4. Courses Description

A0111101 Mathematics (1)

Functions: Intervals, Inequalities, and Absolute Values, New Functions from Old Functions, Exponential Functions, Inverse Functions and Logarithms, The Limit of a Function; Derivatives: Derivatives of Trigonometric Functions, The Chain Rule, Implicit Differentiations, Derivatives of Logarithmic Functions, Linear Approximations and Differentials, Optimization Problems; Integrals: The definite Integral, The Fundamental Theorem of Calculus, Indefinite Integrals and the Net Change Theorem, The Substitution Rule.

***Prerequisite*: None**

A0311101 Discrete Mathematics {3} [3-3]

Introduction to Propositional Logic: Propositions, Logical operations, Predicates Quantification, Mathematical induction; Basic Structures: Sets, Functions, Sequences, Sums; Relations; Recursion; Counting Techniques; Graphs: Euler Circuit, Hamilton Circuit, Planar Graphs, and Coloring Graph.

***Prerequisite*: None**

A0311201 Introduction to Information Technology {3} [3-3]

Basic elements of computing: programming, computer, program, operating environment, data, file; Number systems: decimal, binary, conversion; Describing problem solution using standard flowcharting notation; Linux basics: basic commands, working with files, working with directories, file name substitution, input/output and I/O redirection; Linux shell: overview, programming tools; User-defined commands and shell variables: command files, variables, integer arithmetic; Passing arguments: $#, $#, ${n}; Decisions: exit status, test command, else, exit, elif, Null, && and || constructs; Loops: for, while, until, breaking a loop, skipping commands in a loop; Git: installation and configuration, basic commands, branching.

***Concurrent:* A0331700 Computer Skills (Remedial)**

A0311202 Introduction to Programming* {3} [3-3]

Sequential execution: program structure, command line arguments, string literals, output, Limerick layout; Program errors: syntactic errors, semantic errors, compile time errors, runtime errors; Types, variables and expressions: string, double and integer types, hard-coded data, assignment statement, arithmetic expressions and associativity, type conversions, parsing input data, integer division, grouping expression terms and long statements layout; Conditional and repeated execution: choice and iteration statements, Boolean expressions, relational operators, program design using pseudo code, lists of command line arguments, comments, standard classes; Control statements nested in loops: declaring variables in compound statements, conditional expression operator; Separate methods: private methods, accepting parameters, void methods, returning value, changing parameters does not affect arguments, local and class variables, logical operators, Boolean type, Boolean variables; Program design concepts: designing data, designing algorithm.

Prerequisite**: A0311201 Introduction to Information Technology**

<u>A0311203 Introduction to Programming Lab {1} [1-2]</u>

A set of 25-35 practical tasks integrated with and supporting concepts presented in the course. These tasks are outlined in the theory course and detailed in separate worksheets that students do individually at their pace during weekly laboratory sessions, using the Java language and its tools. Sessions are controlled via a platform designed to track students' work originality and manage submissions for assessment.

Concurrent**: A0311202 Introduction to Programming**

<u>A0312101 Data Structures* {3} [3-3]</u>

Lists: static allocation, dynamic allocation; Stacks: static implementation, linked implementation, operations, applications; Recursion: applications, program stack; Queues: static implementation, linked implementation, operations, applications; General Trees; Binary Trees; Binary Search Trees: traversal, search, add and delete operations; Files: input, output; Graphs: traversal, adjacency matrix, and adjacency list.

Prerequisite**: A0312201 Object Oriented Programming**

<u>A0312201 Object Oriented Programming {3} [3-3]</u>

Introduction to Object Oriented Programming Concepts using Java language: Classes, Objects, Constructors, Encapsulation: Visibility Modifiers; Packages; Overloading; using **this** keyword; using **static** keyword; Array of objects: Store and Process objects in array; Relationships between Classes: Composition, Inheritance: Superclasses and Subclasses, using **super** keyword, Constructor Chaining, Overriding, Polymorphism, Preventing

Extending and Overriding, The Object Class and its toString() Method; Abstract Classes; Interfaces; Exception Handling; introduction to GUI programming.

