Ministry of Higher education and Scientific Research

UNIVERSITY OF ANBAR

COLLEGE OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

Department of Information system

Department of Information system

CATALOUGE

2023/2024

Vision

Our vision is for the department to be, along with the college, an “educational” and “research” edifice in its programs, educational curricula, and scientific research, and it seeks to achieve a prominent position among the relevant departments in Iraqi universities by providing and updating distinguished, modern programs that focus on the requirements of the labor market and technological development.

Mission

The department's main mission is to graduate groups of students who have high academic and professional qualifications to engage in work in all state departments in order to put all the skills they have acquired during their years of study in the service of their country and its scientific.

Programme Educational Objectives

The department aims to achieve the following goals:

1. The department aims to prepare qualified cadres in the field of building systems and databases to provide state departments and institutions with expert cadres and to develop the software industry in Iraq, keep pace with the tremendous developments in this field, and deal with modern technologies and the information network.

2. Prepare the student systematically to be able to study problems and challenges in the field of information systems science and technology.

3. Enabling the systems analyst to lead a programming team to prepare a computer system that solves the problems of users and beneficiaries.

4. Developing the student’s mental abilities through analysis and logical deduction and enabling him to solve programming problems

5. The necessary development of curricula to ensure the integration of modern changes in computer science technology.

6. Encouraging innovative ideas and projects and developing leadership and creativity skills in the field of information technology by urging students to participate in computer events and forums:

- Programmer
- Database designer and administrator
- Information security manager
- Software developer
- Computer Center Manager
- Information systems analyst

Course Description

Courses are coded as follows:

1. Course code and number
2. Course title
3. Parenthesized numerals, e.g., (4-3-1-3), indicate, in order, the credit hours, the classroom hours (1 hour = 1 credit hour), tutorial hours (credit hour = 0), and the laboratory hours (3 hour = 1 credit hour).

Prerequisites, if any, are indicated at the course description. These have been established to assure an adequate and uniform background for students in advanced classes. Occasionally, students may feel they already have the appropriate background for an advanced course because of previous training, transfer credits, or Credit by examination.

Graduation Requirements

Requirements	Credit
University Requirements	12
College Requirements	32
Department Requirements	70
Elective Courses	11
Total	125

University Requirements: 12 credit hours

Course code	Course Title	Credit hours	Weekly hours	Prerequisite
UOA140	English-I	2	2	
UOA240	English -II	2	2	
UOA340	English-III	2	2	
UOA440	English-IIII	2	2	
UOA135	Human Rights	1	1	
UOA201	Democracy	1	1	
UOA137	Arabic Language	2	1	
Total		12	11	

College Requirements: 32 credit hours

Course Code	Course Title	Credit hours	Weekly hours			Prerequisite
			Lec.	Tut.	Lab	
CCIT060	Data Structures	3	2		2	
CCIT061	Object Oriented Programming I	4	3		2	
CCIT062	Numerical Analysis	3	2		2	
CCIT063	Object Oriented Programming II	4	3		2	
CCIT064	Visual Programming I	3	2		2	
CCIT065	Visual Programming II	3	2		2	
CCIT066	Artificial Intelligence I	3	2		2	
CCIT067	Artificial Intelligence II	3	2		2	
CCIT068	Project in IS	6			12	
Total		32	18		28	

Department Requirements: 70 credit hours

Course Code	Course Title	Credit hours	Weekly hours			Prerequisite
			Lec.	Tut.	Lab	
ISDE215	Computational Theory I	2	2		---	
ISDC202	Design and Analysis of Information Systems	2	2		--	
ISDC203	Advanced Mathematics	2	2		--	
ISDE212	Introduction to E-Business Systems	3	3		--	
ISDE317	Algorithms	3	2		2	

ISDE218	Computational Theory II	2	2		--	ISDE215
ISDC205	Design and Analysis of Databases	3	2		2	
ISDE219	Design Internet Pages	3	2		2	
ISDC303	Numerical Analysis	3	2		2	
ISDE321	Compiler I	3	2		2	
ISDC305	Principles Of Computer Network	3	2		2	
ISDC306	Database Management systems	3	2		2	
ISDC307	Project Management Systems	2	2		--	
ISDE324	Compiler II	3	2		2	ISDE321
ISDE325	Data Network	3	2		2	
ISDE326	Distributed Databases	3	2		2	
ISDC309	Software Engineering	2	2		--	
ISDC327	Data Management Systems	2	2		--	
	Decision Support Systems	2	2		--	
ISDC406	Information Security	2	2		--	
ISDE414	Information Retrieve and Search Engine	3	2		2	
ISDE426	Multimedia Computing I	3	2		2	
ISDC403	Data Warehouse	2	2		--	
ISDC422	Web Application Programming	2				
ISDE427	Multimedia Computing II	2				ISDE426
ISDE313	E- Commerce	3				
ISDC404	Data Mining	2				
ISDE419	Mobile Applications Programming	2				

Total	**70**	**73**	**55**	**28**	

Elective Courses :11 credit hours

Course Code	Course Title	Credit hours	Weekly hours			Prerequisite
			Lec.	Tut.	Lab	
ISDC202	System Analysis and design	2	2			
ISDE212	Introduction to electronic information systems	2	2			
ISDE219	Internet page design	3	2		2	
ISDC309	Software engineering	2	2		----	
ISDE313	E-Commerce	2	2		---	
Total		**11**	**10**		**2**	

SECOND LEVEL

Course code	Course Title	Credit Hours	Weekly hours			Prerequisite
			Lec.	Tut.	Lab.	
CCIT061	Object Oriented Programming I	4	3		2	
CCIT060	Data Structures	3	2		2	
ISDE215	Computational Theory I	2	2		----	
ISDC202	Design and Analysis of Information Systems	2	2		----	
ISDC203	Advanced Mathematics	3	3		-----	
ISDE212	Introduction to E-Business Systems	2	2		-----	
UOA201	Freedom and Democracy	1	1		-----	
CCIT063	Object Oriented Programming II	4	3		2	CCIT061
ISDE317	Algorithms	3	2		2	
ISDE218	Computational Theory II	2	2		--	ISDE215
ISDC205	Design and Analysis of Databases	3	2		2	
ISDE219	Design Internet Pages	3	2		2	
CCIT062	Numerical Analysis	3	2		2	
UOA240	English (2)	2	2		--	
Total		37	30		14	

THIRD LEVEL

Course code	Course Title	Credit Hours	Weekly hours			Prerequisite
			Lec.	Tut.	Lab.	
CCIT064	Visual Programming I	3	2		2	
ISDE321	Compiler I	3	2		2	
ISDC305	Principles Of Computer Network	3	2		2	
ISDC306	Database Management systems	3	2		2	
UOA340	English(3)	2	2		-----	
ISDC307	Project Management Systems	2	2		-----	
CCIT065	Visual Programming II	3	2		2	CCIT064

ISDE324	Compiler II	3	2		2	ISDE321
ISDE325	Data Network	3	2		2	
ISDE326	Distributed Databases	3	2		2	
ISDC309	Software Engineering	2	2		----	
ISDC327	Data Management Systems	2	2		----	
Total		34	24		16	

