interface design
- Interaction design
- Usability Heuristics
- UX research and testing
- Accessibility
- Responsive design

Technical front-end web skills such as:

- CSS and HTML maintainability
- Modern front-end tools and paradigms
- Advanced semantic markup
- Component-driven development

UX engineers require the ability to speak the language of UX design, but also have the technical skills to make it happen. In order to succeed in this course, you will already need a sound understanding of and experience in HTML, CSS and JavaScript. Ideally you will also have knowledge of a modern front-end framework which you will build upon during the semester.

What you will be doing:

In this course you will complete a number of practical tasks that involve building front-end solutions to specific UX problems. The course will culminate in a larger project to put all the concepts together in order to solve a non-trivial UX problem in a thorough and high-quality software solution. **NOTE:** there is reading, writing and programming involved!

Prerequisite knowledge:
You will need to start with strong web development skills* (e.g. working with JSON, fetching from REST APIs, good HTML and CSS etc), preferably in a Node.js environment,

as well as the ability to work with Git and GitHub proficiently. Research and referencing skills are also needed.
*except BIT Game Dev pathway

ADVANCED APPLICATION DEVELOPMENT CONCEPTS

Paper Code	**ID730001**
Lecturer(s)	**Martin Junek**
Semester	**Semester 1**

Summary

In this paper, you will design and build usable, secure & attractive applications with dynamic database functionality following an appropriate software development methodology. Current technology stack used in this paper is Progressive Web App using ReactJS and Firebase.

Things you will learn in this course:

- Building flexible apps that work on most modern platforms
- Why and how to use document data store efficiently
- Offline functionality
- Security in a multi-user environment

There are two assessments in this course:

- Checkpoints during semester
- Project: multi-player game

QUALITY ASSURANCE AND SOFTWARE TESTING

Paper Code	**ID733001**
Lecturer(s)	**Paul Admiraal**
Semester	**Semester 2**

Summary

This course takes a broad approach to Software Testing. It is intended mainly to lay the foundation for a potential career in the IT industry as a Test Analyst but also as a means for programmers to improve the quality of their code by experiencing common pitfalls.

We will cover the following:

- Research a topic that has a significant impact on the software testing industry, society and or the software development life cycle
- Produce automated and flexible test plans
- Detect errors in a variety of contexts both familiar and unfamiliar
- Write clear and professional test results to enable developers to reproduce and fix errors for a significant piece of software
- Practice static vs dynamic techniques
- Write test plans, cases, and specifications
- Carry out White Box and Black Box Testing

As part of this paper, you will be studying for the ISTQB Certified Tester Foundation Level (CTFL) exam. This is an internationally recognised qualification which is a great addition to your CV.

OPERATIONS ENGINEERING 2

Paper Code	**ID734001**
Lecturer(s)	**Dr Olayinka Adeleye**
Semester	**Semester 1**

Summary

In this paper, we move beyond managing individual machines and focus on the management of systems consisting of multiple servers that offer a diverse set of services. The intent of the paper is to prepare you for DevOps/System Operations jobs in the industry and thus has a strong practical focus with a focus on providing reliable professionally managed services. As part of this paper, you will work as a team with fellow students to configure and maintain a set of servers, incrementally deploy additional services and manage those throughout the course using industry standard ticketing, configuration management and monitoring tools.

Selected topics:

- Best practices in systems administration
- Effective teamwork
- Software configuration systems
- Service monitoring solutions
- Backup solutions

What is assessed?

Since the paper emphasizes the development of industry-relevant skills, assessments are based on your technical abilities, including the setup and provision of services, but also based on your ability to work effectively as a team. In addition, you will perform an individual assessment in the form of a job application that gives you an opportunity to bring your CV up to scratch and an interview that assesses the skills and experience gained throughout the course.

Why would you want to take this course?

This course is for you if you plan to work in the area of system administration in a larger organization, or as a develops operating at the interface of application development and deployment. You will be comfortable managing larger server environments and master the tools you need for this job.

Note that this paper is a prerequisite for the virtualization paper offered in the 2nd semester of the 3rd year of the degree.

GAME DEVELOPMENT

Paper Code	ID737001
Lecturer(s)	Grayson Orr
Semester	Semester 1

Summary

In this paper, you will apply game programming techniques and tools to develop an effective game. A major part of this paper is collaborating with students enrolled in Communication Design to plan, prototype, publish and exhibit a game.

Things you will learn in this course:

- Iterate on game ideas into a final project
- Scripting/programming in Unity
- Develop game mechanics and implement them into the overall project
- Structure large projects in Unity according to industry best practices
- Communicate with non-IT team members (i.e. designers) to achieve team goals

Taking this paper as part of the GAME DEVELOPMENT PATHWAY (exceptions for 2022):

It is expected that students will have successfully completed:

- Studio 3, Studio 4
- 3D Modelling (Communication Design)
- World building (Communication Design)
- DB2
- Special Topic 2 (Introduction to algorithmic problem solving)

And will be enrolled in:

- Studio 5, Studio 6
- Advanced Algorithms
- Mobile Development
- UX Engineering

Taking this paper as an ELECTIVE ONLY:

It is recommended that students will have done one or more of the following papers before enrolling in this paper:

- Programming 2
- Introductory Application Development Concepts
- DB2
- Studio 4
- Special Topic 2 (Introduction to algorithmic problem solving)

There are two assessments in this course:

- Assignment: 30%
- Game Project: 70%

Prerequisite knowledge: A familiarity with Unity and C# programming in general is expected, as well as confident programming fundamentals (variables, loops, conditionals, etc...). It is STRONGLY RECOMMENDED that learners enrolling in this paper have successfully completed Introduction to Algorithmic Problem Solving (L6 Game Dev).