APPENDIX A

COURSE SYLLABI

Computer Science Program

Faculty of Information Technology

2025-2026 Criteria for Accrediting Computing Programs

Indexed Edition

INDEX OF CATEGORIES

The page numbers below refer to this final indexed edition.

44 course specifications | 124 original syllabus pages | 4 indexed categories

DETAILED COURSE INDEX - FACULTY & SUPPORTIVE

DETAILED COURSE INDEX - DEPARTMENT COMPULSORY

DETAILED COURSE INDEX - SELECTIVE REQUIREMENTS

FACULTY REQUIREMENTS

COURSE SPECIFICATIONS

7 courses | Syllabi pages 7-26

Isra University

Faculty of Information Technology — Computer Science Department

2025–2026 Criteria for Accrediting Computing Programs

Course Specifications — Faculty Requirements

1. Course Number and Name

11021107 - General Physics

2. Credits (Contact Hours/Week for Fall/Spring Semester)

3 (3 Lec , 0 Lab)

3. Course Coordinator

Assigned by the Faculty of Science

4. Textbook and Supplemental Materials

Textbook: Halliday, D., Resnick, R., & Walker, J. Fundamentals of Physics, 11th ed., Wiley.

Supplemental Materials:

- Serway, R. A., & Jewett, J. W. Physics for Scientists and Engineers, 10th ed., Cengage Learning.
- Problem sets, laboratory handouts, and lecture notes provided by the instructor.

5. Course Information

Catalog Description: This course introduces the fundamental principles of classical mechanics. It covers measurement and units, kinematics, Newton's laws of motion, work and energy, momentum, and an introduction to rotational motion, providing a scientific foundation for computing students.

Prerequisites: None

Type of Course: Faculty Requirements

Learning Type: Face to Face

6. Course Objectives and Outcomes (CLOs)

Students who successfully complete this course will be able to:

- **Explain** the fundamental concepts and units of mechanics.
- **Apply** kinematic equations to describe motion.
- **Analyze** forces and motion using Newton's laws.
- **Use** the principles of work, energy, and momentum to solve problems.
- **Solve** problems involving classical mechanics.

This course supports student outcomes by developing:

Outcomes	Selected
1. Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions	✓
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline	
3. Communicate effectively in a variety of professional contexts.	

Outcomes	Selected
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.	
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline	
6. Apply computer science theory and software development fundamentals to produce computing-based solutions.	

7. Mapping between CLOs and SOs

CLO	SO1
CLO1	✓
CLO2	✓
CLO3	✓
CLO4	✓
CLO5	✓

8. List of Topics

The major topics covered in this course are:

- Measurement, units, and vectors.
- Motion in one and two dimensions.
- Newton's laws of motion.
- Work, energy, and power.
- Conservation of energy.
- Momentum and collisions.
- Introduction to rotational motion.

1. Course Number and Name

11031105 - General Mathematics

2. Credits (Contact Hours/Week for Fall/Spring Semester)

3 (3 Lec , 0 Lab)

3. Course Coordinator

Dr. Baha' Abughazaleh (Faculty of Science)

4. Textbook and Supplemental Materials

Textbook:

- Stewart, J. (2020). Calculus: Early Transcendentals (9th ed.). Cengage Learning.
- Grewal, B. S. (2020). Higher Engineering Mathematics (45th ed.). Khanna Publishers.

Supplemental Materials:

- Anton, H., Bivens, I., & Davis, S. (2016). Calculus (11th ed.). John Wiley & Sons.

Online / Open-Access Resources:

- MIT Open Course Ware – Single Variable Calculus.
- Khan Academy – Calculus.
- Paul's Online Math Notes – Calculus I & II.

5. Course Information

Catalog Description: This course introduces students to the fundamental concepts of calculus and related mathematical topics required for Information Technology. Topics include functions, complex numbers, quadratic equations, binomial theorem, sequences and series, limits, continuity, differentiation, and integration. Students develop analytical, logical, and computational problem-solving skills and apply mathematical techniques to support applications in programming, computer science, data analysis, and information technology.

Prerequisites: None

Type of Course: Faculty Requirements

Learning Type: Face to Face

6. Course Objectives and Outcomes (CLOs)

Students who successfully complete this course will be able to:

- **Explain** the concepts of functions, limits, and continuity.
- **Compute** derivatives using differentiation rules.
- **Apply** derivatives to solve optimization and rate problems.
- **Evaluate** basic integrals and their applications.
- **Solve** mathematical problems using calculus techniques.

This course supports student outcomes by developing:

Outcomes	Selected
1. Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions	✓
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline	
3. Communicate effectively in a variety of professional contexts.	
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.	
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline	
6. Apply computer science theory and software development fundamentals to produce computing-based solutions.	

7. Mapping between CLOs and SOs

CLO	SO1
CLO1	✓
CLO2	✓
CLO3	✓
CLO4	✓
CLO5	✓

8. List of Topics

The major topics covered in this course are:

Chapter 1 – Functions

- Functions and types of functions (one-one, into, onto)
- Composition and inverse of functions
- Polynomial, rational, trigonometric, exponential, and logarithmic functions
- Graphs of elementary functions

Chapter 2 – Complex Numbers and Quadratic Equations

- Complex numbers and algebraic operations
- Modulus and argument of complex numbers
- Quadratic equations in real and complex number systems

Chapter 3 – Binomial Theorem

- Binomial theorem for positive integral indices
- General term and middle term

Chapter 4 – Sequences and Series

- Arithmetic progression (A.P.)
- Geometric progression (G.P.)
- Arithmetic and geometric means

Chapter 5 – Limits and Continuity

- Limits of functions
- Properties and theorems of limits
- Limits of polynomial and rational functions
- Continuity and continuous functions

Chapter 6 – Differentiation

- Derivatives of algebraic, trigonometric, exponential, and logarithmic functions
- Chain rule and implicit differentiation
- Applications of derivatives
- Increasing and decreasing functions
- Maxima and minima

Chapter 7 – Integration

- Indefinite integration
- Standard integration formulas
- Fundamental integrals involving algebraic, trigonometric, exponential, and logarithmic functions

1. Course Number and Name

06051112 - English for IT Students

2. Credits (Contact Hours/Week for Fall/Spring Semester)

3 (3 Lec , 0 Lab)

3. Course Coordinator

Dr. Esraa Mahmoud Jamil Al Sariera

4. Textbook and Supplemental Materials

Textbook: English for Computing in Higher Education Studies 2nd Edition" by Patrick Fitzgerald & Marie McCullagh.

Supplemental Materials:

https://www.interbusiness.edu.ge/storage/books/82488578a074198b76d93885d970dcb3.pdf

5. Course Information

Catalog Description: This course aims to develop students' proficiency in English for Information Technology (IT) contexts through a combination of theoretical knowledge and practical application. The course covers essential topics including English as a tool in technology, software development basics, ethics and professional responsibility in IT, technical writing and report preparation, and effective communication in IT environments. Students will build a strong foundation in technical vocabulary related to computing, networking, cybersecurity, and emerging technologies. They will also develop their ability to read, analyze, and produce IT-related texts such as reports, emails, and documentation. The course includes a practical component where students design and deliver presentations on selected IT topics. In addition, students will prepare and present a professional self-introduction, enhancing their communication skills for academic and workplace settings. Emphasis is placed on collaborative learning, critical thinking, and real-world application, enabling students to communicate effectively in both technical and professional IT environments.

Prerequisites: None

Type of Course: Faculty Requirements

Learning Type: Hybrid

6. Course Objectives and Outcomes (CLOs)

Students who successfully complete this course will be able to:

- **Use** technical IT vocabulary accurately in context.
- **Comprehend** and summarize technical texts in computing.
- **Write** clear and well-structured technical documents.
- **Deliver** effective oral presentations on technical topics.
- **Communicate** effectively in professional and academic IT contexts.

This course supports student outcomes by developing:

Outcomes	Selected
1. Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions	
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline	

Outcomes	Selected
3. Communicate effectively in a variety of professional contexts.	✓
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.	
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline	
6. Apply computer science theory and software development fundamentals to produce computing-based solutions.	

7. Mapping between CLOs and SOs

CLO	SO3
CLO1	✓
CLO2	✓
CLO3	✓
CLO4	✓
CLO5	✓

8. List of Topics

The major topics covered in this course are:

Chapter 1 – English – Your Tool in Technology

- Explaining the importance of English in the field of Information Technology and its role in global communication
- Identifying and understanding common technical terms used in the IT industry
- Describing basic concepts related to Information Technology, such as hardware, software, data, and networking
- Using basic IT vocabulary to communicate simple ideas in different IT fields, including:
  - Software Development
  - Networking
  - Cybersecurity
  - Artificial Intelligence

Chapter 2 – English for Software Development & Problem Solving

- Explaining the concept of pseudocode and flowcharts in algorithm design
- Identifying and using common flowchart symbols correctly
- Applying sequential logic to solve simple problems using algorithms
- Defining and using variables and identifying different data types in programming
- Applying arithmetic and comparison operators in simple programming examples
- Explaining and interpreting repetition concepts using algorithms and flowcharts
- Explaining the concept of the Internet and its role in communication and services

- Describing major IT fields, including Artificial Intelligence, Cybersecurity, Cloud Computing, and Software Engineering
- Identifying different career paths in IT and the skills required for each role

Chapter 3 – Ethics and Professional Responsibility in IT

- Defining ethics in IT and distinguishing between ethical and unethical behavior in real scenarios
- Explaining the importance of privacy, data protection, and intellectual property in digital environments
- Analyzing ethical issues (e.g., piracy, data misuse, and AI decisions) and proposing appropriate solutions
- Explaining the importance of ethics and professional responsibility in IT
- Identifying common ethical and cybersecurity issues such as hacking, phishing, malware, identity theft, plagiarism, and software piracy
- Discussing ethical challenges related to Artificial Intelligence and data privacy

Chapter 4 – Technical Writing and Report Guidelines

- Explaining the purpose and characteristics of technical writing
- Identifying the standard structure of a technical report
- Writing professional and organized technical report sections using proper formatting and academic style
- Applying proper formatting and page setup for technical reports
- Using citation and referencing styles correctly while avoiding plagiarism
- Preparing a professional IT CV with appropriate structure and content

Chapter 5 – Part 1 - Microsoft Outlook

- Understanding the purpose and features of Microsoft Outlook in professional communication
- Identifying and describing the main components of the Outlook interface
- Managing emails using folders, rules, and organizational tools effectively
- Writing and sending professional emails with proper structure and formatting
- Using email features such as signatures, attachments, CC, and BCC correctly
- Reading and responding to emails using different viewing options like Reading Pane
- Managing contacts by adding, editing, and using address book features
- Scheduling and organizing events using Outlook Calendar and appointments
- Creating and tracking tasks using Outlook Task management tools
- Applying keyboard shortcuts to improve efficiency in Outlook operations

Chapter 5 – Part 2 - LinkedIn

- Understanding what LinkedIn is and its main features
- Exploring the LinkedIn interface and navigation bar
- Creating and managing a professional LinkedIn profile
- Understanding profile components and recommendations
- Using LinkedIn for networking and job searching
- Building a strong professional online presence

1. Course Number and Name

06051200 - Discrete Mathematics

2. Credits (Contact Hours/Week for Fall/Spring Semester)

3 (3 Lec , 0 Lab)

3. Course Coordinator

Dr. Yousef O. Sharrab

4. Textbook and Supplemental Materials

Textbook:

1. MIT Open Course Ware, 6.042J Mathematics for Computer Science
2. Lehman, Leighton, and Meyer, Mathematics for Computer Science (open textbook)
3. Kenneth H. Rosen, Discrete Mathematics and Its Applications, McGraw Hill

Supplemental Materials:

- MIT Open Course Ware, 6.042J Mathematics for Computer Science; Lehman, Leighton, and Meyer, Mathematics for Computer Science (open textbook); Kenneth H. Rosen, Discrete Mathematics and Its Applications, McGraw Hill.

5. Course Information

Catalog Description: An introduction to the discrete mathematical foundations used in computer science. The course develops formal logic and proof techniques and applies them to sets, functions, relations, number theory, counting, recurrence relations, graphs, trees, Boolean algebra, and discrete probability.
Prerequisites: None
Type of Course: Faculty Requirements
Learning Type: Hybrid

6. Course Objectives and Outcomes (CLOs)

Students who successfully complete this course will be able to:

- **Apply** propositional and predicate logic to formal reasoning.
- **Use** sets, functions, and relations to model problems.
- **Construct** mathematical proofs using appropriate techniques.
- **Apply** counting and combinatorial techniques to solve problems.
- **Analyze** problems using graphs and trees.

This course supports student outcomes by developing:

Outcomes	Selected
1. Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions	✓
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline	
3. Communicate effectively in a variety of professional contexts.	

Outcomes	Selected
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.	
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline	
6. Apply computer science theory and software development fundamentals to produce computing-based solutions.	

7. Mapping between CLOs and SOs

CLO	SO1
CLO1	✓
CLO2	✓
CLO3	✓
CLO4	✓
CLO5	✓

8. List of Topics

The major topics covered in this course are:

- Propositional Logic and Truth Tables
- Predicate Logic and Quantifiers
- Methods of Proof
- Mathematical Induction and Recursion
- Sets, Functions, and Relations
- Number Theory and Modular Arithmetic
- Counting Principles
- Discrete Probability
- Recurrence Relations
- Graphs and Graph Models
- Trees and Spanning Trees
- Boolean Algebra and Applications in Computing

1. Course Number and Name

06051220 - Logic Design

2. Credits (Contact Hours/Week for Fall/Spring Semester)

4 (2 Lec , 2 Lab)

3. Course Coordinator

Dr. Musab_Al-Zghoul et al.

4. Textbook and Supplemental Materials

Textbook: Floyd, T. L. (2016). Digital Fundamentals (11th ed.). Pearson/Prentice Hall.

Supplemental Materials:

- Roth, C. H., Kinney, L. L., & John, E. B. (2020). Fundamentals of Logic Design. Cengage Learning.
- Tocci, R. J., Widmer, N. S., & Moss, G. L. (2006). Digital Systems: Principles and Applications (10th ed.). Prentice Hall.
- Mano, M. M., & Ciletti, M. D. (2017). Digital Design (6th ed.). Pearson.
- Grimaldi, R. P. (2003). Discrete and Combinatorial Mathematics: An Applied Introduction (5th ed.). Addison-Wesley.

Online / Open-Access Resources:

- Course lecture slides and laboratory materials covering number systems, Boolean algebra, combinational logic, and sequential circuits.
- Manufacturer datasheets and technical manuals for logic gates, decoders, encoders, multiplexers, flip-flops, registers, and memory devices.
- Digital-circuit simulation resources for Electronic Workbench (EWB) or equivalent logic-design tools.

5. Course Information

Catalog Description: This course introduces Numbering system, computer codes, Boolean algebra and logic gates, simplification of Boolean functions, combinational logic, MSI and PLD components, synchronous clocked sequential circuits, asynchronous sequential circuits, and shift registers, memories.

Prerequisites: 11021107 - General Physics

Type of Course: Faculty Requirements

Learning Type: Hybrid

6. Course Objectives and Outcomes (CLOs)

Students who successfully complete this course will be able to:

- **Explain** number systems and Boolean algebra.
- **Apply** simplification techniques to logic functions.
- **Design** combinational logic circuits.
- **Analyze** and design sequential logic circuits.
- **Build** and test digital circuits using standard components.