FOURTH LEVEL

Course code	Course Title	Credit Hours	Weekly hours			Prerequisite
			Lec.	Tut.	Lab.	
ISDC406	Information Security	2	2		--	
ISDE426	Multimedia Computing I	3	2		2	
CCIT066	Artificial Intelligent I	3	2		2	
ISDE414	Information Retrieve and Search Engine	2	2		--	
ISDC403	Data Warehouse	2	2		--	
ISDC422	Web Application Programming	3	2		2	
CSDE423	Research Methodology	2	2		--	
ISDE425	Information Security II	2	2		--	
ISDE427	Multimedia Computing II	3	2		2	ISDE426
CCIT067	Artificial Intelligent II	3	2		2	CCIT066
ISDE313	E- Commerce	2	2		--	
ISDC404	Data Mining	2	2		--	
ISDE419	Mobile Applications Programming	3	2		2	
UOA440	English(4)	2	2		--	
CCIT068	Project	6	--		12	
Total		40	28		24	

Curriculum Map of the PEOs with courses according to learning outcomes

Programme Educational Objectives PEOs		Cognitive objectives				Skills objectives of the programme				Thinking skills
		1A	2A	3A	4A	1B	2B	3B	4B	1C
Course code	**Course title**	**Second LEVEL**								
ISDC207	Object Oriented Programming I	√				√	√	√		
CSIT201	Data Structures	√				√	√			√
ISDE215	Computational Theory I		√			√	√			
ISDC202	Design and Analysis of Information Systems									
ISDC203	Advanced Mathematics	√	√			√	√			
ISDE212	Introduction to E-Business Systems	√	√			√	√			
UOA201	Freedom and Democracy	√				√		√		√
ISDE211	Object Oriented Programming II		√	√		√		√		
ISDE317	Algorithms	√		√		√	√			
ISDE218	Computational Theory II	√	√			√	√			
ISDC205	Design and Analysis of Databases		√	√		√	√			√
ISDE219	Design Internet Pages	√	√			√	√			
ISDC303	Numerical Analysis	√	√			√	√			
UOA240	English (2)	√					√	√		
Program Educational Objectives PEOs		1A	2A	3A	4A	1B	2B	3B	4B	1C
Course code	**Course title**	**THIRD LEVEL**								
ISDC308	Visual Programming I	√	√	√		√	√	√		√
ISDE321	Compiler I	√			√		√			√
ISDC305	Principles Of Computer Network	√	√			√	√	√		
ISDC306	Database Management	√	√			√	√			

	systems									
UOA340	English(3)	√				√	√			√
ISDC307	Project Management Systems	√	√			√	√			
ISDE323	Visual Programming II	√				√	√	√		√
ISDE324	Compiler II	√	√			√		√		√
ISDE325	Data Network		√	√		√		√		
ISDE326	Distributed Databases	√			√	√	√			√
ISDC309	Software Engineering									
ISDC327	Data Management Systems	√	√	√		√	√			√
Program Educational Objectives PEOs		**1A**	**2A**	**3A**	**4A**	**1B**	**2B**	**3B**	**4B**	**1C**
Course code	**Course title**	**FOURTH LEVEL**								
ISDC406	Information Security I	√				√				√
ISDE426	Multimedia Computing I	√				√				√
ISDC405	Artificial Intelligent I	√				√				√
ISDE414	Information Retrieve and Search Engine	√				√				√
ISDC403	Data Warehouse	√				√				√
ISDC422	Web Application Programming	√				√				√
CSDE423	Research Methodology		√	√		√	√		√	
ISDE425	Information Security II	√			√	√	√		√	
ISDE427	Multimedia Computing II	√	√			√		√	√	
ISDE429	Artificial Intelligent II	√	√			√	√		√	
ISDE313	E- Commerce	√			√			√		
ISDC404	Data Mining	√			√			√		
ISDE419	Mobile Applications Programming	√	√						√	
UOA440	English(4)	√	√			√			√	
ISDC407	Project	√			√	√	√		√	
Elective courses										

Course code	Course title									
ISDC202	System Analysis and design			√	√			√	√	
ISDE212	Introduction to information systems		√	√						√
ISDE219	Internet page design			√	√					√
ISDC309	Software engineering		√	√				√		
ISDE313	E-Commerce			√	√				√	√

UNIVERSITY REQUIREMENT COURSES

OR

College REQUIREMENT COURSES

OR

Department REQUIREMENT COURSES

Course: Object-Oriented Programming II (CCIT063) (5-3-0-2)

Definition (Object-Oriented Programming I): Object-Oriented Programming (OOP) is a software development approach that uses real-world concepts (objects) to organize the program. In this style, real-world objects are represented as "objects" in the program, defined by a set of properties (variables) and behaviors (functions or methods).

Syllabus Breakdown:

- Week 1: Introduction to operator overloading
- Week 2: Overloading operators using functions
- Week 3: Overloading unary operators
- Week 4: Tips and constraints on operator overloading
- Week 5: Non-member operator functions
- Week 6: Using friend to overload a unary operator
- Week 7: Overloading relational and logical operators
- Week 8: Introduction to inheritance
- Week 9: Controlling access to the base class
- Week 10: Using protected members
- Week 11: Multiple base classes
- Week 12: Constructors and destructors
- Week 13: Passing parameters to base class constructors
- Week 14: Virtual base classes
- Week 15: Final exam

Course Description:

Object-Oriented Programming (OOP) is a common and effective software development approach based on organizing the program around objects that represent real-world things. Here is a comprehensive description of Object-Oriented Programming:

- Objects: In OOP, objects represent real-world entities such as cars, employees, books, etc. Objects consist of data (properties) and behavior (methods or functions). For example, a "car" object might include data such as color, speed, and model, and behaviors such as starting and stopping.
- Classes: Classes are the definitional templates for objects. They contain data (properties) and behavior (methods). Objects are created from classes through a process called "instantiation." For instance, you could have a class named "Car" that defines properties and behaviors shared by all types of cars.
- Inheritance: This concept allows a class to benefit from the properties and behaviors defined in another class. This allows for code reuse and minimizes redundancy. For example, a "Sports Car" class could be based on the "Car" class and include some additional specific properties and behaviors.
- Encapsulation: Refers to bundling the data and behaviors associated with an object into a single unit and hiding it from the outside. This enables control over data access and maintains complexity behind defined barriers.
- Message Passing: Objects exchange information through message sending. For example, a car object might send a message to an engine object to start it.
- Polymorphism: Refers to the ability of objects to behave differently based on the context. For instance, objects from different classes can act in different ways according to each class's implementation of the same method.

Learning Outcomes: Through OOP, we can eliminate redundant code and expand the use of existing classes. We can build programs from standard work units that communicate with each other, rather than having to start writing code from scratch. This leads to development time savings and increased productivity. Data hiding principles help programmers build secure programs that cannot be breached by code in other parts of the program. It's possible to have multiple objects coexist without any interference. It's possible to map real-world problem domain objects to those in the program. Work in a project can easily be divided based on objects. The data-centric design approach allows capturing more model details in an executable form. Object-oriented systems can easily be upgraded from small systems to large systems. Message passing techniques make interfacing with external systems much easier. Program complexity can be easily managed. Reusability is possible.