***Prerequisite*: A0311202 Introduction to Programming**

A0312202 Object Oriented Programming Lab {1} [1-2]

A set of 25-35 practical tasks integrated with and supporting concepts presented in the course. These tasks are outlined in the theory course and detailed in separate worksheets that students do individually at their pace during weekly laboratory sessions, using the Java language and its tools. Sessions are controlled via a platform designed to track students' work originality and manage submissions for assessment.

***Concurrent*: A0312201 Object Oriented Programming**

A0314501 Ethical and Professional Issues in Computing {1} [1-1]

An overview of ethics, Professional ethics of workers and users in the field of information technology, Cyberattacks and Cybersecurity, Privacy, Intellectual property, Ethical decisions in software development.

***Prerequisite*: Passing 80 credit hours**

A0311301 Digital Logic Design {3} [3-3]

Binary Systems: Digital Computers & Systems Binary numbers, Number Base Conversion: Octal & Hexadecimal Numbers, 1's & 2's Complements Binary codes; Boolean Algebra & Logical Gates: Basic Definitions of Boolean Algebra, Theorems of Boolean Algebra, Boolean Functions Digital Logic Gates, IC Digital Logic Families; Simplification of Boolean Function: Karnaugh Map Method with 3 variable , 4 variable, 5 variable Map. Sum of Products, Product of Sums, Don't care; Combinational Logic: Integrated combinatorial circuits, Sequential circuits, Flip-flops, registers, counters, memory units.

***Prerequisite*: A0311101 Discrete Mathematics**

A0312203 Visual Programming {3} [3-3]

Introduction to Visual Programming; Creating Applications with Visual C#; Processing Data; Making Decisions; Loops; File Access: reading and writing; Random Numbers; Methods; Arrays and Lists; Multiform Projects; Databases.

***Prerequisite*: A0312201 Object Oriented Programming**

A0312301 Computer Organization and Architecture {3} [3-3]

Components of a computer: Performance, Technology and Delay Modeling, Intro to Instruction Set Architecture (ISA) Design: MIPS ISA, Translation of High-Level C Constructs into MIPS, Assemblers, Object Code Generation, Linking and Executable Loading, Run-time Execution Environment; Computer Arithmetic and ALU Design, Digital-Logic Design for Sequential Circuits, Register-Transfer Level Description of Systems; Single-Cycle Datapath and Control; Multi-cycle Datapath and Control: Micro-programming and Hard-wired Control Units; Pipelining: Pipelined MIPS Datapath; Pipeline Hazards: Structural, Control, Data; Hazard Detection and Resolution; Pipelining control; Exceptions Handling; SRAM and DRAM Design, Memory Hierarchy, Cache memory design, Virtual memory.

Prerequisite**: A0311301 Digital Logic Design**

A0312401 Fundamentals of Databases {3} [3-3]

Database Concepts; Database Design Methodologies; Data Modeling using ER and EER; Database Integrity Constraints; Relational Model: Relational algebra, Relational Calculus; Functional Dependencies and Normalization.

Prerequisite**: A0311202 Introduction to Programming**

A0312402 Fundamentals of Databases Lab {1} [1-2]

Introduction and Practice on Structural Query Language (SQL): Creating tables, Querying Data Dictionary, Inserting data, Deletion of data, Updating data, Data retrieval, Limiting selected rows, Single row functions, Group functions, Table Joining commands, Subqueries. A set of worksheets covering these topics are distributed which students do individually at their pace during weekly laboratory sessions, using the SQL language. Sessions are controlled via a platform designed to track students' work originality and manage submissions for assessment.

Concurrent**: A0312401 Fundamentals of Databases**

A0312403 Systems Analysis and Design {3} [3-3]

Introduction to systems development: System development life cycle, System Development feasibility; Development of fact finding methods: Context diagram, Data flow diagram, Decision tables and trees, Data dictionary; Conceptual design: DB design, Normalization; System Implementation: Installation, System conversion, Training, Development Tools, Documentation.