This course supports student outcomes by developing:

Outcomes	Selected
1. Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions	✓
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline	
3. Communicate effectively in a variety of professional contexts.	
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.	
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline	
6. Apply computer science theory and software development fundamentals to produce computing-based solutions.	✓

7. Mapping between CLOs and SOs

CLO	SO1	SO6
CLO1	✓	
CLO2	✓	
CLO3		✓
CLO4	✓	
CLO5		✓

8. List of Topics

The major topics covered in this course are:

Chapter 1 – Introduction to Digital Logic

- Digital and analog quantities
- Binary digits, logic levels, voltage levels, and digital waveforms
- Basic logic operations and an overview of digital logic functions
- Integrated circuit terminology and basic fixed-function logic devices

Chapter 2 – Number Systems and Digital Codes

- Decimal, binary, octal, and hexadecimal number systems
- Conversion among decimal, binary, octal, and hexadecimal systems
- Binary addition and subtraction
- 1's complement, 2's complement, and signed binary numbers
- Arithmetic operations with signed binary numbers and overflow
- BCD, Gray code, ASCII, parity, and other commonly used digital codes

Chapter 3 – Logic Gates and Boolean Algebra

- Inverter, AND, OR, NAND, NOR XOR, and XNOR gates
- Truth tables, logic symbols, and Boolean expressions
- Boolean addition, Boolean multiplication, literals, and complements

- Commutative, associative, and distributive laws
- Boolean algebra rules and simplification
- DeMorgan's first and second theorems
- Universal-gate implementation using NAND-only and NOR-only circuits

Chapter 4 – Boolean Functions and Karnaugh Maps

- Analysis of combinational logic circuits
- Sum-of-Products (SOP) and Product-of-Sums (POS) forms
- Standard and canonical forms, minterms, and maxterms
- Karnaugh maps for two, three, four, and five variables
- Simplification of SOP and POS expressions
- Don't-care conditions

Chapter 5 – Arithmetic and Comparison Circuits

- Half adders and full adders
- Parallel binary adders and ripple-carry operation
- Binary subtractors and adder-subtractor circuits
- Decimal/BCD adders
- One-bit, two-bit, and four-bit magnitude comparators
- Code converters, including BCD-to-binary and binary-to-gray conversion

Chapter 6 – Combinational Logic Building Blocks

- Binary decoders and decoders enable inputs
- Active-high and active-low decoder outputs
- Implementation of Boolean functions using decoders
- Binary encoders and priority encoders
- Encoder-decoder pairs
- Multiplexers and implementation of Boolean functions using multiplexers
- Demultiplexers and data-distribution functions

Chapter 7 – Sequential Logic Circuits

- Difference between combinational and sequential logic
- Synchronous and asynchronous sequential circuits
- Clock signals, timing concepts, and clock period
- SR latches, clocked latches, and D latches
- SR, D, JK, and T flip-flops
- Characteristic tables, characteristic equations, and timing parameters

Chapter 8 – Sequential-Circuit Analysis and Design

- Analysis of sequential circuits
- Present state, next state, input, and output relationships
- State tables and state diagrams
- Mealy and Moore models
- Sequential-circuit synthesis using flip-flops
- Binary counters and counter design

Chapter 9 – Registers and Memory

- Registers and shift registers
- Serial and parallel data transfer
- Serial adders
- Basic memory concepts
- Volatile and non-volatile memory types
- ROM and programmable logic memory structures

1. Course Number and Name

06051211 - Programming Fundamentals

2. Credits (Contact Hours/Week for Fall/Spring Semester)

4 (2 Lec, 2 Lab)

3. Course Coordinator

Dr. Mohamed R. Darwish et al,

4. Textbook and Supplemental Materials

Textbook:

- Rush, E. (2022). C#: 3 Books in 1 - The Ultimate Beginner, Intermediate & Advanced Guides to Master C# Programming Quickly with No Experience.
- Doyle, B. (2015). C# Programming: From Problem Analysis to Program Design (5th ed.). Cengage Learning.

Supplemental Materials:

- Microsoft. C# Documentation. Microsoft Learn.
- Microsoft. .NET Documentation. Microsoft Learn.

5. Course Information

Catalog Description: This course introduces the Fundamental concepts of programming using C++ or Java: classes and objects, modeling object (attributes and behaviors), algorithms, problem solving flowcharts, pseudo codes. Basic blocks of programming such as variable names, data types, control structures, functions, arrays.

Prerequisites: None

Type of Course: Faculty Requirements

Learning Type: Hybrid

6. Course Objectives and Outcomes (CLOs)

Students who successfully complete this course will be able to:

- **Explain** the fundamental concepts of programming and problem solving.
- **Apply** control structures to implement program logic.
- **Design** and implement functions to build modular programs.
- **Use** arrays and basic data handling in programs.
- **Develop** structured programs to solve computational problems.

This course supports student outcomes by developing:

Outcomes	Selected
1. Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions	
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline	✓

Outcomes	Selected
3. Communicate effectively in a variety of professional contexts.	
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.	
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline	
6. Apply computer science theory and software development fundamentals to produce computing-based solutions.	✓

7. Mapping between CLOs and SOs

CLO	SO2	SO6
CLO1		✓
CLO2	✓	
CLO3	✓	
CLO4		✓
CLO5	✓	

8. List of Topics

The major topics covered in this course are:

Chapter 1 – Introduction to Programming and C#

- Computer hardware and software; system software and application software
- Software development process: analysis, design, coding, testing, and debugging
- Algorithms, pseudocode, and flowcharts
- Introduction to C#, .NET, Visual Studio, and C# console applications

Chapter 2 – Data Types, Variables, and Expressions

- Data representation, bits, bytes, and binary numbering concepts
- Identifiers, variables, constants, and C# data types
- Integral, floating-point, decimal, Boolean, character, and string types
- Assignment statements, arithmetic operators, expressions, operator precedence, and type conversion

Chapter 3 – Console Input and Output

- Displaying output using Console.Write and Console. WriteLine
- Reading characters and strings using Console. Read, Console.ReadKey, and Console.ReadLine
- Parsing and converting input strings to numeric data types

Chapter 4 – Decision and Selection Statements

- Conditional expressions, equality, relational, and logical operators
- One-way, two-way, nested, and multiple selection if statements
- switch statements and the conditional (ternary) operator

Chapter 5 – Repetition and Looping Statements

- while and do-while loops
- for loops and loop-control variables
- Nested loops, break, and continue statements

Chapter 6 – Arrays

- Declaring, creating, initializing, and accessing one-dimensional arrays
- Reading, printing, and processing arrays using for and foreach loops
- Multidimensional and jagged arrays

Chapter 7 – Methods

- Defining and calling methods
- Method parameters, return values, variables, and scope
- Passing parameters by value and by reference

1. Course Number and Name

06052102 - Data Structures

2. Credits (Contact Hours/Week for Fall/Spring Semester)

3 (3 Lec , 0 Lab)

3. Course Coordinator

Dr. Mohammad Ali H. Eljinini

4. Textbook and Supplemental Materials

Textbook:

Practical Data Structures with C++, C#, and Java. Eljinini M.A., Amazon Press, 2019.

Supplemental Materials:

- Data Structures and Algorithms using C#, Michael McMillan, Cambridge University Press.
- Data Structures and Program Design, Kruse, Leung, Tondo, Prentice Hall
- C-Sharp How-to Program, Deitel & Deitel, Prentice Hall.

Online / Open-Access Resources

- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). *Introduction to Algorithms* (4th ed.) Introduction to Algorithms. MIT Press.
- Morin, P. (2013). *Open Data Structures: An Introduction* (31st ed.) Open Data Structures. Athabasca University Press.
- Shaffer, C. A. (2013). *Data Structures and Algorithm Analysis* (3.2.0.10 ed.) OpenDSA. Virginia Tech.

5. Course Information

Catalog Description: This course covers Algorithmic problem solving, Data Structures (static & dynamic), lists, stacks, queues, graphs, and trees. Recursion and iteration. Students are expected to do lab experiments using C-Sharp or other appropriate high-level language.

Prerequisites: 06051211 - Programming Fundamentals

Type of Course: Faculty Requirements

Learning Type: Hybrid

6. Course Objectives and Outcomes (CLOs)

Students who successfully complete this course will be able to:

- **Describe** the purpose and properties of common data structures.
- **Implement** basic linear and nonlinear data structures.
- **Apply** searching and sorting techniques appropriately.
- **Analyze** alternative data structures for a given computing problem.
- **Build** efficient solutions based on sound data organization principles.

This course supports student outcomes by developing:

Outcomes	Selected
1. Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions	✓
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline	✓
3. Communicate effectively in a variety of professional contexts.	
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.	
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline	
6. Apply computer science theory and software development fundamentals to produce computing-based solutions.	✓

7. Mapping between CLOs and SOs

CLO	SO1	SO2	SO6
CLO1			✓
CLO2		✓	
CLO3		✓	
CLO4	✓		
CLO5		✓	

8. List of Topics

The major topics covered in this course are:

Lesson 1: Introduction to Data Structures

- Introduction to Data Structures
- Random Access Memory (RAM)
- Software Development
- Case Study: Library System

Lesson Videos

1. Part 1 - 1 : Intro
2. Part 1 - 2 : Software
3. Part 1 - 3: Case Study

Lesson 2: Review: IDE, C# Basics, and Problem Solving

- Introduction
- Integrated Development Environments
- Examples: Review of C# Basics
- Problem Solving

Lesson Videos

1. Part 2 - 1 IDE
2. Part 2 - 2 : Examples

Lesson 3: Reviewing Methods, Recursive Methods, Classes and Objects

- Anatomy and Types of Methods
- Writing Methods
- Recursive Methods
- Classes and Objects

Lesson Videos

1. Part 2 - 3 Writing Methods
2. Part 2 - 4 Recursion
3. Part 2 - 5 Classes and Objects

Lesson 4: Arrays of Objects

- Introduction to Arrays
- Problem Solving with Arrays
- Arrays of Objects
- The application of borrowing books

Lesson Videos

1. Part 3 - 1 Introduction to Arrays
2. Part 3 - 2 Examples
3. Part 3 - 3 Arrays of Objects
4. Part 3 - 4 Case Study

Lesson 5: Linked Lists

- Introduction to Linked lists
- Printing linked lists
- Searching Nodes in linked lists
- Delete a node from the linked list
- Problem Solving with Linked Lists

Lesson Videos

1. Part 4 -1 : Introduction to Linked Lists

Lesson 6: Doubly Linked Lists, Case Study

- Circular Doubly Linked lists
- Printing data in circular doubly linked list
- Searching for a doubly linked list
- Delete node from circular doubly linked list
- Updating the Books Borrowing Application

Lesson Videos

2. Part 4 - 2 Circular Doubly Linked lists
3. Part 4 - 3 Examples for Linked Lists
4. Part 4 - 4 Case Study on Linked Lists

Lesson 7: Stacks

- Introduction to Stacks
- Example: Decimal To Binary
- Array-based Stacks, Linked-Based Stacks
- Stacks Applications

Lesson Videos

1. Part 5-1 Introduction to Stacks
2. Part 5-2 Array-Based Stack

3. Part 5-3 Linked Based Stack
4. Part 5-4 Stacks Applications

Lesson 8: Queues

- Introduction to Queues
- Circular Queue
- Array-based Queues, Linked-Based Queues
- Queues Applications, Application of Palindromes

Lesson Videos

1. Part 1 Introduction to Queues
2. Part 2 Queue Representations
3. Part 3 Applications of Queues

Lesson 9: Trees (I)

- Introduction to Trees
- Binary Trees
- Array-Based Trees, Linked-Based Trees
- Applications of Binary trees: Heap
- Priority Queue, Heap Sort

Lesson Videos

1. Introduction to Trees
2. Applications of Trees (Part 1)

Lesson 10: Trees (II)

- Traversing Trees
- Binary Search Tree (BST)
- Searching in a BST
- An application of BST: Tree Sort

Lesson Videos

1. Applications of Trees (Part 2)

Lesson 11: Graphs (I)

- Graph theory, size of graph, degree of vertex, path & Circle
- Types of Graphs, Directed Graphs, Undirected Graphs
- Graphs Representation, Adjacency Matrix, Linked Adjacency List

Lesson Videos

1. Introduction to Graphs

Lesson 12: Graphs (II)

- Applications of Graphs
- Graph Searching

Lesson Videos

1. Application of Graphs

SUPPORTIVE REQUIREMENTS

COURSE SPECIFICATIONS

4 courses | Syllabi pages 28-38

Isra University

Faculty of Information Technology — Computer Science Department

2025–2026 Criteria for Accrediting Computing Programs

Course Specifications — Supportive Requirements

1. Course Number and Name

11031141 - Statistics and Probabilities

2. Credits (Contact Hours/Week for Fall/Spring Semester)

3 (3 Lec, 0 Lab)

3. Course Coordinator

Assigned by the Faculty of Science **(Dr. Ahmed Al-Natoor)**

4. Textbook and Supplemental Materials

Textbook:

- Raqab, M., Awad, A., and Azzam, M. (2022). *Principles of Statistics* (2nd ed.). Academic for Publishing & Distributing.
- Bluman, A. G. (2025). *Elementary Statistics: A Step By Step Approach* (12th ed.). McGraw-Hill Education.

Supplemental Materials:

- Triola, M. F. (2023). Essentials of Statistics (7th ed.). Pearson.

Online / Open-Access Resources:

- Open Intro Statistics: A comprehensive, widely adopted open-source textbook covering descriptive statistics, probability, and inference.
- Collaborative Statistics: An excellent resource for undergraduate-level statistics that aligns well with introductory course modules.
- Electronic Statistics Textbook (Stat Soft): A thorough online technical reference for both descriptive and advanced statistical topics.

5. Course Information

Catalog Description: Introduction to statistics, populations and samples, frequency distributions; measures of centrality, dispersion, skewness and kurtosis; correlation & regression; principles of probability, laws of addition and multiplication, total probability rule, Bayes rule, random variables, discrete and continuous probability distributions, binomial distribution, poisson distribution, Normal distribution.

Prerequisites: 11031105 - General Mathematics
Type of Course: Supportive Requirements
Learning Type: Hybrid

6. Course Objectives and Outcomes (CLOs)

Students who successfully complete this course will be able to:

- **Explain** the fundamental concepts of probability and statistics.
- **Apply** descriptive statistics to summarize and interpret data.
- **Compute** probabilities using discrete and continuous distributions.
- **Analyze** data using estimation and hypothesis-testing techniques.
- **Apply** statistical reasoning to computing-related problems.

This course supports student outcomes by developing:

Outcomes	Selected
1. Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions	✓
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline	
3. Communicate effectively in a variety of professional contexts.	
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.	
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline	
6. Apply computer science theory and software development fundamentals to produce computing-based solutions.	✓

7. Mapping between CLOs and SOs

CLO	SO1	SO6
CLO1		✓
CLO2	✓	
CLO3		✓
CLO4	✓	
CLO5	✓	

8. List of Topics

The major topics covered in this course are:

Chapter 1: Introduction to Statistics and Data Representation

- Populations and samples; discrete and continuous data and variables.
- Frequency tables for raw and grouped data.
- Graphical representation (Bar charts, pie charts, histograms, frequency polygons, and frequency curves).
- Symmetric and skewed data distributions.

Chapter 2: Descriptive Statistics

- Measures of centrality.
- Measures of dispersion (Range, inter-quartile range, variance, and standard deviation).
- Skewness and kurtosis.
- Chebyshev's Rule.

Chapter 3: Measures of Association and Correlation

- Z-score and coefficient of variation.
- Pearson correlation coefficient.

Chapter 4: Principles of Probability

- Random experiments and sample space.
- Probability laws (Addition, multiplication, complement, intersection, union, and DE Morgan laws).
- Conditional probability and Bayes Rule.

Chapter 5: Counting Rules and Sampling

- Permutations and combinations.
- Sampling with and without replacement.

Chapter 6: Random Variables

- Discrete and continuous random variables.
- Cumulative distribution functions (CDF) and probability density functions.
- Expected value and variance.

Chapter 7: Probability Distributions

- Binomial distribution.
- Poisson distribution.
- Normal distribution.

1. Course Number and Name

06052253 - Numeric Analysis

2. Credits (Contact Hours/Week for Fall/Spring Semester)

3 (3 Lec , 0 Lab)

3. Course Coordinator

Dr. Hegazi Ibrahim et al.