Recommended Textbooks:

- "C++ from the Ground Up" by Herbert Scheldt, Third Edition, McGraw-Hill/Osborne, 2013.
- "Object-Oriented Programming in C++" Fourth Edition by Robert Lafore, Copyright © 2002 by Sams Publishing

Prerequisites for the Course: ISDC207

Lab Syllabus:

- Week 1: Introduction to operator overloading
- Week 2: Overloading operators using member functions
- Week 3: Using friend to overload a unary operator
- Week 4: Exam 1
- Week 5: Definition (Inheritance)
- Week 6: Constructors and destructors
- Week 7: Exam 2

Course: Algorithms (ISDE317) (4-2-0-2)

Definition (Algorithms): Algorithms are a set of organized steps used to solve a specific problem or perform a specific task. An algorithm is an essential part of computer science and refers to the precise and methodical way in which a variety of problems are solved.

Syllabus Breakdown:

- Week 1: General structure of the topic and study vocabulary. General vocabulary.
- Week 2: Definition of algorithms, their characteristics, and how to write them, introduction to the article.
- Week 3: Complexity of an algorithm in terms of time and execution, calculating the complexity of the algorithm in terms of time and steps.
- Week 4: Recursion
- Week 5: Review all previous lectures with homework, solving evaluation methods in the previous three lectures.
- Week 6: How to choose a sorting algorithm based on data, introduction to sorting algorithms.
- Week 7: Midterm test
- Week 8: Understanding how an algorithm works. Choice sorting algorithm.
- Week 9: Insertion sort algorithm.
- Week 10: Bubble sort algorithm.
- Week 11: Solving evaluation methods in the previous three lectures.
- Week 12: Representing data as a tree. Trees.
- Week 13: Programming a tree representation. Printing, deleting, and adding to the tree in code form.
- Week 14: How to search in trees. Search algorithms.
- Week 15: Preparation week.
- Week 16: Final exam.

Course Description:

Effectiveness: Algorithms must solve problems efficiently and within a reasonable timeframe, typically measured in time or space units.

Correctness: Algorithms should provide an accurate and correct solution to the specified problem.

Clarity: The steps in the algorithm should be clear and understandable, making it easy to comprehend and implement.

Analysis: Algorithms are analyzed to estimate their efficiency and performance in various cases, such as best, worst, and average scenarios.

Applicability: Algorithms must be practically executable on a computer or within a specific programming environment.

Learning Outcomes of the Course:

A- Knowledge and Understanding: This article is based on knowledge. B- Subject-specific Skills: Learn C++ programming professionally.

Recommended Textbooks: "Data Structures Using C++" by D.S. Malik - This book presents data structure concepts using the C++ programming language.

Prerequisites for the course: None

Computational Theory II (ISDE218) (2-2-0-0):

It is a branch of computer science that focuses on studying algorithms and problems related to computational capacity and data interaction. This branch aims to understand the basic concepts governing computational processes and their ability to solve a variety of problems.

Syllabus:

- Week 1: Basics of regular languages
- Week 2: Context-free grammars
- Week 3: Tree analysis; ambiguity in grammars and languages
- Week 4: Standard models
- Week 5: Chomsky's natural model

- Week 6: Greibach's normal forms
- Week 7: Minimizing CFG
- Week 8: Automatic compression system
- Week 9: Formal definition of deterministic and non-deterministic PDA
- Week 10: Midterm test
- Week 11: Transition functions for NPDA; NPDA implementation
- Week 12: Accepting strings using NPDA
- Week 13: Turing machines
- Week 14: Programming techniques for Turing machines
- Week 15: Decidable languages and problems
- Week 16: Preparation week before the final exam

Course Description:

Automata Theory: Focuses on studying mathematical models that describe the behavior of specific computational devices, such as Finite Automata, Pushdown Automata, and Turing Machines.

Formal Language Theory: Focuses on studying languages used to describe data and interactions within computational devices, encompassing languages processed by Finite Machines and Turing Machines.

Computability Theory: Concerns the study of problems that can be solved by computers and algorithms, defining the limits of what can be computed and executed by computing systems.

Infinite Theory: Focuses on studying computational processes related to infinite counting, infinite sets, and their interactions.

Learning Outcomes of the Course:

- Knowledge and Understanding: Gain a full understanding and mindset of automata theory as a foundation for all computer science.
- Cognitive Skills (Thinking and Analysis): Ability to design FAs, NFAs, grammars, model languages, and basics of minor translators.
- Communication Skills (Personal and Academic): Ability to reduce English grammar and context-free grammars.
- Practical and Subject-Specific Skills (Transferable Skills): Training in practical applications and scientific reporting.

Recommended Textbooks:

- "HTML and CSS: Design and Build Websites" by Jon Duckett - An introductory and comprehensive book covering the basics of HTML and CSS in detail with many practical examples.
- "Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics" by Jennifer Niederst - A book providing a comprehensive introduction to web design including HTML, CSS, JavaScript, and web graphics.

ISDC205 (Database Analysis and Design) (4-2-0-2)

Course Definition (Database Analysis and Design):

This course is categorized as "Required" in the Information Systems department.

Course Syllabus:

- Defining Information Systems Analysis
- Role of the Systems Analyst

- Characteristics of a Systems Analyst
- Types of Subsystems
- Framework for a Systems Approach
- System Characteristics
- Executive Support System (ESS)
- Decision Support System (DSS)
- Office Automation System (OAS)

Course Description:

- Data Modeling: Designing data structures suitable for effectively storing and organizing information, including entities and their relationships.
- Database Design: Developing table structures, defining relationships between tables, identifying primary and foreign keys, and designing data models based on requirements analysis.
- Database Query Languages: Using query languages such as SQL (Structured Query Language) to extract and analyze data from databases.
- Data Security and Sustainability: Studying how to protect data and ensure its sustainability and integrity during storage and retrieval processes.

Recommended Textbooks:

https://link.springer.com/book/10.1007/978-1-84628-655-1

Prerequisites: ISDE215 Laboratory Syllabus:

- Introduction to Database Concepts
- Database Environment

- Relational Model
- Entity Relationship Model
- Introduction to SQL
- Basic SQL Tables
- Database Creation
- Data Modeling
- Constraints & Data Manipulation
- Database Design (Logical and Conceptual)
- Normalization Database Objects User Creation and Management
- Managing DB tables-Data Integrity
- Single and Multiple table queries
- Advanced Queries, Subqueries & Merge, and Introduction to Regular Expression Functions
- Preparatory week before the final Exam

(ISDE219)(Web Page Design)(4-2-0-2)

Course Definition (Web Page Design):

This course is categorized as "Required" in the Information Systems department.

Course Syllabus:

The course covers the basics of designing and building web pages, types of web pages, and how to publish pages and websites on the Internet. The course also deals with the use of one of the commonly used web page editing programs in building a personal or educational website.

Course Description:

- Frontend Design: Studying the design and development of the user frontend interface of pages, including the use of HTML, CSS, and JavaScript to achieve a distinguished user experience.
- User Experience Design (UX): Understanding user needs and designing interactive interfaces that provide a comfortable and enjoyable user experience.
- Graphic Design: Using graphic design software to create graphics, images, and icons that enhance the attractiveness of the pages and express the visual identity of the site.
- Responsive Design: Learning how to design web pages that flexibly adapt to different screen sizes and devices, including smartphones and tablets.
- Backend Design: Understanding the basics of servers, databases, and backend server technology to support and effectively operate web pages.
- Content Analysis and Strategy: Developing strategies to analyze and structure content effectively.