***Prerequisite*: A0312401 Fundamentals of Databases**

A0313101 Algorithms Analysis and Design {3} [3-3]

Introduction: Asymptotic Behavior, O, Omega , Thata notation, analysis of algorithms complexity, proving algorithm correctness with loop invariant, solving recurrences; Sorting: insertion, quick, merge, heap; Advanced Algorithm Analysis and Design: amortized analysis, dynamic programming; Graph: breadth first search, depth first search, Topological sort, minimum spanning tree, shortest path; Advanced data structures: B-trees; String matching: naive, KMP; NP-Completeness: P, NP, NP-Complete classes, proving NP-completeness.

***Prerequisite*: A0312101 Data Structures**

A0313201 Internet Applications Development (1) {3} [3-3]

An overview of the Internet and the World Wide Web (WWW); Hyper Text Markup Language (HTML) to structure web pages; Cascading Style Sheets (CSS) to style web pages; JavaScript (Client-Side) to enhance the user experience: control Statements, operators, functions, arrays; Ajax to build rich webpages: XML, JSON.

***Prerequisite*: A0312401 Fundamentals of Databases**

A0313202 Internet Applications Development (1) Lab {1} [1-2]

A set of practical tasks integrated with and supporting concepts presented in the course. These tasks are outlined in the theory course and detailed in separate worksheets that students do individually at their pace during weekly laboratory sessions, using HTML, CSS, JavaScript, Ajax to design and implement web sites. Sessions are controlled via a platform designed to track students' work originality and manage submissions for assessment.

***Concurrent*: A0313201 Internet Applications Development (1)**

A0313203 Internet Applications Development (2) {3} [3-3]

Introduction to ASP.NET to build Web Application using the MVC pattern: the .NET Framework, ASP.NET & MVC Pattern; URL Routing; Controllers and Actions; Views – Razor Engine; Model Binding; Database Access; Cookies; Sessions; Authentication; Web Services.

Prerequisite**: A0313201 Internet Applications Development (1)**

A0313301 Operating Systems {3} [3-3]

Fundamental Concepts of Operating Systems; Evolution of Operating System; Operating System Structure; Process: Process Management, Inter-process Communication, Process Scheduling, Deadlocks, Process Synchronization; Memory Management; File System Management; I/O Management; Secondary Storage Management; Case Studies.

Prerequisite**: A0312301 Computer Organization and Architecture**

A0313401 Theory of Computation {3} [3-3]

Regular Languages: Deterministic and Non-Deterministic Finite Automata, Regular Expressions, Closure Properties of Regular Languages, Non-regular languages; Context Free Languages and Grammars: Pushdown Automata, Closure properties of Context Free Languages; Turing Machine.

Prerequisite**: A0312101 Data Structures**

A0313402 Advanced Databases {3} [3-3]

Database Management Systems Protection and Security Functions; Views; Transaction Management; Concurrency Control and Serialisability; Database Recovery; Database integrity; Rapid Application Development for Database Systems using CASE tools and 4GLs; PL/SQL Programming.

Prerequisite**: A0312401 Fundamentals of Databases**

A0314201 Mobile Applications Development {3} [3-3]

Android overview: android platform, user interface, dalvik virtual machine, platform architecture, application building blocks, development tools; Building applications: creating and running applications, emulator, activity, android manifest file, layout and layout file, widgets, strings file, intents; Supporting different devices: languages, layout mirroring, screens, platform versions; User interface: user interface fundamentals, linear layout, relative layout, constraint layout; Activity lifecycle: lifecycle call-backs, starting activity, pausing and resuming activity, stopping and restarting activity, recreating activity; Interacting with other activities and applications: intent types, building intent, intent filters, receiving intent, starting another activity, processing intent, common intents; Saving data: internal and external storage, shared preferences, files, SQL database; Dynamic user interface: fragments, fragment types, interacting with other fragments; Web services: REST web services, application dependencies, application permissions, calling using JSON and XML; Publishing android applications: google play, device configurations and languages, APK file and APK package.