4. Textbook and Supplemental Materials

Textbook:

- Chapra, S. C., & Canale, R. P. (2021). Numerical Methods for Engineers (8th ed.). McGraw-Hill Education.
- Burden, R. L., Faires, J. D., & Burden, A. M. (2016). Numerical Analysis (10th ed.). Cengage Learning.

Supplemental Materials:

- Gerald, C. F., & Wheatley, P. O. (2003). Applied Numerical Analysis (7th ed.). Addison-Wesley.
- Kincaid, D., & Cheney, W. (2002). Numerical Analysis: Mathematics of Scientific Computing (3rd ed.). Brooks/Cole.
- Epperson, J. F. (2013). An Introduction to Numerical Methods and Analysis (2nd ed.). Wiley.

Online / Open-Access Resources:

- Atkinson, K. E. (1989). *An Introduction to Numerical Analysis* (2nd ed.). John Wiley & Sons.
- Mathews, J. H., & Fink, K. D. *Numerical Methods Using MATLAB* (4th ed.). Prentice Hall.

5. Course Information

Catalog Description: This course introduces students to fundamental numerical methods used to solve mathematical problems that cannot be solved analytically. Topics include error analysis, matrix algebra, and methods for solving systems of linear and nonlinear equations, finding roots of nonlinear equations, and approximating functions using polynomials. Students apply these techniques computationally to develop accurate and efficient problem-solving skills.

Prerequisites: 11031105 - General Mathematics

Type of Course: Supportive Requirements

Learning Type: Hybrid

6. Course Objectives and Outcomes (CLOs)

Students who successfully complete this course will be able to:

- **Explain** the sources and propagation of errors in numerical computation.
- **Apply** numerical methods to find roots of equations.
- **Use** interpolation and approximation techniques on discrete data.
- **Compute** numerical integration and differentiation.
- **Solving** systems of linear equations using numerical methods.

This course supports student outcomes by developing:

Outcomes	Selected
1. Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions	✓
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline	
3. Communicate effectively in a variety of professional contexts.	
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.	
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline	
6. Apply computer science theory and software development fundamentals to produce computing-based solutions.	

7. Mapping between CLOs and SOs

CLO	SO1
CLO1	✓
CLO2	✓
CLO3	✓
CLO4	✓
CLO5	✓

8. List of Topics

The major topics covered in this course are:

Chapter 1 – Matrices and Systems of Linear Equations

- Introduction to matrices and types of matrices (column, row, rectangular, square, diagonal, identity, null, triangular)
- Matrix operations and the matrix equation
- Solving systems of linear equations using the matrix inverse method

Chapter 2 – Root Finding for Single Nonlinear Equations

- Bisection Method (interval halving, convergence criterion, error estimation)
- Newton-Raphson Method (iterative root-finding using derivatives)
- Secant Method (iterative root-finding without derivatives)

Chapter 3 – Solving Systems of Nonlinear Equations

- Substitution Method
- Elimination Method
- Applications: parabola-line, circle-line, and circle-ellipse intersections

Chapter 5 – Polynomial Approximation and Interpolation

- Taylor Polynomials and Maclaurin Polynomials

- Lagrange Interpolation Formula

Chapter 6 – Sources of Errors

- Types of numerical errors: Absolute Error, Relative Error, Truncation Error, and Roundoff Error

1. Course Number and Name

11021202 - Physics (2)

2. Credits (Contact Hours/Week for Fall/Spring Semester)

3 (3 Lec , 0 Lab)

3. Course Coordinator

Prof. Inshad. Jum'h (Faculty of Science)

4. Textbook and Supplemental Materials

Textbook: • Serway and Jewette, "Physics for Scientists and Engineers with Modern Physics", 9th edition, 2013.

Supplemental Materials:

1. David Halliday, Robert Resnick, and J. Walker, "Fundamentals of Physics Extended, 10th Edition (J. Wiley & Sons, Ine., 2013).
2. B. Westfall, "University Physics with Modern Physics", (MeGraw Hill, 2011).
3. J. S. Walker, "Physics" Fourth Edition, (Addison-Wesley, 2010).
4. Giancoli, "Physics for Scientists &Engineers with Modern Physics", 12th Edition (Pearson, Addition Wesley, 2008).
5. A.L.Stanford andJ.M. Tanner, "Physics for students of science and engineering", Academia Press, 1985.https://doi.org/10.1016/C2013-0-07604-9.

Online / Open-Access Resources

https://jsnegroup.net/data/GENERALPHYSICS(2)-Book_9thedition-Bookphy1019th-civilianteam.pdf

5. Course Information

Catalog Description: This is the second of the physics courses that are focused on electricity and magnetism. It is a calculus-based course. Topics include electrostatic forces, Gauss's law, electric potential, capacitance, dielectrics, electric circuits, magnetic fields, and magnetic forces. Students are introduced to different physical quantities and laws governing electricity and magnetism. It is an introduction to multiple courses students will see in the future such as electric circuits, and electronics.

Prerequisites: None

Type of Course: Faculty Requirements

Learning Type: Face to Face

6. Course Objectives and Outcomes (CLOs)

Students who successfully complete this course will be able to:

- **Explain** the fundamental concepts and units of mechanics.
- **Apply** kinematic equations to describe motion.
- **Analyze** forces and motion using Newton's laws.
- **Use** the principles of work, energy, and momentum to solve problems.
- **Solve** problems involving classical mechanics.

This course supports student outcomes by developing:

Outcomes	Selected
1. Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions	✓
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline	
3. Communicate effectively in a variety of professional contexts.	
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.	
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline	
6. Apply computer science theory and software development fundamentals to produce computing-based solutions.	

7. Mapping between CLOs and SOs

CLO	SO1
CLO1	✓
CLO2	✓
CLO3	✓
CLO4	✓
CLO5	✓

8. List of Topics

The major topics covered in this course are:

Chapter 01: Electric Fields

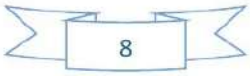
- Properties of Electric Charges
- Charging Objects by Induction
- The Electric Field
- Electric Field Lines.
- Motion of Charged Particles in a Uniform Electric Field.
- Discussion

Chapter 02: Gauss Law

- Electric Flux
- Gauss Law
- Application of Gauss Law to Various Charge Distributions.
- Conductors in Electrostatic Equilibrium

Chapter 03: Electric Potential

- Electric Potential and Potential Difference.
- Potential Difference in a Uniform Electric Field



- Electric Potential and Potential Energy Due to Point Charge.
- Obtaining the Value of the Electric Field from the Electric Potential
- Electric Potential due to Continuous Charge Distributions
- Electric Potential due to a Charged Conductor

Chapter 04: Capacitance and Dielectrics

- Definition of Capacitance
- Calculating Capacitance
- Combination of Capacitors
- Energy Stored in a Charged Capacitor
- Capacitors with Dielectric
- Electric Dipole in an Electric Field
- An Atomic Description of Dielectric

Chapter 05: Current and Resistance

- Electric Current
- Resistance
- Resistance and Temperature
- Electric Power

Chapter 06: Direct Current Circuits

- Electromotive Force
- Resistors Series and Parallel
- Kirchhoff`s Rules
- RC circuits charging

Chapter 07: Magnetic Field

- The Magnetic Fields and Forces
- Motion of a Charged Particle in a Magnetic Field
- Magnetic Force on A Current- Carrying Conductor
- Torque on a Current Loop in a Uniform Magnetic Field

Chapter 08: Sources of the Magnetic Field

- The Biot-Savart Law
- The Magnetic Force Between Two Parallel Conductors
- Amper`s Law
- The Magnetic Field of a Solenoid
- Gauss`s Law IN magnetism

Chapter 09: Faraday's Low

- Emotional emf
- Lenz Low

1. Course Number and Name

11011101 - General Chemistry 1

2. Credits (Contact Hours/Week for Fall/Spring Semester)

3 (3 Lec , 0 Lab)

3. Course Coordinator

Assigned by the Computer Science Department each semester

4. Textbook and Supplemental Materials

Textbook: Brown, T. L., LeMay, H. E., Bursten, B. E., et al. Chemistry: The Central Science, 14th ed., Pearson.

Supplemental Materials:

- Chang, R., & Goldsby, K. A. Chemistry, 13th ed., McGraw-Hill.
- Problem sets, laboratory handouts, and lecture notes provided by the instructor.

5. Course Information

Catalog Description: This course introduces the fundamental principles of general chemistry. It covers atomic structure, the periodic table, chemical bonding, stoichiometry, states of matter, solutions, and the basics of chemical reactions and energy, providing a scientific foundation for computing students.

Prerequisites: None

Type of Course: Supportive Requirements

Learning Type: Hybrid

6. Course Objectives and Outcomes (CLOs)

Students who successfully complete this course will be able to:

- **Explain** the fundamental principles of atomic structure and bonding.
- **Apply** stoichiometric relationships to chemical reactions.
- **Describe** the properties of states of matter and solutions.
- **Analyze** basic chemical reactions and their energy changes.
- **Solve** problems involving fundamental chemical concepts.

This course supports student outcomes by developing:

Outcomes	Selected
1. Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions	✓
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline	
3. Communicate effectively in a variety of professional contexts.	
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.	

Outcomes	Selected
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline	
6. Apply computer science theory and software development fundamentals to produce computing-based solutions.	

7. Mapping between CLOs and SOs

CLO	SO1
CLO1	✓
CLO2	✓
CLO3	✓
CLO4	✓
CLO5	✓

8. List of Topics

The major topics covered in this course are:

- Matter, measurement, and atomic structure.
- The periodic table and periodic properties.
- Chemical bonding and molecular structure.
- Stoichiometry and chemical equations.
- States of matter (gases, liquids, solids).
- Solutions and concentration.
- Introduction to chemical reactions and energy.

DEPARTMENT COMPULSORY

COURSE SPECIFICATIONS

22 courses | Syllabi pages 40-102

Isra University

Faculty of Information Technology — Computer Science Department

2025–2026 Criteria for Accrediting Computing Programs

Course Specifications

1. Course Number and Name

06052232 - Info. Systems Analysis & Design

2. Credits (Contact Hours/Week for Fall/Spring Semester)

3 (3 Lec , 0 Lab)

3. Course Coordinator

Prof. Radwan Al-Dwairi

4. Textbook and Supplemental Materials

Textbook: Systems Analysis and Design: An Object-Oriented Approach with UML, 6th Edition Alan Dennis, Barbara Haley Wixom, David Tegarden 2017.

Supplemental Materials:

- Valacich J., and George J., "Essentials of System Analysis and Design", 6th Edition, Pearson,2014
- SatzingeJohn W., "System Analysis & Design"; 7th Edition; Course Technology, 2015 3.
- Alan Dennis, Barbara Haley Wixom, David Tegarden, "Systems Analysis and Design: An Object-Oriented Approach with UML", 5th edition, Wiley, 2015

5. Course Information

Catalog Description: This course will introduce the concepts and methods of information systems analysis and design based on object-oriented technology. This course deals with concepts, skills, methodologies, techniques, tools, and perspectives essential for systems analysis and design. UML will be used as an analysis and design tool. Through the implementation of a term project, the course will require students to go through the steps of system analysis and design to solve a real-life business problem.

Prerequisites: 06052102 - Data Structures

Type of Course: Department Compulsory

6. Course Objectives and Outcomes (CLOs)

Students who successfully complete this course will be able to:

- **Explain** the fundamental concepts, types, and components of information systems in organizations.
- **Apply** structured and object-oriented analysis techniques to gather and document requirements.
- **Construct** process, data, and behavioral models for a given information system.
- **Analyze** organizational problems to identify feasible information-system solutions.
- **Produce** a complete requirements specification using appropriate analysis tools.

This course supports student outcomes by developing:

Outcomes	Selected
1. Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions	✓
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline	✓
3. Communicate effectively in a variety of professional contexts.	
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.	
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline	
6. Apply computer science theory and software development fundamentals to produce computing-based solutions.	

7. Mapping between CLOs and SOs

CLO	SO1	SO2
CLO1	✓	
CLO2		✓
CLO3		✓
CLO4	✓	
CLO5		✓

8. List of Topics

The major topics covered in this course are:

Chapter 1 – Introduction to Systems Analysis and Design

- Systems development life cycle– Identify the four phases– How it cameabout– Methodology alternatives
- The Unified Process & its extensions
- The Unified Modeling Language (UML)

Chapter 2 – Project Initiation & Management

- Understanding project initiation and how organizations define goals for new systems.
- Understanding how system requests are created to justify and evaluate proposed projects.
- Understanding feasibility analysis (technical, economic, organizational) in project evaluation.
- Understanding the components of a system request and their role in project justification.
- Understanding project selection and prioritization based on value, risk, and resources.
- Understanding project management tools such as WBS and Gantt charts for planning and scheduling.
- Understanding the project workplan and how it is used to track tasks, dependencies, and milestones.

- Understanding task relationships and milestones in project scheduling and control.
- Understanding how Gantt charts are used to visualize and manage project progress.

Chapter 3 – Requirements Determination

- Understanding the role of requirements determination in the SDLC and its importance in defining system needs.
- Understanding how business requirements are derived from system requests during the analysis phase.
- Understanding the difference between functional and nonfunctional requirements in system analysis.
- Understanding how requirements are documented in a requirements definition report.
- Understanding business process automation, improvement, and reengineering (BPA, BPI, BPR) in analyzing system changes.
- Understanding requirements-gathering techniques such as interviews, JAD, questionnaires, document analysis, and observation.
- Understanding how analysts and users collaborate to identify and validate system requirements.
- Understanding how requirements are analyzed, prioritized, and used to build system proposals and models.
- Understanding the role of system proposals in presenting analysis results and supporting project approval.

Chapter 4 – Use Case Diagrams

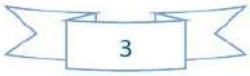
- Understanding the concept of use cases and their role in requirements modeling
- Understanding actors and their types (primary, supporting, offstage)
- Understanding how to identify actors and use cases in a system
- Understanding use case descriptions and user stories
- Understanding use case diagrams and their purpose in system modeling
- Understanding relationships between use cases (include, extend, generalization)
- Understanding how to model system behavior using use case diagrams
- Understanding the relationship between use cases and functional requirements
- Understanding activity diagrams for modeling business processes and workflows

Chapter 5 – Structural Modeling

- Understanding the purpose of structural modeling in system analysis.
- Identifying classes, attributes, operations, and relationships in a system.
- Constructing CRC cards to define class responsibilities and collaborations.
- Applying textual analysis and brainstorming techniques to identify candidate classes.
- Building UML class diagrams with correct relationships and multiplicity.
- Differentiating between association, aggregation, and generalization relationships.
- Creating and interpreting object diagrams as instances of class diagrams.
- Validating and refining structural models through an iterative development process.
- Understanding how structural models support use case realization in system design.

Chapter 6 – Behavioral Modeling

- Understanding the purpose of behavioral modeling in system analysis.
- Identifying and modeling object interactions using sequence and communication diagrams.
- Constructing UML sequence and communication diagrams correctly.
- Differentiating between sequence diagrams and communication diagrams.
- Building and interpreting behavioral state machines for object lifecycles.
- Identifying states, events, transitions, and guard conditions.
- Applying CRUDE analysis to identify complex objects.



- Validating behavioral models and ensuring consistency with other system models.