Course Outcomes:

A. Knowledge and Understanding A1. Explains the basics of designing and building web pages and related concepts, terms, and steps. A2. Explains the types of pages and sites and the languages used in designing and building pages and sites. A3. Explains the components, specifications, and standards of designing pages and sites and their interfaces and their use in evaluating sites. A4. Discusses the most important and popular programs and tools for building and designing sites and pages and compares them.

B. Subject-Specific Skills B1. Summer Training B2. Graduate Studies Research B3. Scientific Reports

Recommended Textbooks:

- "HTML and CSS: Design and Build Websites" by Jon Duckett - A beginner-friendly, comprehensive book covering the basics of HTML and CSS in detail with many practical examples.
- "Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics" by Jennifer Niederst - A book that provides a comprehensive introduction to web design including HTML, CSS, JavaScript, and web graphics.

Prerequisites: None Laboratory Syllabus:

- The basics of designing and building web pages and websites, the main concepts, steps, and stages of website design.
- Types of websites: fixed-content websites and variable-content websites
- Components, specifications, and standard standards for designing pages and websites and designing user interfaces and screens
- Languages and applications used to design web pages and associated tools Microsoft Word, HTML, FrontPage, Dreamweaver, PHP, CSS, CS5, and JavaScript, CGI scripts (Linux based), and VB scripts
- World Wide Web
- History of the World Wide Web
- Mid-term
- Uniform Resource Locator
- CSS
- JavaScript
- Why Study JavaScript?

- JavaScript Can Change HTML Content
- JavaScript Can Change HTML Attribute Values
- JavaScript Functions and Events

CCIT062 (Numerical Analysis) (4-2-0-2):

A branch of mathematics and computer science focused on developing and analyzing digital algorithms and techniques for solving complex mathematical problems that cannot be solved by traditional analytical methods. The course aims to transform mathematical problems into a sequence of computational operations solvable using computers.

Syllabus:

- Week 1: Direct methods for solving systems of linear equations
- Week 2: Simple Gaussian elimination, Gaussian elimination with partial pivoting
- Week 3: Gauss-Jordan method for calculating determinants
- Week 4: LU decomposition, Doolittle's LU decomposition with row swapping
- Week 5: Finding the inverse of a matrix
- Week 6: Iterative methods for solving systems of linear equations
- Week 7: Mid-term exam
- Week 8: Gauss-Seidel method, iterative method
- Week 9: Gauss-Seidel's successive over-relaxation method
- Week 10: Newton-Raphson method
- Week 11: Runge-Kutta method

- Week 12: Lagrange polynomials, Neville's method for data approximation
- Week 13: Numerical methods for differential equations
- Week 14: Numerical methods for integral equations
- Week 15: Preparation week
- Week 16: Final exam

Course Description:

Approximation and Estimation of Solutions: Using approximate methods to compute solutions for complex mathematical problems, such as numerical integration and root approximation. Solving Algebraic and Differential Equations: Developing algorithms for solving difficult algebraic and differential equations, such as solving ordinary and partial differential equations and differential integrations. Estimating Errors and Stability: Estimating errors occurring during numerical calculations and assessing the stability and accuracy of algorithms. Numerical Integration Techniques: Developing techniques for computing numerical integrals using various methods, such as Monte Carlo integrations and numerical integration techniques. Error Analysis and Numerical Solutions: Analyzing the impact of error in numerical calculations and developing solutions to effectively manage and reduce it.

Learning Outcomes of the Course:

1. Knowledge and Understanding
2. Acquiring the ability and skill to recognize and handle numerical methods.

Recommended Textbooks: "Numerical Analysis" by Richard L. Burden and J. Douglas Faires - A comprehensive and renowned book in the field of numerical analysis, covering a wide range of topics in detail with examples and applications.

Prerequisites for the course:

UOA240 English (2-2-0-0):

Enhancing English speaking, reading, and writing skills, learning a large vocabulary to help students handle the English language more easily.

Course Definition:

This course aims to enhance English language skills in speaking, reading, and writing, and to help students handle a large vocabulary and make the use of the English language easier.

Course Syllabus:

- Week 1: Grammar, reading, writing, listening, vocabulary.
- Week 2: Grammar, reading, writing, listening, vocabulary.
- Week 3: Grammar, reading, writing, listening, vocabulary.
- Week 4: Grammar, reading, writing, listening, vocabulary.
- Week 5: Grammar, reading, writing, listening, vocabulary.
- Week 6: Grammar, reading, writing, listening, vocabulary.
- Week 7: Midterm exam.
- Week 8: Grammar, reading, writing, listening, vocabulary.
- Week 9: Grammar, reading, writing, listening, vocabulary.
- Week 10: Grammar, reading, writing, listening, vocabulary.
- Week 11: Grammar, reading, writing, listening, vocabulary.
- Week 12: Grammar, reading, writing, listening, vocabulary.
- Week 13: Grammar, reading, writing, listening, vocabulary.

- Week 14: Review of key topics.
- Week 15: Preparation week.
- Week 16: Final exam.

Course Description:

The primary strategy for delivering this course will include:

1. PowerPoint presentations for data display.
2. Explanations using a whiteboard with different colored markers.
3. Discussions with students during teaching.
4. Interaction with students through practical problem-solving during lectures.
5. Solving various problems with additional exercises.
6. Assigning homework to enhance learning.

Recommended Textbooks:

- "English Grammar in Use" by Raymond Murphy - This book provides simplified and detailed explanations of English grammar with various practical exercises.
- "Advanced Vocabulary in Context" by Donald Scott - This book helps to expand and understand English vocabulary through its use in different contexts.

Prerequisites for the Course: None

CCIT065 (Visual Programming II) (4-2-0-2):

- **Course Definition:**
- Classified as a "required" course in the Information Systems department.

Course Syllabus:

This semester we explore visual programming, focusing on LINQ principles, Windows Forms applications, character and string processing, and file handling. Students will learn to create interactive GUIs (Graphical User Interfaces), manage file I/O tasks, process characters and strings, and query data effectively using LINQ. Projects and exercises are designed for practical application to enhance proficiency in these key areas.

Course Description:

The course aims to provide students with the essential knowledge and skills needed for software development using visual programming languages. Focusing on event-driven programming, data visualization, and user interface design, students will learn how to create, implement, and debug programs using graphical user interfaces. The course strongly emphasizes hands-on experience to help students master visual programming techniques.

Learning Outcomes:

Recommended Textbooks:

- "Visual C# How To Program" by Paul Deitel and Harvey Deitel, Deitel & Associates, Inc. Pearson, 2018.

Prerequisites for the Course: ISDC308

Laboratory Curriculum:

- Windows Forms Applications:
    - Basic form, buttons, text box, title box.
    - Check box, radio buttons, message boxes.

