



Al-Ahliyya Amman University
Faculty of Information Technology

Bachelor of Science in Software Engineering

Program Profile

2019-2020



جامعة عمان الاهلية

Al-Ahliyya Amman University



Table of Contents

I. Program Information	2
1. About the Program	2
2. Program Mission	2
3. Program Educational Objectives	2
4. Student Learning Outcomes	3
5. Program Admission requirements	4
6. Program Career Opportunities	5
II. Program Requirements	6
III. Program Structure	6
1. Study Plan	6
2. Advisory Plan	6
3. Advisory Flow Chart	6
4. Courses Description	7
IV. Contact Information	19



I. Program Information

1. About the Program

The Department of Software Engineering was established in 2006 after the Ministry of Higher Education and Academic Research had accredited a program at the Bachelor Degree level in software engineering. The study commenced at the department in the academic year 2006/2007.

Software Engineering (SWE) is a computing science as well as an engineering discipline based on requirement analysis, design, construction, testing, maintenance, and managing the economics of software engineering. It is relatively a recent discipline that applies the principles of computer science, mathematics, and engineering to achieve high quality and cost-efficient solutions to software problems in a systematic, controlled, and effective manner.

2. Program Mission

Provision of education in a creative environment that pace with new developments in the field of software engineering and stimulating scientific research, while meeting quality assurance standards and the requirements of local and global academic accreditation criteria to guarantee high quality output consistent with the ever increasing competencies needed by markets, and to promote cooperation with the local community.

3. Program Educational Objectives

The objective of the software engineering degree program is to prepare students to success as practicing software engineers engaging in lifelong learning and serving in leadership roles in their chosen career paths. The following “Program Educational Objectives” are focused on achievements of our graduates in the initial part of their careers, especially within three years of graduation. The software engineering degree program will prepare graduates to:

- PEO-1.** Acquire essential skills and competencies in software engineering that make them capable of analyzing, designing and developing high-quality software solutions in various application domains under distinctive and realistic requirements.
- PEO-2.** Engage and succeed in their professional life through teamwork, ethical behavior and effective communication.
- PEO-3.** Demonstrate understanding of the importance of professionalism and ethical responsibility at work.



- PEO-4.** Demonstrate understanding of life-long learning and realizing its importance through professional development, practical training as well as professional certification.
- PEO-5.** Occupy influential roles and positions in their organizations and communities.

4. Student Learning Outcomes

The software engineering program applies the IET learning outcomes, categorized under five main learning outcomes:

4.1. Science & Mathematics:

- SM1p: Knowledge and understanding of scientific principles and methodology necessary to underpin their education in their engineering discipline, to enable appreciation of its scientific and engineering context, and to support their understanding of relevant historical, current and future developments and technologies.
- SM2p: Knowledge and understanding of mathematical and statistical methods necessary to underpin their education in their engineering discipline and to enable them to apply mathematical and statistical methods, tools and notations proficiently in the analysis and solution of engineering problems.
- SM3p: Ability to apply and integrate knowledge and understanding of other engineering disciplines to support study of their own engineering discipline.

4.2. Engineering Analysis:

- EA1p: Understanding of engineering principles and the ability to apply them to analyse key engineering processes.
- EA2p: Ability to identify, classify and describe the performance of systems and components using analytical methods and modelling techniques.
- EA3p: Ability to apply quantitative and computational methods in order to solve engineering problems and to implement appropriate action.
- EA4p: Understanding of, and the ability to apply, an integrated or systems approach to solving engineering problems.

4.3. Design:

- D1p: Understand and evaluate business, customer and user needs, including considerations such as the wider engineering context, public perception and Aesthetics.
- D2p: Investigate and define the problem, identifying any constraints including environmental and sustainability limitations; ethical, health, safety, security and risk issues; intellectual property; codes of practice and standards.
- D3p: Work with information that may be incomplete or uncertain and quantify the effect of this on the design.

- D4p: Apply advanced problem-solving skills, technical knowledge and understanding, to establish rigorous and creative solutions that are fit for purpose for all aspects of the problem including production, operation, maintenance and disposal.
- D5p: Plan and manage the design process, including cost drivers, and evaluate Outcomes.
- D6p: Communicate their work to technical and non-technical audiences.

4.4. Economic, legal, social, ethical and environmental context:

- ET1p: Understanding of the need for a high level of professional and ethical conduct in engineering and a knowledge of professional codes of conduct.
- ET2p: Knowledge and understanding of the commercial, economic and social context of engineering processes.
- ET3p: Knowledge and understanding of management techniques, including project management that may be used to achieve engineering objectives.
- ET4p: Understanding of the requirement for engineering activities to promote sustainable development and ability to apply quantitative techniques where appropriate.
- ET5p: Awareness of relevant legal requirements governing engineering activities, including personnel, health & safety, contracts, intellectual property rights, product safety and liability issues.
- ET6p: Knowledge and understanding of risk issues, including health & safety, environmental and commercial risk, and of risk assessment and risk management techniques.

