جامعـــة إربـــد الأهليـــة
IRBID NATIONAL UNIVERSITY

Course Description: Software Engineering

Course Name:	Introduction to IT			Course Code:	409100
Hours:	Credit	Theoretical	Practical	Pre-Requisite:	
	3	3	0		

This course gives the basic concepts of computers and information technology, both physical and programmatic, and includes: an introduction to physical and programmatic computer components, counting systems, and methods of data representation. Stages of software development, application software and system software, focusing on foundations and methods of problem solving and algorithm design. Introduction to C ++ programming language and includes program structure in C ++ language, basic data types, arithmetic and logical operations, control structures in addition to previewing and compiling software

Course Name:	Computer Skill (2)			Course Code:	401112
Hours:	Credit	Theoretical	Practical	Pre-Requisite:	401099
	3	3	1		

This course covers the basic concepts of a programming language using C++ and includes Development of major programming languages. Description of sentence structure and their implications, analysis of sentence structure and construction, names of variables, and includes linking, verification of type and sphere of influence. Data types, expressions, data references and control sentence structures.

Course Name:	Programming Language (1)			Course Code:	407212
Hours:	Credit	Theoretical	Practical	Pre-Requisite:	401112
	3	3	1		

The basic skills of writing and debugging code using a common programming language (e.g., the C++ or Java programming language), an integrated development environment (IDE) (e.g., MS Visual (VC++) development studio), data types, arithmetic and conditional operators, control structures, functions, parameter passing by value and by reference and arrays

Course Name:	Calculus (1)			Course Code:	404101
Hours:	Credit	Theoretical	Practical	Pre-Requisite:	
	3	3	0		

Functions: domain, operations on functions, graphs of functions, trigonometric, inverse, logarithmic and exponential functions; inverse trigonometric functions; continuity limits. Derivative: differentiation techniques, chain rule, implicit differentiation; the differences. Rolle's Theory, Key Value Theory; increasing and decreasing jobs; concavity; Maximum and minimum function values, graphs including Boolean functions; Indefinite integral. Fundamental Theorem of Calculus. Integration by substitution. The area between the curve and the x-axis.

Course Name:	**Statistics & Probability (1)**			**Course Code:**	**404131**
Hours:	**Credit**	**Theoretical**	**Practical**	**Pre-Requisite:**	
	3	**3**	**0**		
This course will include descriptive statistics, probability, Axioms of probability, rules of probability, conditional probability, independence. Discrete and continuous random variables, expectations, and probability distributions. Sampling distributions t, Chi square, F, CLT distributions. Score estimation: for mean and variance, difference between two means and percentage of variances, hypothesis testing for small, large, and dependent samples, correlation, simple and multiple linear regression. Quality of fit tests.					

Course Name:	**Modeling & Simulation**			**Course Code:**	**401452**
Hours:	**Credit**	**Theoretical**	**Practical**	**Pre-Requisite:**	**404131**
	3	**3**	**0**		
This course discusses different topics in simulation and modeling, such as the uses, advantages and disadvantages of simulation, types of models, the steps in discrete-event system simulation, statistical models, simple queuing models, random numbers and random variates, input modeling, model verification and validation, and its use in input-output analysis. Sample implementations for queuing system simulations are discussed using selected languages.					

Course Name:	**Programming Languages Design & Implementation**			**Course Code:**	**401452**
Hours:	**Credit**	**Theoretical**	**Practical**	**Pre-Requisite:**	**407212**
	3	**3**	**0**		
This course will acquaint the student with the fundamental ideas surrounding the design and implementation of high-level programming languages. The course will stress underlying theoretical concepts as well as a significant, practical course project. At the same time, the course will focus on making this material accessible to students of varied backgrounds.					

Course Name:	**Graph Theory**			**Course Code:**	**404463**
Hours:	**Credit**	**Theoretical**	**Practical**	**Pre-Requisite:**	**404241**
	3	**3**	**0**		
The course deals with graph theoretical notions and problems, and the use of algorithms, both in the mathematical theory of graphs and its applications. In the course, the basic theory of graphs of different kinds is developed in detail, especially trees and bipartite graphs.					