    - List box, combo box, picture box, vertical and horizontal scrolling, NumericUpDown control.
- Strings and Characters:
    - Fundamentals of strings, comparing string constructors, locating strings among characters and substrings within strings.
    - Extracting substrings from strings, chaining strings, various string methods.
    - Fundamentals of characters and methods for handling characters.
- Files:
    - Computer files, file categories, input files, output files.
- Mid-term Exam
- More Windows Forms Applications:
    - Timers, open dialog box, interaction with files and images.
    - Menu bar, toolbars, status bar, progress bar.
- Structures:
    - Basics of structures, structures with constructors.
    - Arrays of structures, lists of structures.
- LINQ Capabilities:
    - Basics of LINQ, querying a set of integers using LINQ.
    - LINQ with structures, querying a collection of employee objects using LINQ.
    -

ISDE324 (Compilers II) (4-2-0-2):

- **Course Definition:**
- Required or Elective / Required in the Information Systems Department.
- **Curriculum Content:**

1. Understanding Compiler Design Fundamentals: Students should understand the basic principles, techniques, and components involved in designing and implementing compilers.
2. Analyzing and Describing Compiler Phases: Students should be able to explain different phases of a compiler, including lexical analysis, syntax analysis, semantic analysis, intermediate code generation, optimization, and code generation.
3. Compiler Implementation: Students should gain practical experience by implementing a simple compiler for a programming language, possibly involving designing and developing the lexical analyzer, parser, semantic analyzer, and code generator.
4. Applying Formal Language Theory: Students should grasp formal languages, regular expressions, context-free grammars, and automata theory, applying this knowledge to analyze and manipulate programming languages.

- **Course Description:**
- The goal of the compiler course is to understand the foundational principles of compiler design, including its various components, algorithms, and necessary data structures.
- **Learning Outcomes:**

- Recommended Textbooks: A. Aho, R. Sethi, J.D. Ullman, "Compilers - Principles, Techniques and Tools," Addison-Wesley, 2007.
- Prerequisites for the Course: Compilers I
- Laboratory Curriculum:
    - Bottom-up parsing (shift-reduce parser)
    - Bottom-up parsing (LR parser)
    - Semantic Analysis (Type systems and type checking)
    - Semantic Analysis (Static analysis and error detection)
    - Mid-term Exam
    - Intermediate Code Generation
    - Intermediate representations
    - Syntax-directed translation and code generation
    - Intermediate Code Generation
    - Control Flow Analysis
    - Basic blocks
    - Data-flow analysis
    - Code Optimization (Part 1: Principles of Optimization, Common optimization techniques, Local code optimization)
    - Global Optimization Methods
    - Code Optimization (Part 2: Loop optimization, Register allocation, and instruction scheduling)

- Code Generation (Part 1: Target machine models and instruction sets, Instruction selection and mapping)
- Code Generation (Part 2: Memory Management and Runtime Support, Addressing modes, Memory management, Runtime support for generated code)
- Compiler Testing and Debugging (Testing strategies for compilers, Compiler validation techniques, Debugging and error handling in compilers)
- Advanced Topics: Just-in-time (JIT) compilation.

ISDE325 Data Networks (4-2-0-2)

- **Course Definition:**
- Required or Elective / Required in the Information Systems Department
- **Curriculum Content:**
- This course provides students with an overview of the concepts and fundamentals of data communication and computer networks. Topics to be covered include: data communication concepts and techniques in a layered network architecture, communication switching and routing, types of communication, network congestion, network topologies, network configuration and management, network model components, layered network models (OSI reference model, TCP/IP networking architecture) and their protocols, and various types of networks (LAN, MAN, WAN, and Wireless networks) and their protocols.
- **Course Description:**
- Understanding Network Fundamentals: Introduce students to the basic concepts and components of computer networks, including network architectures,

protocols, and layers. Exploring Network Protocols: Introduce students to various network protocols such as TCP/IP, UDP, HTTP, FTP, and DNS, and their roles in facilitating communication and data transfer in computer networks.

- **Learning Outcomes:**
- Understand the fundamental concepts and principles of computer networks, including network architectures, protocols, layers, and networking techniques. Explain the functions and interactions of different network layers, including the physical layer, data link layer, network layer, transport layer, and application layer. Demonstrate knowledge in network processing and routing, including IP addressing, subnetting, and routing algorithms. Configure network devices and troubleshoot them, such as routers, switches, and firewalls. Analyze and evaluate network performance and identify and resolve network issues and bottlenecks.
- **Recommended Textbooks:**
- Data Communications and Networking, 3rd or 4th edition, by Behrouz A. Forouzan Computer Networks, Fourth Edition, by Andrew S. Tanenbaum.
- Prerequisites for the Course: None
- Laboratory Curriculum: Introduction to Computer Networks:
  - Basic concepts of computer networks
  - Network architecture and protocols
  - Network standards and organizations Network Models and Protocols:
  - OSI model and TCP/IP protocol suite
  - Data encapsulation and protocol stacks
  - Network addressing and subnetting The Physical Layer and Data Link Layer:

- Transport media and signaling techniques
- Local area Ethernet networks and switching
- MAC addressing, error detection, and correction Network Layer

ISDE326 Distributed Databases (4-2-0-2)

- **Course Definition:**
- Required or Elective / Required in the Information Systems Department
- **Curriculum Content:**
- This course introduces students to the fundamental concepts and skills related to managing data within computer systems. This includes an introduction to relational databases, modeling, and normalization. Also, this course will introduce the structured query language (SQL), which is the standard language for relational database management systems (RDBMS).
- **Course Description:**

1. Understand the relational data model in terms of data structure, data integration, and data processing.
2. Understand and create conceptual database models using entity-relationship diagrams.
3. Design data structures that minimize redundancy and enforce data integrity, complying with regulatory requirements using normalization methodology.
4. Understand the underlying theory behind the relational data model as it applies to interactions with current database management systems.
5. Interpret a specific data model to query the database and transform data into information using SQL (Structured Query Language).

6. Implement a data model in an existing RDBMS.
7. Generate reports based on transaction data, including data sets and summary values.

- **Learning Outcomes:**

1. Apply basic concepts of database systems and applications.
2. Use SQL fundamentals and build queries using SQL to create and interact with databases.
3. Design a commercial relational database system (e.g., Oracle, MySQL) by writing SQL using the system.
4. Analyze and select data storage and retrieval techniques for the database system.

- Recommended Textbooks: Database System Concepts, 7th edition
- **Prerequisites for the Course: None**
- Laboratory Curriculum: Introduction to Database Concepts Database Environment Relational Model Entity Relationship Model Introduction to SQL Basic SQL Tables DB Creation Data Modeling Constraints & Data Manipulation Database Design (Logical and Conceptual) Normalization Database Objects User Creation and Management Managing DB Tables-Data Integrity Single and Multiple Table Queries Advanced Queries, Subqueries & Merge, and Introduction to Regular Expression Functions Preparatory week before the final exam

ISDC309 Software Engineering (2-2-0-0)

- **Course Definition:**
- Required or Elective / Required in the Information Systems Department
- **Curriculum Content:**

1. Understand the Core Principles of Software Engineering: Gain a foundational understanding of software engineering principles, including software development life cycles, methodologies, and best practices.

2. Analyze and Specify Software Requirements: Conduct requirement analysis and create detailed software requirement specifications that clearly outline user needs and system functions.

3. Design Effective Software Architectures: Produce detailed software design documents using appropriate design patterns and architectural styles, ensuring scalability, maintainability, and performance.

4. Develop and Test Software Solutions: Implement software solutions using current programming languages and frameworks, and conduct unit, integration, and system testing to verify that the software meets all specified requirements.

5. Manage Software Projects: Apply project management techniques to plan, execute, and monitor software projects, demonstrating effective risk management and team collaboration skills.

6. Employ Professional Software Engineering Tools and Practices: Use version control systems, adhere to coding standards, participate in code reviews, and exhibit professional software development practices.

7. Critically Evaluate Emerging Software Technologies: Assess new technologies and trends in software engineering, providing reasoned arguments for their potential impact on the field and their suitability for solving current problems.