4.5. Engineering Practice:

- EP1p: Understanding of contexts in which engineering knowledge can be applied (e.g. operations and management, application and development of technology, etc.).
- EP2p: Knowledge of characteristics of particular materials, equipment, processes, or products.
- EP3p: Ability to apply relevant practical and laboratory skills.
- EP4p: Understanding of the use of technical literature and other information sources.
- EP5p: Knowledge of relevant legal and contractual issues.
- EP6p: Understanding of appropriate codes of practice and industry standards.
- EP7p: Awareness of quality issues and their application to continuous improvement.
- EP8p: Ability to work with technical uncertainty.
- EP9p: Understanding of, and the ability to work in, different roles within an engineering team.

5. Program Admission requirements

Admission requirements of the Bachelor's degree in the Software Engineering program include a High School Certificate with an average not less than 60% or equivalent, in the Scientific,



جامعة عمان الاهلية

Al-Ahliyya Amman University



Information Technology, Industrial, Comprehensive Health Education or Agricultural streams.
Additional requirements apply:

▪ **Holders of a Jordanian High School Certificate**

An applicant who has a certificate from a Jordanian school: The school should be a licensed and recognized by the Ministry of Education of Jordan. The student must provide certified copies of the high school certificate with a detailed transcript of the grades obtained in the course of study. Both documents must be certified by both: the school and the Jordanian Ministry of Education.

▪ **Holders of a High School Certificate outside Jordan**

Students who have obtained their certificates from schools outside Jordan: the applicant must have graduated from a licensed school recognized by the Ministry of Education in that country and must provide certified copies of the high school certificate and a detailed transcript of the grades obtained in the course of study. These documents must be certified in the country of study by the Ministry of Education, the Ministry of Foreign Affairs, and the Embassy of the Hashemite Kingdom of Jordan; or the Embassy of that country in Jordan, the Jordanian Ministry of Foreign Affairs.

6. Program Career Opportunities

The need for highly skilled software engineers is ever increasing in a multitude of application areas. SWE graduates can occupy positions in software and IT industry, including the following:

- Software developer in various application areas
- Software architect
- Software designer
- Software systems analyst
- Software testing engineer
- Software quality assurance engineer
- Software maintenance engineer
- Software project manager

II. Program Requirements

Award of the Bachelor degree of Science in Software Engineering requires the successful completion of 132 credit hours compulsory and elective courses with grade point average not less than 60% or equivalent, as follows:

University Requirements (24) Credit Hours	Compulsory University Requirements	(12) Credit Hours
	Elective University Requirements	(12) Credit Hours
Faculty Requirements	Compulsory Faculty Requirements	(21) Credit Hours
Major (Program) Requirements (87) Credit Hours	Compulsory Major Requirements	(33) Credit Hours
	Ancillary Major Requirements	(45) Credit Hours
	Elective Major Requirements	(9) Credit Hours

III. Program Structure

1. Study Plan

- Attachment (1)

2. Advisory Plan

- Attachment (2)

3. Advisory Flow Chart

- Attachment (3)

4. Courses Description

A0111101 Mathematics (1) {3} [3-3]

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,



Al-Ahliyya Amman University

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.

A0311203 Introduction to 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: A0311202 Introduction to Programming.

A0312101 Data Structures {3} [3-3]

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.

A0312201 Object-oriented Programming {3} [3-3]

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.

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.

A0333501 Software Requirements Engineering {3} [3-3]

Fundamentals of Requirements: definitions, functional and non-functional requirements, requirements engineering process; Requirements Elicitation: difficulties in eliciting requirements, process of requirements elicitation, requirements sources, elicitation techniques; Requirements Analysis: quality standard of software requirements, requirements classification, requirements negotiation, requirements prioritization, integrating risk management with the requirements engineering process; Requirements Specification and Modelling: IEEE and Volare standards for requirements documentation, specify requirements in Use cases; Requirements Validation: requirements validation techniques; Requirements Management: the need for change, requirements traceability, requirements management guidelines..



Prerequisite: A0332501 Introduction to Software Engineering.

A0333502 Software Design {3} [3-3]

Principles and methods of software design with a special focus on Object-oriented analysis & design; Domain modelling; Class and object modelling; Behavioural modelling; design patterns; General Responsibility Assignment Software Principles (GRASP) design principles; Design evaluation and improvement; Refactoring; Light introduction to software architecture; Practice by designing a large program.

Prerequisite: A0332501 Introduction to Software Engineering.

A0333503 Software Design 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 provided languages and tools. Sessions are controlled via a platform designed to track students' work originality and manage submissions for assessment.

Concurrent: A0333502 Software Design.

A0333504 Software Construction and Documentation {3} [3-3]

Static checking; Specification & specification design; Mutability & immutability; Debugging & avoiding debugging; Abstract functions & rep invariants; Interfaces & equality; Concurrency & thread safety; Frameworks and middleware: GWT framework, WSO2 Carbon; Software documentation tools: Git, Gerrit.

Prerequisite: A0333502 Software Design.

A0333505 Software Construction and Documentation Lab {1} [1-2]

A set of 35-40 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 provided languages and tools. Sessions are controlled via a platform designed to track students' work originality and manage submissions for assessment.

Concurrent: A0333504 Software Construction and Documentation.