Department requirements: (84 CREDIT HOURS)

Compulsory department requirements: (69 CREDIT HOURS)

Course Name:	**Digital Logic Design**			**Course Code:**	**401105**
Hours:	**Credit**	**Theoretical**	**Practical**	**Pre-Requisite:**	
	3	**3**	**0**		
Number systems, binary codes, Boolean algebra, and logic gates. Simplification of Boolean functions. Combinational logic: Adders, subtractions, code converters, comparators, encoders, decoders, multiplexers, and ROMs. Sequential logic: Flip-Flops, Registers, Counters, and RAM.					

Course Name:	**Linear Algebra (1)**			**Course Code:**	**404241**
Hours:	**Credit**	**Theoretical**	**Practical**	**Pre-Requisite:**	**404101**
	3	**3**	**0**		

System of linear equations, matrices, determinants, vector space in the second and third dimensions, non-vector multiplication, vector multiplication, general vector space, subspaces, linear independence, base and dimension, orthogonal basis, (Gram-Smith) operations, base change, linear transformations.

Course Name:	**Introduction to Algorithms**			**Course Code:**	**401115**
Hours:	**Credit**	**Theoretical**	**Practical**	**Pre-Requisite:**	**201251**
	3	**3**	**0**		

Algorithms, Mathematical induction, classifying functions. Computational complexity of algorithms. Searching and sorting algorithms. Algorithm analysis and design techniques: Divide and conquer greedy methods. Trees and graphs. Hashing algorithms. Combinatorial algorithms. p/Np problems.

Course Name:	**Introduction to Software Engineering**			**Course Code:**	**410120**
Hours:	**Credit**	**Theoretical**	**Practical**	**Pre-Requisite:**	**407212**
	3	**3**	**0**		

This course provides a general introduction to software engineering. It introduces concepts such as software processes and agile methods, and essential software development activities, from initial specification through to system maintenance. Formalisms and tools to assist in software development are also presented, including common design patterns and UML notation. There is a focus on software testing, from unit testing to the testing of software releases. Project management, software security, professional software engineering practice, and IT governance will also be covered. Case studies provide practical examples for many of these concepts.

Course Name:	**Object Oriented Programming**			**Course Code:**	**407212**
Hours:	**Credit**	**Theoretical**	**Practical**	**Pre-Requisite:**	**401112**
	3	**3**	**1**		

This course introduces the basic concepts of object-oriented programming using the Java language. It includes: The basic characteristics of object- oriented programming, which include data abstraction, encapsulation and data masking, inheritance, and multiple shape transformations. The concept of classes, objects, methods of creating objects, overloading, and overriding, interfaces, packages, exception handling, program
programming and introduction to programming user interfaces

Course Name:	**Computer Architecture**			**Course Code:**	**401221**
Hours:	**Credit**	**Theoretical**	**Practical**	**Pre-Requisite:**	**401105**
	3	**3**	**0**		

This course introduces the software and physical components of computer architecture and includes computer structures and their types, instruction set architecture, computation and logic unit, control unit, data transmission lines, instructions and control signals, the hierarchical structure of memory instructions, methods of measuring computer performance, improving computer performance using technology Tubes. Connecting and connecting the input and output units with the central processing unit.

Course Name:	**Data Structures**			**Course Code:**	**401251**
Hours:	**Credit**	**Theoretical**	**Practical**	**Pre-Requisite:**	**407212**
	3	**3**	**1**		

This course covers the basic concepts of data structures and includes abstract data concepts, describing different data structures as abstract data such as: lists, stacks, queues, dictionaries, and tree structures. programming these structures in different ways using the object-oriented programming method the basic concepts for analyzing algorithms depending on time and capacity for different applications of data structures. This course also covers the technique of self-recall to solve problems and a brief introduction to diagrams.