***Prerequisite*: A0312101 Data Structures**

A0314202 Mobile Applications Development Lab {1} [1-2]

A set of 15-20 practical tasks integrated with and supporting concepts presented in the course. These tasks are outlined in the theory course and detailed in separate worksheets that students do individually at their pace during weekly laboratory sessions, using the Java and XML languages and its tools. Sessions are controlled via a platform designed to track students' work originality and manage submissions for assessment.

***Concurrent*: A0314201 Mobile Applications Development**

A0314301 Parallel Computing and Distributed Systems {3} [3-3]

Distributed Systems: definition, types, goals, hardware concepts; Communication: layered protocols, remote procedure call, remote method invocation, synchronous /asynchronous persistent/transient communication, stream based communication; Processes: threading in the server side, threading in the client side, code migration; Naming: name Spaces, locating mobile entities, removing unreferenced entities; Synchronization: physical clock synchronization, logical Clocks, global State, election algorithms, mutual exclusion algorithms, distributed transactions; Consistency and Replication: reasons for replication, consistency protocols; Fault Tolerance: introduction, process resilience, reliable client server communication, distributed Commit protocols, recovery.

***Prerequisite*: A0313301 Operating Systems**

A0314401 Programming Languages Design {3} [3-3]

Introduction: language features, compilation stages; Syntax: BNF, attribute grammar; Lexical Analysis: study of simple lexer; Parsing: Recursive Decent, Bottom up parsing with examples; Binding and Scope: static and Dynamic; Types: type checking, strings, enumeration, arrays, records, lists, unions, pointers, generic; Garbage collection: reference Counting, mark and sweep, Expressions: associativity, overloading, conditional, side effects; Control structures: dangling else, composite, scope, breaks, labeled breaks, continue, switch; Subprograms: procedures, functions, call by value, call by reference, activation record, recursion; Object-Orientation: encapsulation, overloading, polymorphisim, dynamic binding, abstract class, nested classes; Functional Programming: program examples with LISP; Logic Programming: example with Prolog.

***Prerequisite*: A0313401 Theory of Computation**

A0314601 Practical Training {0} [8 Continuous Weeks]

The student is required to do practical training in a well-known software company for a period of 8 weeks full time training with at least (15) hours per week. The student is assigned some tasks to perform that are related to his major, such as developing a software, or learning some new skills, technologies and capabilities. At the end of training, reports should be submitted to the department then evaluated by the supervisor.

***Prerequisite:* A0313203 Internet Applications Development (2) + Passing 90 Credit Hours**

A0314602 Graduation Project (1) {1} [1-3]

This course is the first of two courses for the development of significant software system and employing the knowledge gained from courses throughout the program. Students may follow any suitable process model, and must manage the project themselves, following all appropriate project management techniques. Success of the project is determined in large part by whether students have adequately solved their customer’s problem. A project should be performed by a group of students under the supervision of a faculty member. Students are required to focus on the early stages of the software development process such requirements analysis and design, and submit a report. Project must be presented to a committee of the faculty.

***Prerequisite*: A0313203 Internet Applications Development (2) + Passing 90 Credit Hours**

A0314603 Graduation Project (2) {2} [2-3]

This course is the second part of a sequence of two courses (Graduation Project 1 and Graduation Project 2) that entirety constitute the BSc graduation capstone project. In this project, the students will continue the software development and testing of the problem they picked in Graduation Project 1. The project will culminate in the delivery of a working system, a formal public presentation, and written documentation. Oral and written progress reports are required.

***Prerequisite*: A0314602 Graduation Project (1)**

A0111103 Statistics and Probability {3} [3-3]

Data collections; Sampling; Measure of central tendency; Measures of dispersions; Probability: Rules of probability, Counting rules; Discrete random variables; Binomial distributions; Poisson distributions; Normal distributions; Linear regression and correlation; Applications using software packages.

***Prerequisite*: A0111101 Mathematics (1)**

A0111201 General Physics (1) {3} [3-3]

Units Physical Quantities; Vectors; Motion Along a Straight Line; Newton's Lows of Motion: Applying Newton's Laws; Work, Energy and Power; Electromagnetism: Electric Charge, Electric Field, Electric Potential, Capacitance and Dielectric, Current, Resistance, Electromotive Force, Direct Current Circuits, Alternating Current Circuits.