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.

A0334501 Software Testing and Quality Assurance {3} [3-3]

Definitions, Basic Concepts and an Overview of Software Testing Techniques: software verification and validation, software testing, software quality, relationships of quality assurance and quality control, software testing techniques, a comparison of testing techniques; Black-Box Testing techniques: equivalence partitioning, boundary value analysis, decision tables, state transition testing, use case testing ; White-Box and Experience-based Testing Techniques: static testing techniques, structural testing techniques, experience-based testing techniques; Software Testing Levels, Objectives and Plan: levels of software testing, objectives of software testing, automation of software testing, the test plan, test case fundamentals.

Prerequisite: A0333501 Software Requirements Engineering.

A0334502 Software Testing and Quality Assurance Lab {1} [1-2]

A set of practical tasks integrated with and supporting concepts presented in the Software Testing and Quality Assurance course. These tasks are 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: A0334501 Software Testing and Quality Assurance.

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: A0333501 Software Requirements Engineering.



A0334504 Software Architecture {3} [3-3]

Software architecture: what, why and context; Quality attributes: role, availability, interoperability, modifiability, performance, security, testability, usability; Architectural tactics and patterns: patterns, patterns' catalog, relationship between tactics and patterns; Architecture and requirements; designing architecture: strategy, attribute-driven development (ADD) method; Documenting software architecture: audience, notation, views, building documentation, documenting behavior, documentation and quality attributes; Architecture implementation & testing: implementation, testing; Architecture evaluation: factors, ATAM, lightweight evaluation.

Prerequisite: A0333504 Software Construction and Documentation.

A0334601 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: A0333502 Software Design + Pass 90 Credit Hours.

A0334602 Graduation Project (1) {3} [3-3]

This course is the first part of a sequence of two courses (Graduation Project 1 and Graduation Project 2) that in entirety constitute the BSc graduation capstone project. In this project, the student is expected to develop software for a specific problem by applying previously learned concepts and methods during the course of the project. In this course, the student is typically expected to study the problem, see what others have done, perform the analysis, determine the requirements and suggest/design a solution. The project should be performed by a group of students under the supervision of a faculty member. The project will culminate in a formal public presentation, and written documentation. Oral and written progress reports are required. The project topic may be provided by the faculty or by the students. The topic is subject to the departmental approval.

Prerequisite: A0333502 Software Design + Pass 90 Credit Hours.

A0334603 Graduation Project (2) {3} [3-3]

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: A0334602 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).

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).

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 Modelling, 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 Modelling 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, and Documentation.

Prerequisite: A0312401 Fundamentals of Databases.

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

Introduction: Asymptotic Behavior, O, Omega , Theta 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.

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.

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.

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.

A0333507 Secure Software Development {3} [3-3]

Design: principles, threat modelling, encryption strategy, standardized identity & access management, establish log requirements & audit practices; Secure coding practices: coding standards & conventions, Use safe functions only, Use code analysis tools, handle data safety, handle errors; Manage security risk inherent in the use of 3rd party components; Testing & validation: automated testing, manual testing; Manage security findings: define severity, risk acceptance process; Vulnerability response & disclosure: definition of internal & external policies, definition of roles and responsibilities; Ensure that Vulnerability reporters know whom to contact, manage vulnerability reporters, monitor and manage 3rd party component vulnerability, fix vulnerability, vulnerability disclosure, secure development lifecycle feedback; Planning, implementation & deployment of secure development practices: culture of the organization, expertise & skill level of the organization, product development model and lifecycle, scope of initial deployment, stakeholder management & communications, compliance measurements, SDL process health.

Prerequisite: A0333504 Software Construction and Documentation.



A0334505 Selected Topics in Software Engineering {3} [3-3]

This course aims at introducing the students to a different area of Software Engineering. It will be achieved by exposing them to new tools, techniques, and research in Software Engineering.

Prerequisite: A0333502 Software Design.

A0334506 Component-based Software Development {3} [3-3]

Components: what, why; Current component models: classification criteria, classification of component models; Introduction to EJB (JavaEE); Development environment; Business logic; Data access objects (DAOs); Java persistence API (JPA): entity manager, core entity mapping, validation API, relationships, inheritance, query, JPA/SQL tuning; EJB development scenarios; Session EJB; Configuring EJBs: ENC, JNDI & CDI; EJB/server-side JPA topics; EJB transactions; Web-enabled EJBs with JAX-RS; Web-UI integration; EJB/JavaEE Security and JavaEE Interceptors; JMS; Asynchronous methods; Timers.

Prerequisite: A0333502 Software Design.

A0334507 Web Services Programming {3} [3-3]

Introduction: service-orientation, SOA, layers with services and microservices; Service-oriented analysis and design: REST services and microservices; API & contract design with REST services and microservices, service API & contract versioning with REST service.

Prerequisite: A0334504 Software Architecture.

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.

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.

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.

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)

IV. Contact Information

Dr. Faris M. Taweel

Chairman of Software Engineering Department

Location: Al-Ahliyya Amman University / Amman-Jordan/ Zip-Code (Postal Address): (19328)

Email: taweelf@ammanu.edu.jo