8. Demonstrate Ethical and Social Responsibility: Evaluate the ethical and social implications of software engineering decisions in case studies.

Learning Outcomes of the Course:

1. Explain Basic Concepts of Software Engineering: Clearly illustrate the fundamental principles of software engineering, including various methodologies and software development life cycles (SDLCs).
2. Conduct Requirement Analysis and Specifications: Perform comprehensive sessions to gather requirements and draft detailed software requirement specifications that clearly define user needs and system functionalities.
3. Design Effective Software Architectures: Create detailed software design documents using appropriate design patterns and architectural styles, ensuring scalability, maintainability, and performance.
4. Develop and Test Software Solutions: Implement software solutions using current programming languages and frameworks, and conduct unit, integration, and system testing to ensure that the software meets all specified requirements.
5. Manage Software Projects: Apply project management techniques to plan, execute, and monitor software projects, demonstrating effective risk management and team collaboration skills.
6. Use Professional Software Engineering Tools and Practices: Employ version control systems, adhere to coding standards, participate in code reviews, and display professional software development practices.
7. Critically Evaluate Emerging Software Technologies: Assess new technologies and trends in software engineering, providing reasoned arguments for their potential impact on the field and their suitability for solving current problems.
8. Demonstrate Ethical and Social Responsibility: Assess the ethical and social implications of software engineering decisions in case studies.

Recommended Textbooks:

1. "Software Engineering" by Ian Sommerville - This book provides a solid foundation in software engineering principles, covering the software development lifecycle, methodologies, and more. It's an essential text that is widely respected in the field.

2. "Design Patterns: Elements of Reusable Object-Oriented Software" by Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides - A seminal guide to software design patterns that is fundamental for understanding how to create reusable software components.

Prerequisites for the Course: None

Laboratory Curriculum: None

ISDC327 (Management Information Systems) (2-2-0-0)

Course Definition:

Required or Elective / Required in the Information Systems Department

Curriculum Content:

- To acquaint students with the basic concepts of MIS.
- To explain the basic applications of MIS technology.
- To explore the principles and practice of MIS models.
- To highlight the technical and social issues related to MIS structure.

System Design Principles: The module aims to introduce students to the fundamental principles and concepts involved in designing information systems. This includes understanding system requirements, system modeling techniques, system architecture, and system integration.

Learning Outcomes of the Course:

- Students will become familiar with the various stages of the software development lifecycle, including requirement gathering, system analysis, system design, implementation, testing, deployment, and maintenance.

1. Describe the basic concepts of management information systems.
2. Have a good understanding of the strategies and technologies available for management information systems.
3. Discuss the social and ethical issues related to the structure of management information systems.

Specific Skills:

1. Apply and use MIS applications.
2. Demonstrate skills in using MIS models for various applications.
3. Show skills in applying MIS technology.

The aim is to introduce students to the standard industrial practices and methodologies for developing robust and scalable information systems.

Recommended Textbooks:

Davis, W. S., & Yen, D. C. (Eds.). (2019). The Information System Consultant's Handbook: Systems Analysis and Design

Prerequisites for the Course: None

Laboratory Curriculum: None

Decision Support Systems (DSS) (2-2-0-0)

Course Definition:

Required or Elective / Required in the Department of Computer Information Systems

Curriculum Content:

- Define a DSS, modeling, and decision-making and its processes.
- Determine effective ways of decision-making in an organization.
- Identify strategies for decision-making across an enterprise to mitigate the impact of environmental changes.
- Discuss methods for effectively building DSS components.
- Evaluate the success/failure of DSS.
- Highlight the importance of modeling in problem-solving.
- Describe the capabilities, benefits, and characteristics of DSS.

Learning Outcomes of the Course:

- Students will understand the general concept of decision support systems and interpret the relationship between information and decisions.
- Recognize the types of managerial decisions and their stages.
- Become familiar with the components, classifications, and functions of decision support systems.
- Understand the concept of models and the types of models used in decision support systems.
- Learn the meaning of business intelligence and explain the difference between business intelligence and decision support systems.
- Understand the concept of data warehouses and explain the difference between operational databases and data warehouses.
- Recognize the general concept of data mining, its characteristics, and applications.

- Understand the concept of online analytical processing (OLAP), its characteristics, and how it operates.
- Become acquainted with the concept of data visualization and the most common technologies used.

Recommended Textbooks:

- "Decision Support Systems and Intelligent Systems" by Efraim Turban, Jay E. Aronson, and Ting-Peng Liang - A book covering the basic concepts and practical applications of decision support and intelligent systems.
- "Business Intelligence: A Managerial Approach" by Efraim Turban, Ramesh Sharda, and Dursun Delen - A comprehensive overview of business intelligence systems and their use in supporting business decisions.
- "Analytics at Work: Smarter Decisions, Better Results" by Thomas H. Davenport and Jeanne G. Harris - A review of data analysis tools and techniques and their application in organizational decision-making.
- "Decision Support Systems for Business Intelligence" by Vicki L. Sauter - Focuses on the use of decision support systems to generate organizational intelligence and support decision-making processes.

Prerequisites for the Course: None

Laboratory Curriculum: None

ISDE425 (Data Security II) (2-2-0-0)

- ### Course Definition:
- Required or Elective / Required in the Information Systems Department

- **Course Curriculum:**

- This course provides an introduction to information systems security for undergraduates. It covers topics related to cryptography and information security. Cryptography generally deals with communication in the presence of adversaries, aiming to achieve objectives such as maintaining the privacy and integrity of transmitted data. In the first semester, the focus will be on classical and symmetric key cryptography, including block ciphers and their operational modes. The course focuses on rigorous mathematical formulations of security objectives and aims to train students to identify vulnerabilities in designs. Generally considered challenging by undergraduates, it is primarily theoretical and mathematical in nature, requiring the ability to understand abstract concepts. Students are required to solve assignments, complete homework, and undertake programming projects to develop their skills.

- **Course Description:**

  1. Critical understanding of theories, principles, and specialized concepts in security methods and modern protection technologies.
  2. Explanation of the basic structure of modern encryption protocols and encryption algorithms.
  3. Critical examination and evaluation of different strategies and techniques used in computer security, network security, and encryption strategies. Study of the algorithmic ideas embedded in the design and implementation of various computer security terms.

- **Learning Outcomes:**

  1. Increased awareness and understanding of the importance of information security.

2. Enhanced knowledge of common security threats and best practices for protecting sensitive information.
3. Enhanced skills in recognizing security incidents and responding to them.
4. Greater appreciation of the role of information security in overall business operations.
5. Increased motivation for implementation and adherence to security policies and procedures.
6. Improved ability to communicate and collaborate with IT and security teams.
7. Enhanced ability to educate and train others on best information security practices.
8. Increased confidence in addressing security concerns and making informed decisions regarding information security.

- **Recommended Textbooks:**
  - Stallings, W. (2019). Cryptography and Network Security: Principles and Practices. Upper Saddle River.
  - Schatz, Daniel; Bashroush, Rabih; Wall, Julie (2017). "Towards a More Representative Definition of Cyber Security". Journal of Digital Forensics, Security and Law. 12 (2). ISSN 1558-7215.
  - Wenliang Du, Computer Security: A Hands-on Approach (2017, self-published).
- **Prerequisites for the Course:** None

ISDE427 (Multimedia Computing II) (4-2-0-2)

- **Course Definition:**

- Required or Elective / Elective in the Information Systems Department

Course Curriculum:

The objective of multimedia computing is to provide the user with the best possible information from the database. One common form of interaction in multimedia computing is the user query, which is then utilized by the multimedia computing system to retrieve information that may be useful or relevant to the user. The focus is on multimedia computing rather than merely on multimedia data.