1. Course Number and Name

06052112 - Object-Oriented Techniques

2. Credits (Contact Hours/Week for Fall/Spring Semester)

3 (2 Lec , 2 Lab)

3. Course Coordinator

Dr. Khalil Barhoum

4. Textbook and Supplemental Materials

Textbook:

- Headfirst C#
  Andrew Stellman & Jennifer Greene
- O'Reilly Media, 2024
  ISBN: 9781098141783 [Course Plan...2Practical | Word]

Supplemental Materials:

- C# 11 and .NET 7 – Modern Cross-Platform Development Fundamentals
  Mark J. Price, Packt Publishing, 2022
- Pro C# 10 with .NET 6
  Andrew Troelsen & Phil Japikse, Apress, 2022 [CoursePlan...2Practical | Word]

5. Course Information

Catalog Description: This course provides a structured introduction to object-oriented programming with emphasis on both conceptual understanding and practical. Students develop a complete software system incrementally across the semester while applying object-oriented principles such as classes, encapsulation, constructors, static members, and object relationships.

The course advances to inheritance, method overloading and overriding, abstract classes, and polymorphism using both inheritance and interfaces. Through continuous design and implementation within a weakly lab assignment, students strengthen their ability to build reusable and extensible software solutions and to communicate and justify object-oriented design decisions using correct technical terminology.

Prerequisites: 06051211 - Programming Fundamentals

Type of Course: Department Compulsory

6. Course Objectives and Outcomes (CLOs)

Students who successfully complete this course will be able to:

- **Describe** the core principles of the object-oriented paradigm.
- **Design** classes and object models for a given problem domain.
- **Implement** programs using encapsulation, inheritance, and polymorphism.
- **Apply** abstraction and reuse to build modular software components.
- **Develop** object-oriented solutions that meet specified requirements.

This course supports student outcomes by developing:

Outcomes	Selected
1. Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions	
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline	✓
3. Communicate effectively in a variety of professional contexts.	
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.	
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline	
6. Apply computer science theory and software development fundamentals to produce computing-based solutions.	✓

7. Mapping between CLOs and SOs

CLO	SO2	SO6
CLO1		✓
CLO2	✓	
CLO3	✓	
CLO4		✓
CLO5	✓	

8. List of Topics

The major topics covered in this course are:

- Methods Review
- Classes and Objects
- Programmer-Defined Classes
- Instance and Local Variables
- Encapsulation
- Constructors
- Method Overloading
- Static Members (Data and Methods)
- Objects as Parameters
- Class Relationships (Dependency, Aggregation, Composition)
- Inheritance (Basic to Advanced)
- Method Overriding
- Abstract Classes
- Designing for Inheritance
- Polymorphism via Inheritance
- Polymorphism via Interfaces

1. Course Number and Name

60522010 - Algorithms

2. Credits (Contact Hours/Week for Fall/Spring Semester)

3 (3 Lec , 0 Lab)

3. Course Coordinator

Assigned by the Computer Science Department each semester

4. Textbook and Supplemental Materials

Textbook:

- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, "Introduction to Algorithms", The MIT Press, Fourth Edition. 2022.

Supplemental Materials:

- Introduction to Algorithms. T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, MIT Press, 2019.
- Analysis of Algorithms: An Active Learning Approach, Jeffrey J. McConnell, Jones, and Bartlett Publishers, 2018.
- Algorithms in C, Robert Sedgwick, Addison Wesley, 2020.

5. Course Information

Catalog Description: Fundamentals of algorithmic problem solving, fundamentals of analysis of algorithm efficiency, Divide and Conquer, space and time tradeoffs- Concentration on algorithms applications and development of algorithms.

Prerequisites: 06052102 - Data Structures

Type of Course: Department Compulsory

6. Course Objectives and Outcomes (CLOs)

Students who successfully complete this course will be able to:

- **Analyze** the time and space complexity of algorithms using asymptotic notation.
- **Apply** standard algorithm-design strategies to solve computing problems.
- **Implement** classical searching, sorting, and graph algorithms.
- **Compare** alternative algorithms for a given problem in terms of efficiency.
- **Design** correct and efficient algorithmic solutions to new problems.

This course supports student outcomes by developing:

Outcomes	Selected
1. Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions	✓
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline	
3. Communicate effectively in a variety of professional contexts.	
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.	

Outcomes	Selected
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline	
6. Apply computer science theory and software development fundamentals to produce computing-based solutions.	✓

7. Mapping between CLOs and SOs

CLO	SO1	SO6
CLO1	✓	
CLO2		✓
CLO3		✓
CLO4	✓	
CLO5		✓

8. List of Topics

The major topics covered in this course are:

- Introduction to algorithms
- Fundamentals of algorithm design
- Space complexity analysis
- Time complexity analysis
- Best case analysis
- Worst-case analysis
- Average-case analysis
- Big-O notation
- Big-Omega notation
- Big-Theta notation
- Stepping techniques
- Choosing techniques
- Looping techniques
- Recursion
- Bubble sort
- Insertion sort
- Selection sort
- Quick sort
- Merge sort
- Heap short
- Sequential search
- Binary search
- State-space search
- Depth-first search (DFS) algorithm

- Breadth-first search (BFS) algorithm
- Genetic algorithms

1. Course Number and Name

06052122 - Computer Architecture

2. Credits (Contact Hours/Week for Fall/Spring Semester)

3 (3 Lec , 0 Lab)

3. Course Coordinator

Dr Mohamed AlKhazalee

4. Textbook and Supplemental Materials

Textbook: Stallings, W. (2018). Computer Organization and Architecture: Designing for Performance (11th ed.). Pearson.

Supplemental Materials:

- Patterson, D. A., & Hennessy, J. L. (2014). Computer Organization and Design: The Hardware/Software Interface. Elsevier.
- Tanenbaum, A. S., & Austin, T. (2013). Structured Computer Organization (6th ed.). Pearson.
- Null, L., & Lobur, J. (2018). The Essentials of Computer Organization and Architecture (5th ed.). Jones & Bartlett Learning.

Online / Open-Access Resources:

- Course lecture slides and instructional materials for computer architecture, memory systems, cache, and I/O organization.
- Manufacturer processor and memory documentation, technical manuals, and architecture reference guides.

5. Course Information

Catalog Description: Hardware components of a computer system, computer abstractions and technology, IAS structure and the instruction fetch-execute cycle, history of computer architecture, computer levels architecture, top-level view of computer components, computer memory systems, computer cache memory, I/O organization.

Prerequisites: 06051220 - Logic Design

Type of Course: Department Compulsory

6. Course Objectives and Outcomes (CLOs)

Students who successfully complete this course will be able to:

- **Explain** the organization and operation of the major components of a computer system.
- **Describe** instruction set architecture and the instruction execution cycle.
- **Analyze** processor Datapath, control, and pipelining concepts.
- **Evaluate** the impact of memory hierarchy and I/O on system performance.
- **Apply** performance metrics to compare architectural alternatives.

This course supports student outcomes by developing:

Outcomes	Selected
1. Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions	✓
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline	
3. Communicate effectively in a variety of professional contexts.	
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.	
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline	
6. Apply computer science theory and software development fundamentals to produce computing-based solutions.	✓

7. Mapping between CLOs and SOs

CLO	SO1	SO6
CLO1	✓	
CLO2		✓
CLO3	✓	
CLO4	✓	
CLO5		✓

8. List of Topics

The major topics covered in this course are:

Chapter 1 – Computer Abstractions and Technology

- The computer revolution and the development of computer technology
- Classes of computers: personal computers, servers, supercomputers, and embedded systems
- The post-PC era, personal mobile devices, cloud computing, and warehouse-scale computers
- Eight great ideas in computer architecture
- Levels of program code and the relationship between application software, system software, and hardware
- Components of a computer and basic performance concepts

Chapter 2 – Computer Evolution and Architecture History

- Integrated circuits, gates, memory cells, and the basic functions of a computer
- Computer families and the IBM System/360
- DEC PDP-8 and the development of bus-based computer systems
- Evolution of semiconductor memory
- Microprocessor history and the evolution of Intel x86 processors
- Embedded computers and modern processor developments

Chapter 3 – Computer Levels and Architectures

- Basic computer components: processor, memory, input/output, and system bus
- Measures of capacity, speed, storage, and access time
- Hardware-software equivalence and abstraction
- Computer-level hierarchy and virtual machines
- Von Neumann and non-Von Neumann architectures
- Parallel, cloud, grid, cluster, and distributed computing

Chapter 4 – IAS Structure and Instruction Cycle

- Computer structure and function
- Printed circuit boards, motherboards, chips, and multicore processors
- IAS memory organization and instruction format
- IAS registers: MBR, MAR, IR, IBR, PC, AC, and MQ
- Fetch-decode-execute cycle and processor actions
- Instruction-cycle operation in the IAS computer

Chapter 5 – Top-Level View of Computer Components

- Processor, memory, and input/output modules
- Instruction cycle with and without interrupts
- Interrupt processing, interrupt handlers, and multiple-interrupt priorities
- Programmed I/O, interrupt driven I/O, and direct memory access (DMA)
- Computer interconnection structures
- System bus organization: data, address, and control buses
- Bus interconnection and point-to-point interconnection

Chapter 6 – Computer Memory Systems

- Memory hierarchy and its cost, capacity, and speed relationships
- Characteristics of computer memory systems and addressable units
- Sequential, direct, random, and associative access methods
- Memory performance parameters: access time, cycle time, and transfer rate
- Semiconductor memory organization
- Dynamic RAM (DRAM) and Static RAM (SRAM)
- ROM, PROM, EPROM, EEPROM, and flash memory
- Interleaved memory and error detection and correction

Chapter 7 – Cache Memory

- Cache-memory principles and multilevel cache organization
- Cache read operation, hit, miss, and hit ratio
- Cache lines, blocks, tags, and address decomposition
- Direct mapping
- Fully associative mapping
- Set-associative mapping
- Cache replacement algorithms and write policies
- Unified and split caches, and L1, L2, and L3 cache levels

Chapter 8 – Input/Output Organization

- I/O modules and external-device interfaces
- I/O addressing and data-transfer mechanisms
- Programmed I/O and interrupt driven I/O
- Direct memory access and I/O channels
- Interaction among processors, memory, and I/O components

1. Course Number and Name

06053113 - Visual Programming

2. Credits (Contact Hours/Week for Fall/Spring Semester)

3 (2 Lec , 2 Lab)

3. Course Coordinator

Dr Faisal Alzayood

4. Textbook and Supplemental Materials

Textbook: Price, M. J. (2023). C# 12 and .NET 8 - Modern Cross-Platform Development Fundamentals: Start Building Websites and Services with ASP.NET Core 8, Blazor, and EF Core 8 (8th ed.). Packt Publishing.

Supplemental Materials:

- Deitel, P., & Deitel, H. (2017). Visual C# How to Program. Prentice Hall.
- Deitel, P., & Deitel, H. (2017). C# 6 for Programmers (6th ed.). Prentice Hall.
- Butow, E., et al. (2002). C#: Your Visual Blueprint for Building .NET Applications. Hungry Minds.

Online / Open-Access Resources

- Microsoft Learn. C# documentation and Windows Forms documentation.
- Microsoft Learn. SQL Server, ADO.NET, Dataset, DataGrid View, and data-binding documentation.

5. Course Information

Catalog Description: Basics of visual programming, solid foundation of the syntax and semantics of a visual programming language used to develop both windows-based and web-based applications. The theoretical part emphasizes the design and creation of Graphical User Interfaces (GUI) using C#. The practical part includes lab exercises to apply these concepts through developing applications and interactive models.

Prerequisites: 06052112 - Object-oriented Methods

Type of Course: Department Compulsory

6. Course Objectives and Outcomes (CLOs)

Students who successfully complete this course will be able to:

- **Describe** the event-driven programming model and visual development environment.
- **Design** graphical user interfaces using forms and controls.
- **Implement** event handlers to create interactive applications.
- **Apply** data access techniques to connect applications to data sources.
- **Develop** a complete visual application that meets given requirements.

This course supports student outcomes by developing:

Outcomes	Selected
1. Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions	
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline	✓
3. Communicate effectively in a variety of professional contexts.	
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.	
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline	
6. Apply computer science theory and software development fundamentals to produce computing-based solutions.	✓

7. Mapping between CLOs and SOs

CLO	SO2	SO6
CLO1		✓
CLO2	✓	
CLO3	✓	
CLO4		✓
CLO5	✓	

8. List of Topics

The major topics covered in this course are:

Chapter 1 - Introduction to Visual Programming and Visual Studio

- Introduction to Visual Studio as an Integrated Development Environment (IDE)
- Creating a new Visual C# Windows Forms project
- Projects, solutions, Form1.cs, and the project physical folder
- Using Solution Explorer, Toolbox, and the Properties window
- Running, building, and locating the executable application file

Chapter 2 - Events and Event Handling

- Events and event-driven programming concepts
- Event handlers and their relationship to controls
- Default events and creating default event handlers
- Creating handlers for non-default events
- Associating the same event handler with multiple controls

Chapter 3 - Graphical User Interface Controls: Part I

- Forms, form properties, methods, and the Load event
- Buttons, labels, text boxes, and changing properties at run time

- Group Box and Panel containers
- Checkbox and Radio Button controls
- Font class and composite properties
- Message Box and Picture Box controls
- Input conversion between strings and numeric data types

Chapter 4 - Graphical User Interface Controls: Part II

- Mouse events and Mouse Event Args
- Keyboard events, Key Press Event Args, and Key Event Args
- Numeric Up Down, ToolTip, and Link Label controls
- Menu Strip, menu items, shortcut keys, and menu events
- List Box and Combo Box properties, events, and methods
- List View, Image List, columns, items, and subitems
- Tree View nodes and hierarchical structures
- Working with multiple forms

Chapter 5 - SQL Database and Data Binding

- Basic database concepts: databases, tables, fields, records, and primary keys
- Creating a Microsoft SQL Server database file
- Creating and modifying database tables
- Configuring a data source and dataset in Visual Studio
- Displaying database data using DataGrid View
- Adding, updating, deleting, and saving records
- Data binding between GUI controls and database data sources

1. Course Number and Name

06053214 - Website Design (1)

2. Credits (Contact Hours/Week for Fall/Spring Semester)

3 (2 Lec , 2 Lab)

3. Course Coordinator

Dr.Ali ALhejoj et al.

4. Textbook and Supplemental Materials

Textbook: Robbins, J. N. Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics (6th ed.). O'Reilly Media, 2025.

Supplemental Materials:

- McFedries, P. Web Design Playground: HTML & CSS the Interactive Way (2nd ed.). Manning Publications, 2024.
- Duckett, J. Web Design with HTML, CSS, JavaScript and jQuery Set. dl.acm.org, 2017.
- Deitel, P. J., & Deitel, H. M. Internet & World Wide Web - How to Program (4th ed.). Prentice Hall, 2008.

5. Course Information

Catalog Description: This course introduces the basic concepts of the Internet and Internet browsers, Internet applications, web page creation tools and languages. Basic XHTML (frames, forms), cascading style sheets, scripting and scripting languages. Dynamic XHTML (object-based programming and events). Students are required to do a Mini- project.

Prerequisites: 06051211 - Programming Fundamentals

Type of Course: Department Compulsory

6. Course Objectives and Outcomes (CLOs)

Students who successfully complete this course will be able to:

- **Describe** the technologies and standards underlying client-side web development.
- **Design** well-structured web pages using HTML and CSS.
- **Apply** responsive and usability principles to web layout.
- **Implement** interactive client-side behavior using scripting.
- **Communicate** design decisions through documented, standards-compliant web pages.