Course Name:	Operating Systems			Course Code:	401332
Hours:	Credit	Theoretical	Practical	Pre-Requisite:	401115
	3	3	0		

This course covers the basic concepts of operating systems. Topics covered: evolution of operating systems (OS), operating system architecture, operating system tasks, including managing processing operations and scheduling them (time sharing, deadlocks, strategies for managing and protecting storage units, architecture of secondary storage units. Distributed systems, managing input units / Output)

Course Name:	Database			Course Code:	409252
Hours:	Credit	Theoretical	Practical	Pre-Requisite:	401251
	3	3	1		

This course introduces the concepts of databases and database management systems and includes an introduction to the database systems model relational data, including relationship algebra, relationship differentiation, synthetic query language, database design methodologies, Entity Relationships and Constraints Integrity Conditions model, conceptual database design and functional dependency standard formulas and normalization methods.

Course Name:	Advanced Software Engineering			Course Code:	410255
Hours:	Credit	Theoretical	Practical	Pre-Requisite:	410120
	3	3	0		

This course focuses on software engineering for smart, critical, and complex software-intensive systems. The course contains four modules. 1) Requirements specification module focuses on methods to transit from user requirements to high-quality technical requirements; 2) Testing management model focuses on testing strategies; 3) Code quality module focuses on code analysis, code review, and code refactoring; 4) Complex system module focuses on verification and validation of complex software systems.

Course Name:	Software Quality Assurance			Course Code:	410330
Hours:	Credit	Theoretical	Practical	Pre-Requisite:	410312
	3	3	0		

This course will delve into various software quality topics. The course takes a holistic view of software quality considering both building the right software and building the software right. First, to learn about building the right software we will focus on understanding software requirements elicitation, specification, analysis, validation, and verification. Next, the course considers building the software right with a focus on code quality and software testing. In addition, we will cover important tools to aid in maintaining software quality like version control and bug trackers.

Course Name:	Web Application Programming			Course Code:	407356
Hours:	Credit	Theoretical	Practical	Pre-Requisite:	410232
	3	3	1		

This course covers the theoretical and practical concepts of web programming and includes the Study of static hypertext markup language Dynamic, JavaScript, PHP Script, Application and page Frames, Web Forms, Controls and Validation Home pages, data linking, and server computer technology, including practical applications using the Active Server Pages language in a .Net environment and dealing with databases using ADO.Net technology

Course Name:	Visual Programming			Course Code:	410232
Hours:	Credit	Theoretical	Practical	Pre-Requisite:	407212
	3	3	1		

This course introduces the concepts of the visual programming environment and the characteristics of object-oriented programming. It includes a Study of Microsoft Data net technology platforms, which include: the architecture of this technology, the languages used in it, the library The standard born net, the virtual machine for running programs, the way projects work, their translation and implementation. C# includes the following main topics: integrated development environment, user interface design, control tools, Event handling, control structures, functions, classes, objects, exception handling, graphics, and instructions for handling Files.

Course Name:	Database Management Systems			Course Code:	410345
Hours:	Credit	Theoretical	Practical	Pre-Requisite:	409252
	3	3	0		

The course aims at advanced topics in database systems, ideas about the relational model for many types of database technology aspects such as SQL, database optimization, distributed database processing and vision support, data protection problems, concepts about data retrieval, parallel data processing, data security and integrity Data, object-oriented databases, introduction to database management, standardization, data design, and data mining.

Course Name:	Human-Computer Interaction			Course Code:	410216
Hours:	Credit	Theoretical	Practical	Pre-Requisite:	410232
	3	3	0		

Designing practical, effective, and enjoyable computer interfaces is a skill used by system architects who study human-computer interaction. This course covers the theory, design process, and programming concepts that underlie successful human-computer interaction. The design, implementation, and evaluation of interactions are topics that the students learn about well. Understanding the theory underlying effective human-computer interaction and the accepted practices for good user interface design, such as the "usability" process, are essential for the design process. This process includes iterative evaluation, which we will learn and put into practice using case studies that are based on scenarios. Students will put their newly acquired skills to use in a variety of real-world assignments to design effective user interfaces.

Course Name:	Requirement Engineering			Course Code:	410312
Hours:	Credit	Theoretical	Practical	Pre-Requisite:	410255
	3	3	0		

Requirements engineering is one of the least understood and hardest phases in the development of software products, mainly because requirements are often unclear in the minds of most stakeholders. This course deals with identifying stakeholders, eliciting and verifying requirements from them, and translating into detailed requirements for a new software product, analysis, and modeling of requirements. The first steps in the direction of software design and quality assurance aspects of the software requirements phase of the software development process.