Course Description:

The "Multimedia Computing" course focuses on the study of how computer technologies are used in analyzing, processing, producing, and distributing multimedia. This course aims to equip students with the concepts and skills necessary to understand and apply the theoretical foundations and practical techniques related to multimedia, and use them across various applications. Topics in multimedia computing may include:

- Media Design Principles: Understanding the basics of media design and the factors influencing the design and development of effective multimedia.
- Analysis and Processing of Multimedia: Exploring the algorithms and methods used in the analysis and processing of multimedia such as images, sound, and video.
- Design and Development of Interactive Applications: Learning how to develop interactive applications that utilize multimedia, such as digital games and interactive applications.
- Virtual Reality and Augmented Reality: Studying the technologies that enable virtual and augmented reality experiences and their applications in fields like education and entertainment.

- Visual and Auditory Communication: Understanding how to design, develop, and distribute animations, films, and audios within the context of multimedia.
- Distribution and Publishing: Studying how to publish and distribute multimedia over the internet, networks, and various media.

Learning Outcomes:

A - Knowledge and Understanding A1. Describe the decision-making process. A2. Justify how the process can be improved with this system. A3. Objectives/Goals of DSS. A4. Discuss how those objectives/goals meet user needs. A5. Discuss how the decision support system can be integrated into normal business operations. A6. Explain the types of information that the system will require and how that information will be maintained.

B - Subject-Specific Skills B1. Summer training. B2. Graduate research. B3. Scientific reports.

Recommended Textbooks:

- "Multimedia: Computing, Communications & Applications" by Ralf Steinmetz and Klara Nahrstedt - This book provides a comprehensive and detailed overview of multimedia applications, technologies, and developments.
- "Multimedia Systems" by Ralf Steinmetz and Klara Nahrstedt - This book covers topics such as encoding, video and audio compression, animation, and the storage and distribution of media.
- "Introduction to Multimedia Systems" by Sugata Mitra - Provides a comprehensive introduction to multimedia concepts, technologies, and applications.

Prerequisites for the Course: (CSIT402)

Laboratory Curriculum:

- Introduction to Multimedia Computing.
- Concepts of the Internet and Search Engine Framework.
- Internet protocols.
- Internet Strategy.
- What is Multimedia Computing?
- General Multimedia Computing model.
- Executive Information Systems.
- Text representations and preprocessing.
- Tokenization & Normalization.
- Search Engines Components.
- Morphological normalization.
- Information Retrieval and Search Engine.
- Types of Search Engines.

CCIT067 Artificial Intelligence II (4-2-0-2)

- **Course Definition:**
- This course is categorized as "Required" in the Information Systems department.

Course Syllabus:

- Teaching Methods: The course is divided into theoretical and practical components. In the theoretical part, students learn about the scientific topic through explanations supported by examples. This is followed by interactive question-and-answer sessions between students and the instructor, where the instructor encourages students to solve examples themselves to enhance their understanding of the topic and boost their self-confidence.

In the practical part, the learning process takes place in the laboratory. Students learn the fundamentals of logical programming by presenting programming examples and encouraging constructive discussions among students to expedite the understanding process.

- **Course Description:**
  - Understand the definitions, characteristics, and types of artificial intelligence.
  - Distinguish between different AI search techniques.
  - Design intelligent systems to solve everyday problems.

Learning Outcomes of the Course:

- Introduce students to a new scientific subject, enhancing their knowledge in the field of computer science, specifically artificial intelligence.
- Familiarize students with the term "artificial intelligence" and its applications in computer science, engineering, and other related fields.
- Develop students' computational and mathematical skills.
- Teach students how to use logical thinking to represent and solve problems.
- The course aims to provide an understanding of the fundamentals and principles of AI methods, including algorithms and computer programs that mimic human, animal, or other behaviors. These methods enable computers (machines) to learn, reason, and interact with specific situations. One of these intelligent applications is the humanoid robot.

Recommended Textbooks:

- "Artificial Intelligence: A Modern Approach" by Stuart Russell and Peter Norvig, Pearson Education 2020.

Prerequisites for the Course: (ISDC405)

Laboratory Syllabus:

- General Introduction.
- The History of AI.
- Systematic Search: Basic Graph Concepts; State Space Representation of Problems.
- Depth-First Search.
- Breadth-First Search.
- Hybrid Search.
- Propositional Logic and Resolution in Propositional Logic.
- Predicate Logic: Basic Concepts and Definitions.
- Predicate Logic: Examples.
- Mid Term Exam.
- Horn Clauses; Unification and Skolemization.
- Clause Normal Form.
- Modus-Ponens and Resolution Inference Rules in Predicate Logic.
- Control Strategies for Resolution Inference (Problem Solving).

ISDE313 E-Commerce (2-2-0-0)

- **Course Definition:**

- This course is categorized as "Required" in the Information Systems department.

Course Syllabus:

- Objective: To familiarize students with the basic concepts of E-Commerce and to explain the basic applications of E-Commerce technology. To explore the principles and practice of e-commerce models and highlight the technical and social issues related to E-Marketing Plans.

Course Description:

- Describes the fundamental concepts of E-Commerce.

Learning Outcomes of the Course:

- Students will have a good understanding of the strategies and technologies available for electronic business.
- Describe the social and ethical issues related to the E-Marketing Plan and interactive television and e-marketing.

Special Skills Related to the Subject:

- Apply and use e-marketing applications.
- Demonstrate skills in using e-business models for various applications.
- Show skills in applying Content Management Systems (CMS).

Recommended Textbooks:

- "E-commerce 2020: Business, Technology, Society" by Kenneth C. Laudon and Carol Guercio Traver - This book offers a comprehensive overview of e-commerce from a business, technology, and societal perspective.

- "E-Commerce: Strategy, Technologies and Applications" by David Whiteley - Covers the concepts, strategies, technologies, and applications of e-commerce.
- "E-Commerce Essentials" by Kenneth C. Laudon and Carol Guercio Traver - Provides an easy-to-understand introduction to the fundamentals and applications of e-commerce.

Prerequisites for the Course: None Laboratory Syllabus: None

ISDC404 Data Mining (2-2-0-0)

- **Course Definition:**
- This course is categorized as "Required" in the Information Systems department.

Course Syllabus:

- Data Mining Concepts.
- Data Mining Tasks.
- Advanced Data Mining.
- Classification algorithms.
- Clustering algorithms.

Course Description:

- Methods of improving the appearance of data mining.
- Preprocessing methods.
- Association Rule Discovery.
- Sequential Pattern Discovery.
- Hypothesis-Based vs. Exploratory-Based.

Learning Outcomes of the Course:

- Acquire skills and proficiency in data mining techniques.
- Learn to distinguish between descriptive and predictive skills.
- Handle classification and clustering skills.
- Use methods for data analysis and information extraction.

Recommended Textbooks:

- "DATA MINING AND ANALYSIS, Fundamental Concepts and Algorithms" by Mohammed J. Zaki, Rensselaer Polytechnic Institute, Troy, New York, 2014.
- "data_mining_tutorial" Copyright 2014 by Tutorials Point (I) Pvt. Ltd.