This course supports student outcomes by developing:

Outcomes	Selected
1. Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions	
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline	✓
3. Communicate effectively in a variety of professional contexts.	✓

Outcomes	Selected
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.	
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline	
6. Apply computer science theory and software development fundamentals to produce computing-based solutions.	✓

7. Mapping between CLOs and SOs

CLO	SO2	SO3	SO6
CLO1			✓
CLO2	✓		
CLO3	✓		
CLO4			✓
CLO5		✓	

8. List of Topics

The major topics covered in this course are:

Introduction

- Course Plan, Regulations, Course Webpage
- Introduction to Internet
- Lab: Explore IU e-learning platform and course webpage
- Asynchronous: Course Videos (1 point) + Discussion Forum 1 (2 points)

HTML Basics

- Learn HTML Using Notepad
- HTML Basic Example
- Lab: Write first HTML page (Hello World) using Notepad
- Asynchronous: Course Videos (1 point) – First HTML example

HTML Headings and Paragraphs

- HTML Headings and paragraphs
- HTML Editors
- Lab: Create headings & paragraphs, test in browser
- Asynchronous: Course Videos (1 point)

HTML Links and Images

- HTML Links
- HTML Images
- Lab: Add hyperlinks and images to a webpage
- Asynchronous: Course Videos (1 point)

HTML Elements and Attributes

- HTML Elements (div, span, lists)
- HTML Attributes

- Lab: Build page using div, span, lists, and attributes
- Asynchronous: Course Videos (1 point)

HTML Attributes and Styles

- HTML Attributes (id, class, style)
- HTML Styles
- Lab: Apply inline & internal CSS for styling
- Asynchronous: Course Videos + Discussion Forum 2 + Quiz #3

HTML Text Formatting and Comments

- HTML Text Formatting
- HTML Comments
- Lab: Use bold, italic, small, mark, and comments
- Asynchronous: Course Videos (1 point)

HTML Colors and CSS

- HTML Colors
- HTML CSS (Internal & External)
- Lab: Link external CSS file to HTML
- Asynchronous: Course Videos (1 point)

CSS Selectors and Colors

- CSS Selectors (element, class, id)
- CSS Colors
- Lab: Apply CSS selectors to style text & background
- Asynchronous: Course Videos (1 point)

CSS Backgrounds and Borders

- CSS Backgrounds
- CSS Borders
- Lab: Set background images, borders, padding
- Asynchronous: Course Videos + Discussion Forum 4

CSS Box Model

- CSS Padding & Margin
- CSS Box Model
- Lab: Adjust box model (margin, border, padding)
- Asynchronous: Course Videos (1 point)

JavaScript Introduction

- JavaScript Introduction
- JavaScript Syntax
- Lab: Write alert (), console.log (), and variables
- Asynchronous: Course Videos (1 point)

JavaScript Output and Statements

- JavaScript Output (document. Write, inner HTML)
- JavaScript Statements
- Lab: Use output methods and if statements
- Asynchronous: Course Videos + Discussion Forum 5

Final Project and Review

- Discuss Final Project
- Review for Final Exam
- Lab: Final project integration (HTML + CSS + JS)
- Asynchronous: No asynchronous activity

1. Course Number and Name

06014115 - Website Design (2)

2. Credits (Contact Hours/Week for Fall/Spring Semester)

3 (2 Lec , 2 Lab)

3. Course Coordinator

Dr. Abdulrahman Mohamed El-Saied

4. Textbook and Supplemental Materials

Textbook:

- David Powers, PHP 8 Solutions: Dynamic Web Design Made Easy, A press, 5th edition, 2021.
- Luke Welling, Laura Thomson, PHP and MySQL Web Development (5th edition),2016.

Supplemental Materials:

- Roben Nexion. Learning PHP MySQL JavaScript and CSS 3rd Edition.2011
- David powers PHP Solutions: Dynamic Web Design Made Easy 2nd ed. Edition
- Alan Forsythe Joy of PHP: A Beginner's Guide to Programming Interactive Web Applications with PHP and MySQL Kindle Edition, 2010
- PHP Tutorial -https://www.w3schools.com/php/

5. Course Information

Catalog Description: This unit introduces students to design, development and implementation of server-side applications, the use of multimedia and human interaction on the browser side. Students gain practical experience creating dynamic web applications that interact with a database using client-side scripts, server-side scripts and compiled server programs. Security, access rights, financial transactions and legal issues are also covered. This unit incorporates substantial practical experience in applying theoretical concepts. Students are required to submit mini projects.

Prerequisites: 06053214 - Website Design (1)

Type of Course: Department Compulsory

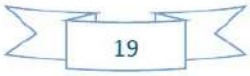
6. Course Objectives and Outcomes (CLOs)

Students who successfully complete this course will be able to:

- **Explain** the architecture of dynamic, data-driven web applications.
- **Design** server-side application logic and data models.
- **Implement** database-backed web applications using server-side scripting.
- **Apply** session management and basic security practices.
- **Communicate** the structure of a web application through documentation and demonstration.

This course supports student outcomes by developing:

Outcomes	Selected
1. Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions	
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline	✓



Outcomes	Selected
3. Communicate effectively in a variety of professional contexts.	✓
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.	
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline	
6. Apply computer science theory and software development fundamentals to produce computing-based solutions.	✓

7. Mapping between CLOs and SOs

CLO	SO2	SO3	SO6
CLO1			✓
CLO2	✓		
CLO3	✓		
CLO4			✓
CLO5		✓	

8. List of Topics

The major topics covered in this course are:

- Introduction to web platforms
- HTML and CSS fundamentals
- Introduction to PHP
- PHP variables
- PHP output statements
- PHP constants
- PHP operators
- Arithmetic and assignment operators
- Comparison operators
- Logical operators
- String concatenation
- PHP control structures
- if statements
- switch statements
- for loops
- foreach loops
- Built-in functions
- User-defined functions
- Function parameters
- Indexed arrays
- Associative arrays

- PHP superglobals
- $GLOBALS
- $_SERVER
- $_REQUEST
- $_POST
- $_GET
- Web forms and input types
- GET and POST requests
- SQL fundamentals
- MySQL databases
- Using phpMyAdmin
- PHP database management
- File handling
- Building a database-driven website

1. Course Number and Name

6053130 - Databases

2. Credits (Contact Hours/Week for Fall/Spring Semester)

3 (2 Lec , 2 Lab)

3. Course Coordinator

Dr. Faisal Yousef Alzyoud

4. Textbook and Supplemental Materials

Textbook: Ramez Elmasri, Shamkant B. Navathe. Fundamentals of Database Systems, Addison-Wesley, 7th Edition, 2017.

Supplemental Materials:

- Dr. Mukesh Negi, Fundamentals of Database Management System: Learn essential concepts of Database Systems, BPB Publications, first Edition, 2019
- Inside Oracle Designer 2000, Albert Lulushi
- Database systems, C.J. Date, 6th edition, 2000, Addison-Wesley.
- Modern Database Management, Jefferey A. Hoffer, Mary B. Prescott, Fred R. Mcfadden. Sixth Edition, Prentice Hall.
- SURVIVAL GUIDE ORACLE DBA ® Joseph B. Greene 2edOnline / Open-Access Resources
- Teach Yourself Oracle 8 In 21 Days, Sams Publishing 2000.

5. Course Information

Catalog Description: The concepts and practices necessary for design, use, and implementation of database systems. Topics covered include concepts, database architecture, relational data model, relational database constraints, relational algebra, data modeling using entity - relationship model, enhanced entity-relationship, relational database design by ER and EER - to - relational mapping, functional dependencies and normalization. Relational DB design algorithms and further dependencies, Students are required to do mini projects using SQL.

Prerequisites: 06052102 - Data Structures

Type of Course: Department Compulsory

6. Course Objectives and Outcomes (CLOs)

Students who successfully complete this course will be able to:

- **Explain** the concepts and architecture of database management systems.
- **Design** conceptual and relational data models for a given domain.
- **Apply** normalization to produce well-structured relational schemas.
- **Implement** queries and updates using SQL.
- **Develop** a database application that meets specified requirements.

This course supports student outcomes by developing:

Outcomes	Selected
1. Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions	✓
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline	✓
3. Communicate effectively in a variety of professional contexts.	
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.	
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline	
6. Apply computer science theory and software development fundamentals to produce computing-based solutions.	✓

7. Mapping between CLOs and SOs

CLO	SO1	SO2	SO6
CLO1	✓		
CLO2		✓	
CLO3			✓
CLO4		✓	
CLO5		✓	

8. List of Topics

The major topics covered in this course are:

Part 1: Introduction to Databases

- Chapter 1 – Databases and Database Users
  - ✓ Introduction to database concepts and applications
  - ✓ Types of databases and database users
  - ✓ Advantages of using a DBMS
- Chapter 2 – Database Systems Concepts and Architecture
  - ✓ Data models, schemas, and instances
  - ✓ Three-schema architecture and data independence
  - ✓ Database languages and interfaces
  - ✓ Database system environment and centralized/Client-Server architectures

Part 2: Conceptual Data Modeling and Database Design

- Chapter 3 – Data Modeling Using the Entity Relationship (ER) Model
  - ✓ Basic concepts: Entities, attributes, and relationships
  - ✓ Types of attributes (simple, composite, single-valued, multi-valued)
  - ✓ Relationship types, roles, and structural constraints (cardinality ratios, participation)
  - ✓ Weak entity types
- Chapter 4 – The Enhanced Entity Relationship (EER) Model

- ✓ Specialization/generalization and inheritance
- ✓ Constraints and characteristics of specialization/generalization
- ✓ Union types (categories)
- ✓ Modeling of complex application domains using EER

Part 3: The Relational Data Model and SQL

- Chapter 5 – The Relational Data Model and Relational Database Constraints
  - ✓ Relational model concepts (domains, attributes, tuples, relations)
  - ✓ Characteristics of relations
  - ✓ Relational constraints (domain, key, entity integrity, referential integrity)
- Chapter 6 – Basic SQL
  - ✓ Data definition language (DDL) and data manipulation language (DML) in SQL
  - ✓ Basic retrieval queries (SELECT statement)
  - ✓ Creating, modifying, and deleting tables
- Chapter 7 – More SQL: Complex Queries, Triggers, Views, and Schema Modification
  - ✓ Nested queries and joins
  - ✓ Aggregate functions and grouping
  - ✓ Views and their management
  - ✓ Triggers and assertions
- Chapter 8 – The Relational Algebra and Relational Calculus
  - ✓ Operations of relational algebra (select, project, join, etc.)
  - ✓ Tuple relational calculus
  - ✓ Domain relational calculus
- Chapter 9 – Relational Database Design by ER- and EER-to-Relational Mapping
  - ✓ Mapping ER and EER constructs to relations
  - ✓ Algorithm for relational database design

Part 4: Database Programming Techniques

- Chapter 10 – Introduction to SQL Programming Techniques
  - ✓ Embedded SQL and dynamic SQL
  - ✓ JDBC and SQLJ
- Chapter 11 – Web Database Programming Using PHP
  - ✓ Web application architecture
  - ✓ Connecting and querying databases from PHP

Part 5: Object, Object-Relational, and XML: Concepts, Models, Languages, and Standards

- Chapter 12 – Object and Object-Relational Databases
  - ✓ Object database concepts
  - ✓ Object-Relational features in SQL
- Chapter 13 – XML: Extensible Markup Language
  - ✓ XML and related technologies (DTD, Schema, XSL)
  - ✓ Querying XML (XPath, XQuery)

Part 6: Database Design Theory and Normalization

- Chapter 14 – Basics of Functional Dependencies and Normalization for Relational Databases
  - ✓ Informal design guidelines for relation schemas
  - ✓ Functional dependencies (FDs)
  - ✓ Normal forms (1NF, 2NF, 3NF, BCNF)
- Chapter 15 – Relational Database Design Algorithms and Further Dependencies
  - ✓ Algorithms for relational database schema design
  - ✓ Multivalued dependencies and 4NF
  - ✓ Join dependencies and 5NF

1. Course Number and Name

60532230 - Operating Systems

2. Credits (Contact Hours/Week for Fall/Spring Semester)

3 (3 Lec , 0 Lab)

3. Course Coordinator

Dr. Ibrahim Darwish

4. Textbook and Supplemental Materials

Textbook: Abraham Silberschatz, Greg Gagne, Peter B. Galvin, "Operating System Concepts", 10th Edition, Wiley, (May 2018).

Supplemental Materials:

- Andrews Tanenbaum, "Modern Operating Systems", 4 th edition, Pearson (March 20, 2014)
- William Stallings "Operating systems, internals and design principles", 8th editions, Pearson, (February 2, 2017)

Online / Open-Access Resources:

Operating Systems: Three Easy Pieces (OSTEP)

- A comprehensive open-access textbook covering operating system principles, virtualization, concurrency, persistence, and modern OS design.

The Linux Documentation Project (TLDP)

- Provides tutorials, guides, and references for Linux system administration and programming.

GNU Operating System Documentation

- Official documentation for GNU/Linux tools, shell programming, and system utilities.

Linux Man Pages Online

- Comprehensive reference for Linux system calls, libraries, and commands.

Open-Source Operating Systems Laboratory (OSDev Wiki)

- Educational resource for understanding operating system architecture and implementation.

MIT OpenCourseWare – Operating System Engineering

- Lecture notes, labs, and assignments from a renowned operating systems course.

5. Course Information

Catalog Description: Operating system (OS) is system software that controls all the computer's resources and provides the base upon which the application programs are built. The aim of this course is to establish principles for understanding, designing and implementing topics of several operating systems and their components. Topics include Definition of operating system, review of hardware, software and firmware, operating systems organization and structure, Process Management (processes, threads, CPU scheduling, processes synchronization, deadlocks), memory management (real storage, virtual storage), distributed computing, disk performance optimization.

Prerequisites: 06052102 - Data Structures

Type of Course: Department Compulsory

6. Course Objectives and Outcomes (CLOs)

Students who successfully complete this course will be able to:

- **Explain** the structure, services, and components of an operating system.
- **Analyze** process scheduling and synchronization mechanisms.
- **Apply** memory management and virtual memory techniques.
- **Describe** file system and storage management strategies.
- **Evaluate** operating-system design choices in terms of performance and reliability.

This course supports student outcomes by developing:

Outcomes	Selected
1. Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions	✓
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline	✓
3. Communicate effectively in a variety of professional contexts.	
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.	
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline	
6. Apply computer science theory and software development fundamentals to produce computing-based solutions.	✓

7. Mapping between CLOs and SOs

CLO	SO1	SO2	SO6
CLO1			✓
CLO2	✓		
CLO3		✓	
CLO4			✓
CLO5	✓		

8. List of Topics

The major topics covered in this course are:

Chapter 1 - Introduction to Operating System

- Computer-System Organization
- Computer-System Architecture
- Operating-System Structure
- Operating-System Operations
- Process Management
- Memory Management

- Storage Management
- Protection and Security
- Kernel Data Structures
- Computing Environments Open-Source OS

Chapter 2 - Operating-System Structures

- Operating System Services
- User Operating System Interface
- System Calls
- Types of System Calls
- System Programs
- Operating System Design and Implementation
- Operating System Structure
- Operating System Debugging
- Operating System Generation
- System Boot

Chapter 3 – Process

- Process Concept
- Process States
- Process Control Block (PCB)
- Scheduling Queues Process Scheduling
- Context Switching
- Process Creation and Termination

Chapter 4 - Threads and Multithreading

- Thread Concept
- Benefits of Multithreading
- User-Level Threads
- Kernel-Level Threads
- Multicore Programming
- Thread Libraries

Chapter 5 - CPU Scheduling

- Scheduling Criteria, FCFS, SJF, SRTF, Priority Scheduling, Round Robin
- Multilevel Queue Scheduling
- Multilevel Feedback Queue Scheduling
- Performance Evaluation

Chapter 6 - Process Synchronization

- Race Conditions
- Critical Section Problem
- Mutual Exclusion
- Peterson's Solution
- Hardware Synchronization Methods
- Atomic Operations

Chapter 7 - Synchronization Tools

- Mutex Locks
- Semaphores, Monitors, Condition Variables
- Classical Synchronization Problems (Producer-Consumer, Readers-Writers, Dining Philosophers)

Chapter 8 – Deadlocks

- System Model, Necessary Conditions for Deadlock
- Resource Allocation Graphs
- Deadlock Prevention
- Deadlock Avoidance (Banker's Algorithm)
- Detection and Recovery

Chapter 9 - Main Memory Management

- Memory Allocation
- Address Binding
- Logical vs Physical Address Space
- Swapping
- Contiguous Memory Allocation, Fragmentation, and Compaction

Chapter 11- Virtual Memory

- Demand Paging
- Page Replacement Algorithms (FIFO, Optimal, LRU, Second Chance)
- Thrashing,
- Working Set Model
- Segmentation and Paging Systems

Chapter 12 - File System Interface

- File Concepts
- Access Methods
- Directory Structures
- File Sharing, File Protection, and File Allocation Methods
- Free Space Management

1. Course Number and Name

06052221 - Computer Design and Organization

2. Credits (Contact Hours/Week for Fall/Spring Semester)

3 (3 Lec , 0 Lab)

3. Course Coordinator

Dr. Abdulrahman Mohamed El-Saied

4. Textbook and Supplemental Materials

Textbook:

- William Stallings, Computer Organization and Architecture: Designing for Performance, Pearson, 11th Edition, 2021, ISBN: 978-0135205129, https://www.pearson.com , https://williamstallings.com/ComputerOrganization/.