***Prerequisite*: None**

A0111207 Electronic Physics {2} [2-2]

Semiconductor Diodes: Structure, Characteristics, Applications; Bipolar Junction Transistors (BJT): Structure, DC biasing, AC analysis, Applications; Field- Effect Transistors (FET): Structure, DC Biasing, AC Analysis, Applications; Op-Amp: Structure, Inverting and Noninverting Amplifier, Op-Amp Practical Circuits: Unity Gain, Summing, Integrator and Differentiator.

***Prerequisite*: A0111201 General Physics (1)**

A0111208 Electronic Physics (Practical) {1} [1-2]

Oscilloscope and Function Generator Operation; Ohm's and Kirchhoff's Laws; Semiconductor Diode: Characteristics, Half Wave and Full Wave Rectification, Clipper and Clampers, Logic Gates; BJT Characteristics, Transistor As A Logic Gate; FET Characteristics; Op-Amp: Inverting and Noninverting Amplifier; Op-Amp Practical Circuits: Unity gain, Summing, Integrator and Differentiator.

***Concurrent*: A0111207 Electronic Physics**

A0112101 Linear Algebra {3} [3-3]

Systems of linear equations: Homogeneous and non-homogeneous systems ; Matrices: operations on matrices ; Determinants: Operations on Determinates ; Vector spaces ; Method for solving systems of linear equations: Case study ; Eigenvalues and Eigenvectors ; Error norms of vector and matrix ; Applications using software packages.

***Prerequisite*: A0111101 Mathematics (1)**

A0113101 Numerical Analysis {3} [3-3]

Computer Arithmetic's and Error: Case study, applications to real world problems ; Solutions of equations in one variable ; Interpolation and polynomial approximation ; Numerical integration ; Numerical differentiations ; Maple implementations of algorithms studied.

***Prerequisite*: A0111101 Mathematics (1)**

A0332501 Introduction to Software Engineering {3} [3-3]

System Development Methodologies: Software Engineering Processes, Waterfall, Prototype, Incremental, and Spiral, with focus on the Unified Process in its agile form; Principles of Software Engineering: Requirements Elicitation, Validation and Verification; Review of Principles of Object Orientation; Object Oriented Analysis Using UML: Behavioural UML Diagrams Use Case, Sequence, Activity, And State Diagrams; Structural UML Diagrams: Object, Class, and Package Diagrams.

***Prerequisite*: A0312201 Object Oriented Programming**

A0334503 Software Project Management {3} [3-3]

Introduction to project management; The project management and IT context; Project management process groups; Project integration management; Project scope management; Project schedule management; Project cost management; Project quality management; Project communications management; Project risk management.

***Prerequisite*: A0312403 Systems Analysis and Design**

A0341301 Networks and Information Security essentials {3} [3-3]

The course studies the basic of computer networks: types of networks, main devices, Ethernet technology, principles and structure of IP addressing; overview of the common protocols such as: TCP, UDP, HTTP, HTTPS, POP, IMAP, SMTP, DNS, FTP, DHCP; basic security measures and tools: malware, general means of authentication, password-based authentication, physical security, firewall basics; cryptography: symmetric and asymmetric algorithms, hash functions, basics of digital signature and steganography.

***Prerequisite*: A0311201 Introduction to Information Technology**

A0342301 Computer Networks {3} [3-3]

The course studies the architecture, functions, components, and models of computer networks; the main functions and protocols of the application, transport, network, data link, and physical layers of the OSI and TCP/IP layered models are discussed; the principles and structure of IP addressing, purpose and types of NAT, basics of routing, and the fundamentals of Ethernet standard are introduced.

***Prerequisite*: A0341301 Networks and Information Security Essentials**

A0342303 Cybersecurity {3} [3-3]

This course covers some of the main topics of protecting information and information systems such as: access control, web security, email security, social engineering; some of the common vulnerabilities and attacks are also covered: DoS, spoofing, MitM, buffer overflow, CSS; tools related to some of the aforementioned topics will be covered as well; ethical and legal considerations related to cybersecurity are discussed such as: privacy, intellectual property, cybercrime, cyberwarfare, and the organizations involved in the formulation of such laws.