Course Name:	Software testing			Course Code:	410423
Hours:	Credit	Theoretical	Practical	Pre-Requisite:	410405
	3	3	1		

The focus of this course will be on software testing. A variety of test techniques will be covered along with applicability aspects as well as the bindings on software reliability modeling. The course includes Introduction to Software Verification and Validation, Software Testing Overview and Classifications, Functional (black box) Testing, Structural (white box) Testing, Integration Testing, Mutation Testing, Model-based Testing, and Test case generation, Software Reliability Modeling together with bindings on testing, Overview of the testing process, testing tools, and automation. The contents will reflect the latest research topics as well as industrial practices. Guest lectures by industrial experts will be the highlight of this course.

Course Name:	Design & Analysis of Information Systems			Course Code:	410432
Hours:	Credit	Theoretical	Practical	Pre-Requisite:	409251
	3	3	0		

This course gives an overview of the general stages of developing information systems (the software development life cycle), which include: preparing feasibility studies, methods of collecting information on system requirements (interviews, questionnaires, observation and participation in Application design), requirements analysis using the synthetic method (data flow diagrams, data dictionary), System design procedures, including: design of system architecture, design of input processes and output forms, design of interfaces Conversation with the user and database design. The course will deal with implementation, software testing, technical support, and maintenance software.

Course Name:	Computer-Aided Software Engineering (CASE TOOLs)			Course Code:	410335
Hours:	Credit	Theoretical	Practical	Pre-Requisite:	409255
	3	3	0		

The aim of the module is to develop an understanding of and gain experience with the basic techniques of software engineering with the help of computer-aided software. The course provides an introduction to life cycle models, and various casework benches used in an integrated environment followed by hands-on experience on CASE TOOLS. Also, techniques for developing CASE tools that match the needs of software engineers will be addressed.

Course Name:	Software documentation			Course Code:	410420
Hours:	Credit	Theoretical	Practical	Pre-Requisite:	410405
	3	3	0		

This course provides an introduction to the task-oriented approach used by software engineers to create software documentation. Students will learn to write software documentation through a structured process, including user analysis, planning, designing, testing, and reviewing. The course covers various software documentation types, such as tutorials, procedures, and reference guides, and teaches students how to write each type effectively. Additionally, students will gain proficiency in using tools to produce comprehensive documentation for any software, while adhering to standard guidelines and employing clear English structures.

Course Name:	Software construction and Modeling			Course Code:	410405
Hours:	Credit	Theoretical	Practical	Pre-Requisite:	407356
	3	3	0		

This course introduces the students to software modeling and the object-oriented analysis and design process. The course covers the major Unified Modeling Language diagrams including use cases, class diagrams, sequence diagrams, activity diagrams, and deployment diagrams. Furthermore, case studies will be presented to teach the students how to use and apply each UML model in real-life scenarios.

Course Name:	**Graduation Project**			**Course Code:**	**410442**
Hours:	**Credit**	**Theoretical**	**Practical**	**Pre-Requisite:**	**90 Credit Hours**
	3	**0**	**0**		

This course allows students to demonstrate their intellectual, artistic, and creative abilities by developing a project in one of the fields of Information Technology. The graduation project challenges students to transcend the learning they are learning in their scheduled educational programs. Students will complete their projects in focused fields of study under the guidance and supervision of the faculty. These projects will demonstrate students' ability to: apply, analyze, synthesize, evaluate information, and connect knowledge with understanding.

Course Name:	**Practical Training**			**Course Code:**	**410440**
Hours:	**Credit**	**Theoretical**	**Practical**	**Pre-Requisite:**	**90 Credit Hours**
	3	**0**	**0**		

This course provides the possibility of training in the use of computer tools and their applications in various fields. Training is conducted in public or private sector institutions under the supervision of the faculty members in the department. The purpose of field training Under the supervision of practical experiences, students assemble the knowledge, experience, and skills presented during the academic part of the program in a practical environment. Field training is a student's learning experience and on-site work contribution. Expected field training to provide learning opportunities that are not available in the classroom.