Prerequisites for the Course: None

ISDE419 Mobile Applications Programming (4-2-0-2)

- **Course Definition:**
- This course is categorized as "Optional" in the Information Systems department.

Course Syllabus:

- Mobile Applications Programming: A comprehensive course designed to provide students with the necessary knowledge and skills to develop mobile applications for various platforms.

Course Description:

- Students aim to understand the fundamental concepts and principles of mobile app development. This includes understanding the complexities of platform architecture, user interface design, and the application lifecycle, which are crucial for creating effective mobile applications.

- Additionally, the critical goal is to recognize the diverse mobile device families and operating systems in use. Recognizing the differences and similarities between these platforms enables students to develop applications that meet the needs of a broad user base.

- Understanding the motivations for choosing mobile app programming and the advantages it offers is essential. This includes recognizing the personal and professional benefits of working in this field and understanding the industry demand for mobile applications.

- As challenges inevitably arise in mobile app development, students need to identify these problems and develop effective strategies to overcome them. This includes addressing the shortcomings that can be encountered in this profession.

Learning Outcomes of the Course:

- Students will gain a comprehensive understanding of the fundamental concepts and principles of mobile app development, including delving into the complexities of platform architecture, the art of designing user-friendly interfaces, and understanding the full lifecycle of mobile applications.

- Furthermore, students will become skilled in recognizing and distinguishing between different families of mobile devices and operating systems. This proficiency enables them to develop applications that meet a wide range of user needs, considering the differences and similarities between different platforms.

Recommended Textbooks:

- "Mobile Phone Programming: and its Application to Wireless Networking" by Fitzek, Frank HP, and Frank Reichert, eds., Springer Science & Business Media, 2007.

Prerequisites for the Course: None

Laboratory Syllabus:

- Introduction to Mobile App Development
- Mobile Phone Family
- Reasons for using mobile application programming
- Advantages of Mobile Application Programming
- Disadvantages of Mobile Application Programming
- The future trends of mobile application programming
- Wireless Technologies and Architectures
- Flexible Mobile Phone I
- Flexible Mobile Phone II
- Mobile App Data Management (Local Storage, Databases)
- Short-Range Communication Systems
- Mid-term Exam + Navigation Patterns in Mobile Apps
- Wireless Technologies and Architectures
- Mobile App Multimedia Integration (Images, Audio, Video)
- Device Sensors and Integration
- Mobile App Security and Privacy
- User Authentication and Authorization
- Mobile Device Management (MDM)
- Cross-Platform Development Techniques
- Location-Based Services (LBS)
- Types of Location-Based Services

- Augmented Reality (AR) and Virtual Reality (VR)
- AR in Mobile Apps
- Challenges and Considerations
- Future Trends
- Operating Systems in Mobile Application Programming
- Project Work and Presentation

UOA440 English 4 (2-2-0-0)

- **Course Definition:**
- This course is categorized as "Required" in the Information Systems department.

Course Syllabus:

- Advanced Reading Skills:
  - Analyze and interpret complex texts, including literary works, academic articles, and media sources.
  - Develop effective reading comprehension strategies, such as skimming, scanning, and note-taking.
  - Recognize main ideas, supporting details, and implicit meanings in texts.
  - Evaluate the credibility and validity of sources.
- Mastering Writing:
  - Develop advanced writing skills, including essay structure, arguments, and organization.

    - Enhance grammatical rules and sentence structures for clarity and coherence.
    - Conduct research and integrate credible sources into written work.
    - Improve editing and proofreading techniques for error-free writing.
- Oral Communication:
    - Deliver engaging and convincing presentations on various topics.
    - Participate in debates and discussions, expressing opinions and defending them.
    - Improve pronunciation, diction, and fluency in spoken English.
    - Enhance active listening skills and respond appropriately to others.
- Grammar and Vocabulary:
    - Review and reinforce advanced grammatical concepts, such as complex sentence structures, verb forms, and conditional sentences.
    -

Course Description:

- Develop advanced reading skills: Improve students' ability to understand and analyze complex texts from a variety of genres, including literary works, academic articles, and media sources. Enhance critical thinking skills by engaging with challenging materials and extracting essential information.
- Enhance writing proficiency: Improve students' writing skills to produce well-organized, coherent compositions and articles. Emphasize argument development, organization, and clarity in written work. Enhance research skills and the ability to integrate reliable sources into academic writing.

- Improve oral communication skills: Enhance students' oral proficiency and fluency through presentations, debates, and discussions. Develop effective communication strategies, including active listening and persuasive speaking, and the ability to participate in collaborative teamwork.

- Expand knowledge of grammar and vocabulary: Enhance and broaden students' understanding of advanced grammatical structures and their usage. Enhance vocabulary acquisition through learning and applying domain-specific terms and academic vocabulary relevant to various topics.

- Enhance critical thinking and analysis: Encourage students to engage in critical thinking and analysis of information from diverse sources. Develop the ability to evaluate arguments, compile information, and draw informed conclusions. Apply analytical skills to literary texts to interpret themes, literary techniques, and cultural contexts.

- Develop cultural awareness: Explore the cultural and historical contexts of literary works to develop a deeper understanding of different perspectives and cultural diversity. Enhance empathy and appreciation for diverse voices and experiences.

- Develop independent learning skills: Encourage students to take responsibility for their learning through self-directed study, research, and practice. Develop effective time management, goal setting, and self-assessment strategies to enhance lifelong learning.

Learning Outcomes of the Course:

Recommended Textbooks:

- "New Headway Plus Upper Intermediate" by Liz and John Soars, Oxford University Press, 2009.

Prerequisites for the Course: None

Laboratory Syllabus: None

UOA137 Graduation Project (12-0-0-12)

- **Course Definition:**
- This course is categorized as "Required" in the Information Systems department.

Course Syllabus:

- Project Graduate:
    - Introduction
    - Literature Review
    - Methodology
    - Data Analysis and Findings

Course Description:

- Research and Investigation: Conduct comprehensive research on a specific topic or problem related to the field of study.
- Problem Identification: Identify a real problem or challenge that requires attention or resolution.
- Objective Setting: Clearly define the objectives and goals of the graduation project, clarifying what the project aims to achieve or contribute to.
- Literature Review: Conduct a comprehensive review of existing literature, studies, and research related to the selected topic to understand the current state of knowledge in the field.
- Methodology Development: Develop an appropriate methodology or approach for studying the problem and achieving the project's objectives.

- Data Collection: Collect relevant data through surveys, experiments, observations, interviews, or other suitable methods.
- Data Analysis: Analyze the collected data using appropriate statistical or qualitative analysis techniques to draw meaningful insights and conclusions.

Learning Outcomes of the Course:

- Research Skills: Demonstrate the ability to conduct independent research, including literature review, data collection, and analysis to explore a specific topic or problem.
- Problem-Solving Skills: Identify and define a real problem or challenge, develop innovative solutions, and evaluate their feasibility and effectiveness.
- Critical Thinking: Apply critical thinking skills to evaluate existing knowledge, analyze data, and draw logical conclusions based on evidence.

Recommended Textbooks:

- "HANDBOOK OF RESEARCH METHODOLOGY," Edition 1.

Prerequisites for the Course: None

Laboratory Syllabus: None