Supplemental Materials:

- David A. Patterson, John L. Hennessy, Computer Organization and Design: the hardware /software interface, Elsevier Inc, 2014
- Morris Mano M, Computer System Architecture, 3rd edition revised, Prentice Hall India, 2017.

5. Course Information

Catalog Description: The purpose of this course is to explore the levels of computer organization in computers, data representation and numbering systems, internal representation of signed numbers and floating-point numbers, IEEE-754 FPF, computer arithmetic and logical operations, MIPS architecture Endianness, Assembly Language, and Machine Language.

Prerequisites: 06052122 - Computer Architecture

Type of Course: Department Compulsory

6. Course Objectives and Outcomes (CLOs)

Students who successfully complete this course will be able to:

- **Explain** the organization of a computer at the register-transfer level.
- **Describe** data representation and arithmetic-logic operations.
- **Design** Datapath and control unit components.
- **Analyze** the execution of instructions across hardware components.
- **Apply** organization principles to evaluate simple processor designs.

This course supports student outcomes by developing:

Outcomes	Selected
1. Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions	✓
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline	
3. Communicate effectively in a variety of professional contexts.	

Outcomes	Selected
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.	
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline	
6. Apply computer science theory and software development fundamentals to produce computing-based solutions.	✓

7. Mapping between CLOs and SOs

CLO	SO1	SO6
CLO1	✓	
CLO2		✓
CLO3		✓
CLO4	✓	
CLO5	✓	

8. List of Topics

The major topics covered in this course are:

- Course syllabus, course overview, and objectives
- Textbook and online learning resources
- Introduction to computer design and organization
- Computer architecture and organization
- Hardware and software
- Levels of programming languages
- Translation systems
- Computer architecture versus computer organization
- Classes of computing applications and their characteristics
- Post-PC era
- Program performance
- Ideas in computer architecture
- Von Neumann model
- System bus model
- Levels of machines
- Data representation
- Number-system conversions: binary, quaternary, octal, decimal, and hexadecimal
- Binary-coded decimal (BCD)
- Signed-number representation
- Complement systems
- Floating-point numbers
- IEEE 754 floating-point standard
- Addition and subtraction operations

- Multiplication operation
- Division operation
- Computer hardware operations and operands
- MIPS architecture
- MIPS instruction formats
- MIPS addressing modes
- MIPS instructions
- Arithmetic instructions
- Logical instructions
- Immediate instructions
- Branching instructions
- Decision-making instructions
- Loop instructions
- Procedure-call instructions
- Machine-level instruction representation
- Endianness
- Loading large values
- Pseudo instructions
- System components
- Operating systems
- Linkers and loaders

1. Course Number and Name

06032250 – Basics of Software Engineering

2. Credits (Contact Hours/Week for Fall/Spring Semester)

3 (3 Lec , 0 Lab)

3. Course Coordinator

Ibrahim Abu Nahleh

4. Textbook and Supplemental Materials

Textbook:

- Software Engineering (10th Edition- 2016) Ian Sommerville, University of St Andrews, Pearson.

Supporting Textbooks

- Software Engineering: A Practitioner's Approach – Pressman – 8th Edition, McGrow-Hill
- Using UML: Software Engineering with Objects and Components – Rob Pooley – 2nd Edition, Addison Wesley
- Object-Oriented Software Engineering – Gruegge, and Dutoit – 3rd Edition, Pearson.

Online / Open-Access Resources

- Fundamentals of Software Engineering - Ghezzi, Jazayeri, and Mandrioli – 2nd Edition, Prentice Hall.

5. Course Information

Catalog Description: This course provides an overview of engineering as a discipline; the course introduces students to the fundamental principles, models and methodologies of software engineering. It covers basic knowledge about software processes. It provides minimum prerequisite knowledge for more detailed and specialized study of software engineering. Students gain experience, via a team project, of life-cycle development of software systems.

Prerequisites: 06052112 - Object-Oriented Techniques

Type of Course: Department Compulsory

6. Course Objectives and Outcomes (CLOs)

Students who successfully complete this course will be able to:

- **Explain** software process models and the phases of software development.
- **Apply** requirements engineering and design techniques to a software project.
- **Develop** test plans and apply quality-assurance practices.
- **Function** effectively as a member of a software development team.
- **Communicate** project artifacts through documentation and presentation.

This course supports student outcomes by developing:

Outcomes	Selected
1. Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions	
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline	✓
3. Communicate effectively in a variety of professional contexts.	✓
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.	
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline	✓
6. Apply computer science theory and software development fundamentals to produce computing-based solutions.	✓

7. Mapping between CLOs and SOs

CLO	SO2	SO3	SO5	SO6
CLO1				✓
CLO2	✓			
CLO3				✓
CLO4			✓	
CLO5		✓		

8. List of Topics

The major topics covered in this course are:

Chapter 1 – Introduction
- Professional software development and the software engineering definition
- Software engineering ethics and professional responsibilities
- Software engineering and the challenges of the modern world

Chapter 2 – Software Processes
- Software process models (Waterfall, Incremental, Integration and Configuration)
- Process activities and coping with change
- Process improvement and the CMMI framework

Chapter 3 – Agile Software Development
- Agile methods and the Agile Manifesto
- Extreme Programming (XP) practices
- Scrum framework and its application

Chapter 4 – Requirements Engineering
- Functional and non-functional requirements
- Requirements elicitation and analysis techniques

• Requirements specification and validation
• Requirements management

Chapter 5 – System Modeling

• Context models and system boundaries
• Interaction, structural, and behavioral models
• Model-driven engineering and its benefits

Chapter 6 – Architectural Design

• Architectural design decisions and viewpoints
• Architectural patterns (MVC, Layered, Repository, Client-Server)
• Application architectures (Transaction processing, Event-based, Language processing)

Chapter 7 – Design and Implementation

• Object-oriented design using UML
• Design patterns and their applications
• Implementation issues and software reuse
• Open-source development

Chapter 8 – Software Testing

• Development and system testing strategies
• Test automation and tools
• Test-driven development and acceptance testing

Chapter 9 – Software Evolution

• Software maintenance and evolution processes
• Legacy system management and assessment
• Software reengineering and refactoring

Part 2: System Dependability and Security

Chapter 10 – Dependable Systems

• Dependability properties (Availability, Reliability, Safety, Security)
• Socio-technical systems and their characteristics
• Dependability engineering and redundancy

Chapter 11 – Reliability Engineering

• Reliability metrics and measurements
• Software reliability modeling and prediction
• Fault-tolerant software design

Chapter 12 – Safety Engineering

• Safety-critical systems and hazard analysis
• Safety requirements and risk assessment
• Design for safety and safety cases

Chapter 13 – Security Engineering

• Security concepts and security engineering processes
• Security risk management and threat modeling
• System survivability and security patterns

Chapter 14 – Resilience Engineering

• Resilience and dependability concepts
• Diversity and redundancy for resilience
• Cyber resilience and operational resilience

1. Course Number and Name

06052153 - Computation Theory

2. Credits (Contact Hours/Week for Fall/Spring Semester)

3 (3 Lec , 0 Lab)

3. Course Coordinator

Dr. Ali Mohammad Ibrahim

4. Textbook and Supplemental Materials

Textbook: Hertz, J., Krogh, A., & Palmer, R. G. (2018). Introduction to the Theory of Neural Computation. CRC Press.

Supplemental Materials:

- Linz, P. (2016). An Introduction to Formal Languages and Automata (6th ed.). Jones & Bartlett Learning.
- Sipser, M. (2012). Introduction to the Theory of Computation (3rd ed.). Thomson Course Technology.

5. Course Information

Catalog Description: This course introduces the fundamental concepts of formal languages, automata theory, and computability. Topics include regular languages and regular expressions, deterministic and nondeterministic finite automata, conversion of NFA to DFA, minimization methods for DFA, context-free languages, pushdown automata, Turing machines, unsolvable problems, and computable functions.

Prerequisites: 06051200 - Discrete Mathematics

Type of Course: Department Compulsory

6. Course Objectives and Outcomes (CLOs)

Students who successfully complete this course will be able to:

- **Explain** the relationships among formal languages, grammar, and automata.
- **Construct** finite automata, pushdown automata, and Turing machines for given languages.
- **Apply** formal techniques to classify languages in the Chomsky hierarchy.
- **Analyze** decidability and the limits of computation.
- **Prove** properties of languages and machines using formal reasoning.

This course supports student outcomes by developing:

Outcomes	Selected
1. Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions	✓
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline	
3. Communicate effectively in a variety of professional contexts.	

Outcomes	Selected
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.	
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline	
6. Apply computer science theory and software development fundamentals to produce computing-based solutions.	✓

7. Mapping between CLOs and SOs

CLO	SO1	SO6
CLO1		✓
CLO2		✓
CLO3	✓	
CLO4	✓	
CLO5	✓	

8. List of Topics

The major topics covered in this course are:

- Introduction
- Sets and n-tuples
- Functions
- Alphabets and strings
- Predicates and quantifiers
- Proof by contradiction
- Mathematical induction
- Computation models
- Definition of regular expressions
- Operands and operators in regular expressions
- Languages associated with regular expressions
- Finite languages
- Properties of regular languages
- Formal languages and their types
- Machines for regular expressions
- Finite automata
- Transition graphs
- Nondeterministic finite automata (NFA)
- Kleene's Theorem
- Discussion of midterm examination results
- Decidability of deterministic finite automata (DFA)
- DFA minimization
- Nonregular languages

- Finite automata with output
- Moore machines
- Mealy machines
- Context-free grammar (CFGs)
- Simplification of context-free grammar
- Chomsky Normal Form
- Greibach Normal Form
- Pushdown automata
- Turing machines

1. Course Number and Name

06013175 - Modeling and Simulation

2. Credits (Contact Hours/Week for Fall/Spring Semester)

3 (3 Lec , 0 Lab)

3. Course Coordinator

Assigned by the Computer Science Department each semester

4. Textbook and Supplemental Materials

Textbook: Discrete Events System Simulation, J. Banks, J. S. Carson, B. L. Nelson and D. M. Nicol, Pearson Education, 2019.

Supplemental Materials:

- Simulation Modeling and Analysis: A. M. Law & W. D. Kelton, McGraw Hill. 3rd edition, 2018.
- Introduction to Simulation and Risk Analysis, Evans James R. and David L. Olson, Prentice Hall, 2018.

5. Course Information

Catalog Description: The course provides a comprehensive and state of art of all important aspects of modelling and simulation as a problem solving approach which focus on following topics: Introduction to Modelling and Simulation, the Nature of Simulation Models and Simulation, Monte –Carlo simulation Model, the Simulation Process, Random Events (Variates) generation, Probability and Statistics in simulation, Simulation In practice Queuing system models Output Analysis and Experimentation for System Simulation Process Model.

Prerequisites: 11031141 - Statistics and Probabilities

Type of Course: Department Compulsory

6. Course Objectives and Outcomes (CLOs)

Students who successfully complete this course will be able to:

- **Explain** the concepts of systems, models, and simulation.
- **Formulate** mathematical and logical models of real-world systems.
- **Apply** discrete-event and continuous simulation techniques.
- **Generate** and use random variates in simulation experiments.
- **Analyze** and validate simulation output to support decision making.

This course supports student outcomes by developing:

Outcomes	Selected
1. Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions	✓
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline	✓
3. Communicate effectively in a variety of professional contexts.	

Outcomes	Selected
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.	
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline	
6. Apply computer science theory and software development fundamentals to produce computing-based solutions.	

7. Mapping between CLOs and SOs

CLO	SO1	SO2
CLO1	✓	
CLO2	✓	
CLO3		✓
CLO4		✓
CLO5	✓	

8. List of Topics

The major topics covered in this course are:

- Introduction to simulation
- Real-world simulation models
- Types of simulation models
- Discrete simulation models
- Continuous simulation models
- Simulation model parameters
- Random-number generation
- Random-number generation using Excel
- Random-number generation using MATLAB
- Mid-square method
- Programming the mid-square method
- Congruential method
- Programming the congruential method
- Monte Carlo simulation
- Monte Carlo applications in business
- Graphical analysis of Monte Carlo simulations
- Mathematical models using Monte Carlo simulation
- Numerical integration using Monte Carlo simulation
- Single-server queuing models
- Graphical and tabular analysis of queuing models
- Probability simulation models

- Poisson distribution simulation
- Normal distribution simulation
- Binomial distribution simulation

1. Course Number and Name

06013256 - Smart Phone Programming

2. Credits (Contact Hours/Week for Fall/Spring Semester)

3 (2 Lec , 2 Lab)

3. Course Coordinator

Assigned by the Computer Science Department each semester

4. Textbook and Supplemental Materials

Textbook:

- Flutter Apprentice, Fourth Edition
  Kodeco Inc., 2024
  Covers Flutter widgets, Dart, data fetching, state management, and app-store deployment.
- Official Flutter Documentation
  Google, 2026
  Flutter setup, widgets, layouts, API documentation, tutorials, and learning resources.

Supplemental Materials:

- Dart Documentation
  Google, 2026
  Language guide, libraries, packages, and effective Dart practices.
- Firebase for Flutter Documentation
  Google, 2025/2026
  Flutter plugins for Firebase Authentication, Cloud Firestore, Analytics, Crashlytics, and related services.
- Beginning App Development with Flutter: Create Cross-Platform Mobile Apps
  Rap Payne, Apress, 2020

5. Course Information

Catalog Description: This course provides a structured introduction to mobile application development using Flutter and Dart. Students develop complete cross-platform mobile applications incrementally across the semester while applying mobile user-interface principles, widget-based design, navigation, state management, data handling, and platform integration. The course emphasizes practical implementation, clean code, responsive interfaces, debugging, testing, and deployment preparation for Android and iOS. The course advances asynchronous programming, REST API consumption, local storage, Firebase integration, authentication, database connectivity, accessibility, and performance optimization. Through weekly lab assignments and a final mobile application project, students strengthen their ability to design, build, test, and present reliable Flutter applications while using correct technical terminology and appropriate software-development practices.

Prerequisites: 06052112 - Object-Oriented Paradigms

Type of Course: Department Compulsory

6. Course Objectives and Outcomes (CLOs)

Students who successfully complete this course will be able to:

- **Describe** the architecture and lifecycle of mobile applications.
- **Design** mobile user interfaces following platform guidelines.
- **Implement** mobile applications that respond to user and system events.
- **Apply** local and remote data storage in mobile applications.
- **Develop** a complete smartphone application that meets given requirements.