Elective department requirements: (9 credit hours)

Course Name:	**Project Management**			**Course Code:**	**410228**
Hours:	**Credit**	**Theoretical**	**Practical**	**Pre-Requisite:**	**410345**
	3	**3**	**0**		

This course composes four perspectives of Project Management and Software Project Management. These are the Project Management Perspective that was introduced by the Institute of Project Management (PMI), the Software Project Management Perspective that was introduced by the IEEE Computer Society (IEEE-CS SWEBOK), the SCRUM Software Project Management Perspective that was introduced during the past 15 years or so by different professional AGILE-SCRUM bodies (AGILE-SCRUM).

Course Name:	**Special topics in software engineering**			**Course Code:**	**410309**
Hours:	**Credit**	**Theoretical**	**Practical**	**Pre-Requisite:**	**60 credit hours**
	3	**3**	**0**		

A special topics course in software engineering typically focuses on specific, advanced areas within the broader field of software engineering. These courses delve into specialized topics that may not be covered extensively in a general software engineering curriculum. The specific content of a special topics course can vary depending on the instructor, student interests, and current trends in the industry.

Course Name:	**Formal Methods**			**Course Code:**	**410333**
Hours:	**Credit**	**Theoretical**	**Practical**	**Pre-Requisite:**	**404152 + 410120**
	3	**3**	**0**		

This course introduces students to the practical applications of formal methods in the software development process. It covers key topics from the CS2013 Final Report by ACM and the IEEE Computer Society, including the role of formal specification and analysis techniques, program assertion languages, formal software modeling, and analysis methods, as well as tools that support formal methods. The course aims to show how using formal methods, even in a pragmatic way, can lead to the development of high-quality, maintainable software systems while managing development costs effectively.

Course Name:	**Reverse software engineering**			**Course Code:**	**410344**
Hours:	**Credit**	**Theoretical**	**Practical**	**Pre-Requisite:**	**410312**
	3	**3**	**0**		

This course content enables us to understand the importance of Reverse software engineering, its techniques, and concepts that are used to manage various industrial computer applications/software. Students will be made aware of software engineering concepts, differences between forward and reverse engineering, UML diagrams, rapid prototyping and reverse engineering process. Students will also do case studies of various computer applications/software with the use of UML diagrams, rapid prototyping and reverse engineering processes.

Course Name:	**Software Metrics and Standards**			**Course Code:**	**410404**
Hours:	**Credit**	**Theoretical**	**Practical**	**Pre-Requisite:**	**410432**
	3	**3**	**0**		

Software Metrics is a key area in software engineering that focuses on the quantitative measurement and evaluation of software quality, performance, and development processes. It aims to help students and professionals understand how to assess various aspects of software systematically to make informed decisions and enhance quality.
Software Standards are a set of rules and guidelines used to standardize software design, development, and quality assurance processes. These standards ensure that the final product meets predefined quality benchmarks and complies with organizational or industry-wide requirements. They are typically defined by international organizations such as ISO (International Organization for Standardization) and IEEE (Institute of Electrical and Electronics Engineers).

Supportive department requirements: (6 credit hours)

Course Name:	**Discrete Mathematics**			**Course Code:**	**404152**
Hours:	**Credit**	**Theoretical**	**Practical**	**Pre-Requisite:**	**409100**
	3	**3**	**0**		

This course will include an introduction to logic, hypothetical logic, predicate logic, formal and informal proofs, sets, and static operations. Countable and uncountable functions and combinations. Integers and modular arithmetic, sequences, aggregates, mathematical induction, repetition, counting, permutations, combinations, probability, relationships, graphing, and tree theory.

Course Name:	**Networks and Data Communication**			**Course Code:**	**407326**
Hours:	**Credit**	**Theoretical**	**Practical**	**Pre-Requisite:**	**45 Credit Hours**
	3	**3**	**0**		

This is a first-class on the fundamentals of data communication networks, their architecture and network layers, operations principles, transmission protocols, and performance analyses. One goal will be to give some insight into the rationale of why networks are structured the way they are today and to understand the issues facing the designers of next-generation data networks. Much of the class will focus on network algorithms and their performance.