***Prerequisite*: A0342301 Computer Networks**

A0313204 Computer Graphics {3} [3-3]

Graphics Systems and Models: Graphics Programming, Input and Interaction in OpenGL, Geometrical Objects and Transformations in 2D and 3D, homogeneous coordinates, matrix representation, windows and viewports; Viewing in 3D: projections, hidden surface removal; Color: color models, Light, shading and materials, Illumination and Shading, light sources; modeling: geometry processing, rasterization, fragment processing; Clipping: hidden surface removal, antialiasing. Discrete techniques: buffers, bit and pixel operations, texture mapping, compositing; Programmable shaders: OpenGL shading language, fragment shaders, cub and bump maps; Modelling Techniques: trees, scene graphs; Curve and surface representation: Advanced rendering techniques, ray tracing, radiosity, image based rendering.

***Prerequisite*: A0312101 Data Structures**

A0313403 Introduction to Data Science {3} [3-3]

Introduction to data science; The basics of Python; Data preparation; Exploratory data analysis; Preparing to model the data; Introduction to machine learning; Data visualization.

***Prerequisite*: A0312401 Fundamentals of Databases**

A0314203 Artificial Intelligence {3} [3-3]

Artificial Intelligence Applications; Intelligent Agent Systems; Searching Techniques for Problem Solving; Production Systems; Knowledge Representation Techniques; PROLOG; Neural Networks.

***Prerequisite*: A0312101 Data Structures**

A0333506 Human Computer Interaction {3} [3-3]

Introduction to Human Computer Interaction; Principles of Interaction Design; The Human: human abilities, human perceptual system, human cognitive system; Ergonomics; Design, implementation and evaluation of interactive computing system for human use; Components of an interactive system; The computer: interacting with computers, virtual reality concept, virtual reality applications; Non-traditional interfaces; User Interface Design & Programming.

***Prerequisite*: A0312403 Systems Analysis and Design**

A0342304 Introduction to Linux System {3} [3-3]

Fundamental concepts of Linux: file system, commands, utilities, text editing, shell programming and text processing utilities; Linux shells: command line syntax and features, filename generation, redirection, pipes and quoting mechanisms; Navigating the file system: Controlling file access, File and directory naming rules and conventions, Manipulating files and links; controlling the Terminal: Working with vi, Monitoring and controlling processes, Using command line editing, command substitution, using backup commands; controlling processes: print jobs, communicating over the network, Group Policy management and configuration.

***Prerequisite*: A0341301 Networks and Information Security Essentials.**

A0343301 Cloud Computing and Security {3} [3-3]

Introduction to cloud computing: basic concepts and terminology, essential cloud characteristics; cloud service and deployment models: the cloud service models, the cloud deployment models; cloud-enabling technology: multitenant technology, service technology, virtualization technology; fundamental cloud security: basic terms and concepts, cloud security threats

***Prerequisite*: A0342301 Computer Networks**

A0343504 Ethical Hacking {3} [3-3]

This course is an introduction to the principles and techniques of using hacking skills for defensive purposes; topics includes: overview of penetration testing, penetration testing methodologies, gathering intelligence, network scanning, conducting vulnerability scanning, vulnerabilities exploitation, retaining access with backdoors and malware, covering hacking tracks and evading detection.

***Prerequisite*: A 0342303 Cybersecurity**

A0344301 Internet of Things and Security {3} [3-3]

IoT architecture: devices, networking, cloud computing, and big data analysis; Real-time data collection in IoT ; Data analytics in IoT ; IoT applications and requirements ;Security threats and techniques in IoT ; IoT challenges: computation and communication constraints, power constraints, maintenance cost, reliability, data trustworthiness, security, and privacy.

***Prerequisite*: A0313201 Internet Applications Development (1)**

A0314402 Selected Topics in Computer Science {3} [3-3]

The most recent technological topics in computer science that are not covered in other courses in the program plan.

***Prerequisite*: A0312101 Data Structures**