This course supports student outcomes by developing:

Outcomes	Selected
1. Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions	
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline	✓
3. Communicate effectively in a variety of professional contexts.	
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.	
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline	
6. Apply computer science theory and software development fundamentals to produce computing-based solutions.	✓

7. Mapping between CLOs and SOs

CLO	SO2	SO6
CLO1		✓
CLO2	✓	
CLO3	✓	
CLO4		✓
CLO5	✓	

8. List of Topics

The major topics covered in this course are:

- Mobile Application Development Overview
- Flutter and Dart Ecosystem
- Development Environment Setup
- Dart Language Review
- Flutter Project Structure
- Widgets and the Widget Tree
- Stateless Widget and Stateful Widget

- Layouts: Row, Column, Stack, List View, and Grid View
- Styling, Themes, and Material Design
- User Input, Forms, and Validation
- Navigation and Routing
- State Management Fundamentals
- Asynchronous Programming: Futures and async/await
- Working with REST APIs and JSON
- Local Storage and Shared Preferences
- SQLite and Local Database Basics
- Firebase Integration
- Authentication and Cloud Fire store
- Device Features and Permissions
- Debugging, Testing, and Error Handling
- Responsive Design and Accessibility
- App Build and Deployment for Android and iOS

1. Course Number and Name

06042150 - Information Security

2. Credits (Contact Hours/Week for Fall/Spring Semester)

3 (3 Lec , 0 Lab)

3. Course Coordinator

Assigned by the Computer Science Department each semester

4. Textbook and Supplemental Materials

Textbook: Stallings, W., & Brown, L. Computer Security: Principles and Practice, 4th ed., Pearson.

Supplemental Materials:

- Whitman, M. E., & Mattord, H. J. Principles of Information Security, 7th ed., Cengage Learning.
- Security lab exercises, case studies, and lecture notes provided by the instructor.

5. Course Information

Catalog Description: Information security basics, cyberattacks, user authentication methods, basic cryptography, modern symmetric ciphers, public-key cryptosystems, key management, message authentication, digital signatures, secured software design, application security software threats, social, legal, and ethical issues.

Prerequisites: 06052140 - Computer Networks

Type of Course: Department Compulsory

6. Course Objectives and Outcomes (CLOs)

Students who successfully complete this course will be able to:

- **Explain** the fundamental goals, threats, and principles of information security.
- **Apply** cryptographic techniques to protect confidentiality and integrity.
- **Analyze** vulnerabilities and attacks against computing systems and networks.
- **Recognize** professional and ethical responsibilities in security practice.
- **Apply** access control and security management techniques to protect systems.

This course supports student outcomes by developing:

Outcomes	Selected
1. Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions	✓
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline	
3. Communicate effectively in a variety of professional contexts.	
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.	✓
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline	

Outcomes	Selected
6. Apply computer science theory and software development fundamentals to produce computing-based solutions.	✓

7. Mapping between CLOs and SOs

CLO	SO1	SO4	SO6
CLO1	✓		
CLO2			✓
CLO3	✓		
CLO4		✓	
CLO5			✓

8. List of Topics

The major topics covered in this course are:

- Introduction to information security
- Principles of secure design
- Cyberattacks and types of attacks
- User authentication
- Malicious software
- Symmetric encryption methods
- Substitution techniques
- Transposition techniques
- Product ciphers
- Simplified Data Encryption Standard (S-DES)
- Asymmetric encryption methods
- RSA encryption algorithm
- Information security mini project

1. Course Number and Name

06013176 - Artificial Intelligence

2. Credits (Contact Hours/Week for Fall/Spring Semester)

3 (3 Lec , 0 Lab)

3. Course Coordinator

Assigned by the Computer Science Department each semester

4. Textbook and Supplemental Materials

Textbook: Stuart Russell, Peter Norvig, "Artificial Intelligence: A modern approach", 4/E, Prentice Hall, 2021.

Supplemental Materials:

- "Artificial Intelligence – Structures and Strategies for Complex problem solving", George F. Luger, Pearson International Edition, Sixth edition, 2009.
- "Pattern Recognition and Machine Learning. Christopher M. Bishop." 2008.
- Python for Programmers with Introductory AI case studies Paul J. Deitel, Harvey M. Deitel 2019
- "Artificial Intelligence: A new synthesis" Nils Nilsson, Morgan Kaufmann, 1998
- Artificial Intelligence, E. Rich, McGraw Hill, 1991

5. Course Information

Catalog Description: Basic Problem-Solving Strategies, Heuristic Search, Problem Reduction & AND/OR Graphs, Knowledge Representation, Expert Systems, Generating Explanations, Uncertainty Reasoning, Game Playing, Planning, Machine Learning, pattern recognition, Computer Vision, AI Programming Languages (Python, LISP Or PROLOG), Introduction to Neural Networks, introduction to Genetic Algorithm, Introduction to Robots and intelligent agents.

Prerequisites: 06052201 - Algorithms

Type of Course: Department Compulsory

6. Course Objectives and Outcomes (CLOs)

Students who successfully complete this course will be able to:

- **Explain** the concepts of intelligent agents and the scope of artificial intelligence.
- **Apply** search techniques to solve problems and play games.
- **Represent** knowledge and perform reasoning using logical formalisms.
- **Analyze** AI techniques for a given problem and select an appropriate approach.
- **Develop** an AI-based solution using suitable methods and tools.

This course supports student outcomes by developing:

Outcomes	Selected
1. Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions	✓
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline	✓

Outcomes	Selected
3. Communicate effectively in a variety of professional contexts.	
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.	
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline	
6. Apply computer science theory and software development fundamentals to produce computing-based solutions.	✓

7. Mapping between CLOs and SOs

CLO	SO1	SO2	SO6
CLO1			✓
CLO2	✓		
CLO3			✓
CLO4	✓		
CLO5		✓	

8. List of Topics

The major topics covered in this course are:

- Introduction to artificial intelligence
- Intelligent agents
- Python applications for artificial intelligence
- Introduction to machine learning
- Classification methods
- Classification model evaluation
- Support vector machines (SVMs)
- Regression methods
- Regression model evaluation
- Clustering techniques
- Research methods in artificial intelligence
- Deep learning
- Convolutional neural networks (CNNs)
- Problem-solving through search
- Expert systems

1. Course Number and Name

06054158 – Compiler Design

2. Credits (Contact Hours/Week for Fall/Spring Semester)

3 (3 Lec , 0 Lab)

3. Course Coordinator

Dr. Ibrahim Abu Nahleh

4. Textbook and Supplemental Materials

Textbook:

- Alfred V. Aho, Monica S. Lam, Ravi Sethi, and Jeffrey D., Ullman, Compilers: Principles, Techniques, & Tools, 2nd edition, Addison-Wesley, 2007.

Supplemental Materials:

- Keith Cooper Linda Torczon, Engineering a Compiler, Morgan Kaufmann, Elsevier, 2012
- Michael L. Scott, Programming Language Pragmatics, 4th Edition, Morgan Kaufmann; 4th edition (December 25, 2015)

Online / Open-Access Resources

- Terence Parr, Language Implementation Patterns: Create Your Own Domain-Specific and General Programming Languages (Pragmatic Programmers), Pragmatic Bookshelf; 1 edition (January 10, 2010)

5. Course Information

Catalog Description: Introduction to compiling, lexical analysis, symbol tables, parsing, syntax-directed translation, type-checking, run-time organization, intermediate code generation, code optimization.

Prerequisites: 06052153 - Computation Theory

Type of Course: Department Compulsory

6. Course Objectives and Outcomes (CLOs)

Students who successfully complete this course will be able to:

- **Explain** the phases and structure of a compiler.
- **Apply** lexical analysis and parsing techniques to source programs.
- **Construct** components of a compiler using formal language tools.
- **Analyze** syntax-directed translation and semantic checking.
- **Develop** intermediate code generation for simple language constructs.

This course supports student outcomes by developing:

Outcomes	Selected
1. Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions	✓
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline	✓
3. Communicate effectively in a variety of professional contexts.	
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.	
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline	
6. Apply computer science theory and software development fundamentals to produce computing-based solutions.	✓

7. Mapping between CLOs and SOs

CLO	SO1	SO2	SO6
CLO1			✓
CLO2		✓	
CLO3			✓
CLO4	✓		
CLO5		✓	

8. List of Topics

The major topics covered in this course are:

Chapter 1 – Introduction

- Overview of compiler structure and phases
- The analysis-synthesis model of compilation
- Grouping of phases into front-end, middle-end, and back-end
- The role of the symbol table and error handling
- Evolution of programming languages and compilers

Chapter 2 – A Simple Syntax-Directed Translator

- Syntax definition using Context-Free Grammars (CFGs)
- Parse trees and ambiguity
- Associativity and precedence of operators
- Syntax-directed translation using attributes
- Implementing a simple translator (infix to postfix)
- Lexical analysis and parsing fundamentals
- Symbol table implementation

Chapter 3 – Lexical Analysis

- Role of the lexical analyzer
- Input buffering techniques
- Specification of tokens using regular expressions
- Recognition of tokens using finite automata
- From regular expressions to Non-deterministic Finite Automata (NFA)
- From NFA to Deterministic Finite Automata (DFA)
- Minimization of DFA
- Lexical analyzer generator tools (Lex/Flex)

Chapter 4 – Syntax Analysis

- Role of the parser
- Top-down parsing techniques
  - ✓ Recursive-descent parsing
  - ✓ LL(1) grammar and predictive parsing
  - ✓ First and Follow sets
  - ✓ Non-recursive predictive parsing
- Bottom-up parsing techniques
  - ✓ Handle pruning and shift-reduce parsing
  - ✓ LR parsers (SLR, Canonical LR, LALR)
  - ✓ Construction of LR parsing tables
  - ✓ Ambiguous grammar and precedence parsing (operator precedence)
- Parser generator tools (Yacc/Bison)

Chapter 5 – Syntax-Directed Translation

- Syntax-directed definitions (S-attributed and L-attributed)
- Syntax-directed translation schemes
- Implementing syntax-directed translators
- Bottom-up evaluation of inherited attributes
- Top-down translation and recursive-descent translators

1. Course Number and Name

06052157 - Computer Graphics

2. Credits (Contact Hours/Week for Fall/Spring Semester)

3 (3 Lec , 0 Lab)

3. Course Coordinator

Dr. Mohamed R. Darwish

4. Textbook and Supplemental Materials

Textbook:

- Hearn Baker Carithers, "Computer graphics with openGL", Pearson Publishing Company, 4th Edition, 2014.
- David J. E, ck, Introduction to Computer Graphics, Version 1.4 Ed., 2023, https://math.hws.edu/eck/cs424/downloads/graphicsbook-linked.pdf
- The OpenGL Programmer's Guide (the Redbook) and the OpenGL Reference Manual (The Blue book), Addison-Wesley

Supplemental Materials:

- Foley, J. D., van Dam, A., Feiner, S. K., & Hughes, J. F. Computer Graphics: Principles and Practice, 3rd ed., Addison-Wesley.
- OpenGL programming guides, lab exercises, and lecture slides provided by the instructor.

5. Course Information

Catalog Description: This course introduces the fundamentals of computer graphics, covering algorithms for rendering lines and circles, geometric transformations in 2D and 3D (translation, scaling, and rotation), and methods for creating two-dimensional views using curves and projections. Lab assignments are included to reinforce the concepts learned.

Prerequisites: 60522010 - Algorithms

Type of Course: Department Compulsory

6. Course Objectives and Outcomes (CLOs)

Students who successfully complete this course will be able to:

- **Explain** the fundamental concepts and components of computer graphics systems.
- **Apply** algorithms for drawing primitive graphics.
- **Implement** geometric transformations in two and three dimensions.
- **Apply** viewing, projection, and clipping techniques.
- **Develop** a graphics application using a graphics programming interface.

This course supports student outcomes by developing:

Outcomes	Selected
1. Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions	
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline	✓
3. Communicate effectively in a variety of professional contexts.	
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.	
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline	
6. Apply computer science theory and software development fundamentals to produce computing-based solutions.	✓

7. Mapping between CLOs and SOs

CLO	SO2	SO6
CLO1		✓
CLO2		✓
CLO3	✓	
CLO4		✓
CLO5	✓	

8. List of Topics

The major topics covered in this course are:

Chapter 1 – Overview of Computer Graphics

- A brief history of computer graphics
- Applications for Computer Graphics
- Computer Graphics Packages

Chapter 2 – Introduction to Graphic Tools

Video Display Devices

- Refresh Cathode-Ray Tube
- Raster Scan Displays
- Random-Scan Displays
- Color CRT Monitors
- Beam-Penetration Method
- Shadow-Mask Method
- Direct-View Storage Tubes
- Flat-Panel Displays

- Plasma Panel
- Thin-Film Electroluminescent Display (LED – LCD)

Chapter 3 – OpenGL- Line Drawing

- A free OpenGL utility toolkit
- Install Code blocks and Free glut
- Color and Gray Scale
- Line Attributes and Functions
- Slope-intercept formula / algorithm.
- Parametric form / algorithm.

Chapter 4 – Line Drawing Algorithms

- Digital Differential Analyzer (DDA) algorithm.
- Bresenham's line drawing algorithm.
- Computer Graphics Line Clipping

Chapter 5 – Circle and Ellipse Drawing Algorithm

- Midpoint Circle Algorithm
- Midpoint Ellipse Algorithm

Chapter 6 – Computer Graphics Polygon Filling

- Polygon Filling (Solid - pattern)
- Polygon representation
- Line Polygon Filling algorithm.
- Boundary Filling algorithm.
- Span flood Filling algorithm.

Chapter 7 – Computer Graphics Transformation in 2-D

- Translation
- Scaling
- Rotation
- Reflection
- Sharing

Chapter 8 – Computer Graphics Transformation 3-D

- Translation
- Scaling
- Rotation
- Reflection
- Sharing

1. Course Number and Name

06043162 - Network & Server Programming

2. Credits (Contact Hours/Week for Fall/Spring Semester)

3 (3 Lec , 0 Lab)

3. Course Coordinator

Assigned by the Computer Science Department each semester

4. Textbook and Supplemental Materials

Textbook: Kurose, J. F., & Ross, K. W. Computer Networking: A Top-Down Approach, 8th ed., Pearson.

Supplemental Materials:

- Stevens, W. R., Fenner, B., & Rudoff, A. M. UNIX Network Programming, Vol. 1, 3rd ed., Addison-Wesley.
- Socket-programming tutorials, lab exercises, and sample code provided by the instructor.

5. Course Information

Catalog Description: This course teaches different types of Client/Server interactions. Mainly, it is a practical course targeted towards advanced skills. The topics are DB Connectivity, Dynamic web applications using java Spring Boot, Client/Server communication using Sockets and Datagrams programming.

Prerequisites: 06052140 - Computer Networks

Type of Course: Department Compulsory

6. Course Objectives and Outcomes (CLOs)

Students who successfully complete this course will be able to:

- **Explain** the client-server model and application-layer network protocols.
- **Design** networked applications using appropriate communication models.
- **Implement** client-server applications using socket programming.
- **Apply** server-side techniques to build network services.
- **Develop** a complete networked application that meets given requirements.

This course supports student outcomes by developing:

Outcomes	Selected
1. Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions	✓
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline	✓
3. Communicate effectively in a variety of professional contexts.	
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.	

Outcomes	Selected
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline	
6. Apply computer science theory and software development fundamentals to produce computing-based solutions.	✓

7. Mapping between CLOs and SOs

CLO	SO1	SO2	SO6
CLO1	✓		
CLO2		✓	
CLO3			✓
CLO4		✓	
CLO5		✓	

8. List of Topics

The major topics covered in this course are:

- Building MySQL databases using tools such as Navicat
- Accessing remote databases
- Working with phpMyAdmin and MySQL databases
- Introduction to Java Database Connectivity (JDBC)
- JDBC drivers and database connections
- Executing SELECT commands using JDBC
- Executing INSERT, UPDATE, and DELETE commands using JDBC
- Navigating database result sets
- Traditional and modern web design
- Responsive web design
- Review of HTML and CSS fundamentals
- Introduction to Bootstrap
- Bootstrap grid system
- Bootstrap components and classes
- Static versus dynamic web programming
- Introduction to Java Spring Boot
- Spring Boot main application class
- Controller classes
- Spring annotations
- Response View and ModelAndView
- Web forms and HTML input types
- GET and POST requests
- Request forwarding and redirection
- Session tracking

- Cookies
- Executing code during application startup
- Introduction to JdbcTemplate
- Executing SQL commands using JdbcTemplate
- Validation annotations
- Displaying validation messages on HTML pages
- Introduction to client-server communication
- Connection-oriented communication
- Connectionless communication
- Sockets and datagrams
- Socket-based client-server communication
- Datagram-based client-server communication

1. Course Number and Name

06052140 - Computer Networks

2. Credits (Contact Hours/Week for Fall/Spring Semester)

3 (3 Lec , 0 Lab)

3. Course Coordinator

Assigned by the Computer Science Department each semester

4. Textbook and Supplemental Materials

Textbook:

- James, F. K. (2025). Computer networking.

https://read.amazon.com/sample/B0FP1SXZV3?clientId=share.

Supplemental Materials:

- Andrew S. Tanenbaum, Nick Feamster, and David J. Wetherall, Computer Networks, Pearson, sixth edition, 2020.

Online / Open-Access Resources:

- Behrouz A. Forouzan, "Data Communications and Networking", 5th Edition, McGraw Hill, 2008.

5. Course Information

Catalog Description: Logical and physical of computer networks, architecture and transmission alternatives. OSI-reference model, ALOHA protocol, CSMA protocols, LAN, IEEE standards and protocols (token ring, token bus and Ethernet), physical layer basics, network layer in TCP/IP model, Internet Protocol (IP), IPv4 protocol, IPv4 addressing, IPv4 classificatory addressing, NetID and Host_ID, networking, default mask, grouping of devices in networks, routing table (Routing Information Base (RIB)).

Prerequisites: 6053223 - Operating Systems

Type of Course: Department Compulsory

6. Course Objectives and Outcomes (CLOs)

Students who successfully complete this course will be able to:

- **Explain** the layered architecture and protocols of computer networks.
- **Analyze** the functions of the data link, network, and transport layers.
- **Apply** addressing and routing concepts to network design.
- **Describe** common network applications and services.
- **Evaluate** network configurations in terms of performance and reliability.

This course supports student outcomes by developing:

Outcomes	Selected
1. Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions	✓
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline	✓
3. Communicate effectively in a variety of professional contexts.	
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.	
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline	
6. Apply computer science theory and software development fundamentals to produce computing-based solutions.	✓

7. Mapping between CLOs and SOs

CLO	SO1	SO2	SO6
CLO1			✓
CLO2	✓		
CLO3		✓	
CLO4			✓
CLO5	✓		

8. List of Topics

The major topics covered in this course are:

Chapter 1 – Computer Networks and the Internet

- What is the Internet? (A nuts-and-bolts and service description)
- Network edge (end systems, access networks, physical media)
- Network core (packet switching, circuit switching, network structure)
- Delay, loss, and throughput in packet-switched networks
- Protocol layers and their service models (OSI and TCP/IP architecture)
- History of computer networking and the Internet

Chapter 2 – Application Layer

- Principles of network applications
- The Web and HTTP (non-persistent and persistent)
- Electronic mail (SMTP, POP3, IMAP)
- DNS (The Domain Name System)
- Peer-to-Peer (P2P) file distribution (BitTorrent)
- Video streaming and content distribution networks (CDNs)
- Socket programming with TCP and UDP

Chapter 3 – Transport Layer
- Introduction and transport-layer services
- Multiplexing and demultiplexing
- Connectionless transport: UDP
- Principles of reliable data transfer (rdt 1.0, 2.0, 2.1, 2.2, 3.0)
- Connection-oriented transport: TCP (segment structure, reliable transfer, flow control, connection management)
- Principles of congestion control
- TCP congestion control (AIMD, Tahoe, Reno, CUBIC)
- Fairness and explicit congestion notification (ECN)

Chapter 4 – The Network Layer: Data Plane
- Overview of the network layer (forwarding and routing)
- Router architecture and switching fabrics
- IPv4 and IPv6 addressing (subnetting, CIDR, NAT)
- The Internet Protocol (IP)
- Generalized forwarding and Software-Defined Networking (SDN)

Chapter 5 – The Network Layer: Control Plane
- Routing algorithms (LS and DV algorithms)
- Internet routing protocols: RIP, OSPF, BGP
- ICMP and network management
- SDN control plane and network management configuration
- Middleboxes and future of the network layer

Chapter 6 – The Link Layer and Local Area Networks
- Introduction and services of the link layer
- Error-detection and correction techniques (parity, checksum, CRC)
- Multiple access protocols (TDMA, FDMA, ALOHA, CSMA/CD, CSMA/CA)
- LANs and Ethernet (MAC addresses, frame structure)
- Link-layer switches and their operation
- Wireless LANs and personal area networks (Bluetooth, NFC)
- Virtualization: VLANs, overlay networks, SDN

1. Course Number and Name

06054190 - Practical Training

2. Credits (Contact Hours/Week for Fall/Spring Semester)

3 (0 Lec , 6 Lab)

3. Course Coordinator

Assigned by the Computer Science Department each semester

4. Textbook and Supplemental Materials

Textbook: No required textbook; the host organization's documentation and standards serve as the primary reference.

Supplemental Materials:

- Departmental practical training guidelines and report templates.
- Professional and ethics guidelines, and supervisor-provided materials.
- Practical Training Manual

5. Course Information

Catalog Description: This course aims to provide the possibility of training on the use of information technology and its applications in various fields; training is carried out in public or private sector institutions and under the supervision of faculty members in the department. The purpose of the supervised field training is to accumulate the knowledge, experience and skills provided during the academic part of the program in a practical environment. Field training aims to provide students with practical experience and the ability to contribute to work and to the training site. Field training is expected to provide learning opportunities not available in the classroom.

Prerequisites: Departmental approval (completion of required credit hours: Pass 90 Credit Hrs.)

Type of Course: Department Compulsory

6. Course Objectives and Outcomes (CLOs)

Students who successfully complete this course will be able to:

- **Apply** academic knowledge and skills to real-world computing tasks.
- **Function** effectively within a professional work environment.
- **Communicate** training activities and outcomes through reports and presentations.
- **Recognize** professional responsibilities and workplace ethics.
- **Reflect** on the training experience to identify areas for growth.

This course supports student outcomes by developing:

Outcomes	Selected
1. Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions	✓
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline	

Outcomes	Selected
3. Communicate effectively in a variety of professional contexts.	✓
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.	✓
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline	✓
6. Apply computer science theory and software development fundamentals to produce computing-based solutions.	

7. Mapping between CLOs and SOs

CLO	SO1	SO3	SO4	SO5
CLO1	✓			
CLO2				✓
CLO3		✓		
CLO4			✓	
CLO5		✓		

8. List of Topics

The major topics covered in this course are:

- Orientation to the training organization.
- Workplace policies, ethics, and professionalism.
- Application of computing skills to assigned tasks.
- Teamwork and professional communication.
- Documentation of training activities.
- Reporting and presentation of outcomes.
- Reflection and self-assessment.

1. Course Number and Name

06054191 - Graduation Project-CS

2. Credits (Contact Hours/Week for Fall/Spring Semester)

3 (0 Lec , 6 Lab)

3. Course Coordinator

Assigned by the Computer Science Department each semester

4. Textbook and Supplemental Materials

Textbook: No required textbook; references are selected according to the chosen project topic.

Supplemental Materials:

- Sommerville, I. Software Engineering, 10th ed., Pearson (as a general reference).
- Departmental graduation-project handbook, report templates, and supervisor-provided resources.

5. Course Information

Catalog Description: This capstone course requires students to plan, design, implement, and present a substantial computing project. Working in teams under supervision, students integrate knowledge from across the program, apply professional practices, and deliver a complete solution with full documentation.

Prerequisites: Pass 90 Credit Hrs.

Type of Course: Department Compulsory

6. Course Objectives and Outcomes (CLOs)

Students who successfully complete this course will be able to:

- **Analyze** a substantial computing problem and define its requirements.
- **Design** and implement a complete computing-based solution.
- **Function** effectively as a member or leader of a project team.
- **Communicate** the project through documentation and a formal presentation.
- **Recognize** professional, ethical, and societal considerations in the project.

This course supports student outcomes by developing:

Outcomes	Selected
1. Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions	✓
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline	✓
3. Communicate effectively in a variety of professional contexts.	✓
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.	✓

Outcomes	Selected
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline	✓
6. Apply computer science theory and software development fundamentals to produce computing-based solutions.	✓

7. Mapping between CLOs and SOs

CLO	SO1	SO2	SO3	SO4	SO5	SO6
CLO1	✓					
CLO2		✓				
CLO3					✓	
CLO4			✓			
CLO5				✓		

8. List of Topics

The major topics covered in this course are:

- Project identification and proposal.
- Requirements analysis and planning.
- System design and architecture.
- Implementation and integration.
- Testing, evaluation, and quality assurance.
- Documentation and reporting.
- Final presentation and defense.

SELECTIVE REQUIREMENTS

COURSE SPECIFICATIONS

11 courses | Syllabi pages 104-133

Isra University

Faculty of Information Technology — Computer Science Department

2025–2026 Criteria for Accrediting Computing Programs

Course Specifications — Selective (Elective) Requirements

1. Course Number and Name

06013231 - Database Systems Management

2. Credits (Contact Hours/Week for Fall/Spring Semester)

3 (3 Lec , 0 Lab)

3. Course Coordinator

Assigned by the Computer Science Department each semester

4. Textbook and Supplemental Materials

Textbook: Ramez Elmasri, Shamkant B. Navathe, Fundamentals of Database Systems, 7th Edition, Addison-Wesley. 2018.

Supplemental Materials:

- Elmasri, R., & Navathe, S. B. Fundamentals of Database Systems, 7th ed., Pearson.
- DBMS administration guides, lab exercises, and lecture notes provided by the instructor.
- C.J. Date, Database systems, 7th edition, Addison-Wesley

5. Course Information

Catalog Description: Introduction to Database Systems Management, Database Systems components and architecture. Data Protection Problems, Recovery, Concurrency, Security, and Data Integrity. Database administration and tuning. All practical applications are implemented in PLSQL.

Prerequisites: 06053130 - Databases

Type of Course: Selective Requirements

6. Course Objectives and Outcomes (CLOs)

Students who successfully complete this course will be able to:

- **Explain** advanced database architecture and management concepts.
- **Apply** query optimization and indexing techniques to improve performance.
- **Manage** transactions, concurrency, and recovery in a database system.
- **Analyze** database security and integrity requirements.
- **Develop** administration procedures for a relational database system.

This course supports student outcomes by developing:

Outcomes	Selected
1. Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions	✓
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline	✓

Outcomes	Selected
3. Communicate effectively in a variety of professional contexts.	
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.	
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline	
6. Apply computer science theory and software development fundamentals to produce computing-based solutions.	✓

7. Mapping between CLOs and SOs

CLO	SO1	SO2	SO6
CLO1	✓		
CLO2			✓
CLO3		✓	
CLO4	✓		
CLO5		✓	

8. List of Topics

The major topics covered in this course are:

- Introduction to databases
- Creating databases using Microsoft Access
- Database management system (DBMS) components
- Organizational data versus personal data
- Database models
- Three-schema DBMS architecture
- Oracle Live SQL
- Introduction to database transactions
- Transaction and system concepts
- Creating tables and executing SELECT statements
- Transaction properties
- Transaction execution states
- PL/SQL data types and variables
- Causes of transaction failure
- Database log files
- PL/SQL conditional statements
- Recovery using log files
- Commit points and rollback operations
- Transaction schedules
- Recoverable schedules

- Cascadeless and strict schedules
- Serializable schedules
- PL/SQL loops
- Concurrency-control principles
- Locking and two-phase locking protocols
- Types of two-phase locking protocols
- Database deadlocks
- Deadlock types
- Deadlock detection and resolution
- PL/SQL cursors
- Built-in and explicit cursors
- Timestamp-based protocols
- Database recovery systems
- Database storage structures
- Log-based recovery

1. Course Number and Name

06014254 - Information Systems Retrieval

2. Credits (Contact Hours/Week for Fall/Spring Semester)

3 (3 Lec , 0 Lab)

3. Course Coordinator

Assigned by the Computer Science Department each semester

4. Textbook and Supplemental Materials

Textbook: Modern Information Retrieval Ricardo Baeza-Yates, Berthier Ribeiro-Neto,, Addison Wesley.

Supplemental Materials:

- Salton and Buckley, "Improving Retrieval Performance by Relevance Feedback". In Readings in Information Retrieval, ed. Karen Sparck Jones and Peter Willett, Salton's "Automatic Text Processing" by Addison-Wesley
- Frakes and Baeza-Yates's "Information Retrieval: Data Structures and algorithms " from Prentice-Hall.
- Christopher D. Manning Prabhakar Raghavan, Hinrich Schütze "An Introduction to Information Retrieval" Cambridge University Press Cambridge, England, 2009

5. Course Information

Catalog Description: This course covers the fundamentals of Information Retrieval Systems, as well as recent advances in Information Retrieval (IR), the study of indexing, preprocessing, and querying textual data, in addition to Search Engines and web search.

Prerequisites: 06053130 - Databases

Type of Course: Selective Requirements

6. Course Objectives and Outcomes (CLOs)

Students who successfully complete this course will be able to:

- **Explain** the concepts and components of information retrieval systems.
- **Apply** indexing and retrieval models to text collections.
- **Implement** ranking and probabilistic retrieval techniques.
- **Analyze** the effectiveness of retrieval systems using evaluation measures.
- **Develop** a simple information retrieval application.

This course supports student outcomes by developing:

Outcomes	Selected
1. Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions	✓
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline	✓
3. Communicate effectively in a variety of professional contexts.	

Outcomes	Selected
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.	
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline	
6. Apply computer science theory and software development fundamentals to produce computing-based solutions.	✓

7. Mapping between CLOs and SOs

CLO	SO1	SO2	SO6
CLO1			✓
CLO2	✓		
CLO3		✓	
CLO4	✓		
CLO5		✓	

8. List of Topics

The major topics covered in this course are:

- Introduction to information retrieval systems
- Components of classical information retrieval systems
- Text-processing operations
- Stemming algorithms
- Index-term selection
- Stop-word elimination
- User interfaces for information retrieval
- Information retrieval models
- Boolean retrieval model
- Vector-space model
- Similarity measurements
- Information retrieval evaluation
- Indexing and searching
- Query operations
- Query languages
- Web search
- Google Search
- Datasets and corpora
- Arabic information retrieval
- Arabic-language processing issues
- Search engine optimization (SEO)

1. Course Number and Name

06033274 - Human Computer Interface

2. Credits (Contact Hours/Week for Fall/Spring Semester)

3 (3 Lec , 0 Lab)

3. Course Coordinator

Dr. Ibrahim Abu Nahleh

4. Textbook and Supplemental Materials

Textbook: J. Ferry. Human-Computer Interaction. Advanced Guide: Mastering the Art of User-Centered Design. An Advanced Guide to Human-Computer Interaction. Kindle Edition,2024.

Supplemental Materials:

- B. Albert & T. Tullis. Measuring User Experience: Collecting, Analyzing, and Presenting UX Metrics (Interactive Technologies) 3rd Ed. Morgan Kaufmann, 2022.

Online / Open-Access Resources

- S. MacKenzie. Human-Computer Interaction: An Empirical Research Perspective 2nd Edition, Kindle Edition,2024.

5. Course Information

Catalog Description: The study of human-computer interaction enables system architects to design useful, efficient, and enjoyable computer interfaces. This course teaches the theory, design procedure, and programming practices behind effective human interaction with computers. The students well receive knowledge about interaction design, implementation, and evaluation. The design process requires a solid understanding of the theory behind successful human-computer interaction, as well as an awareness of established procedures for good user interface design, including the 'usability engineering' process. Iterative evaluation is an important aspect of this procedure, and we will learn and practice prototyping and evaluation using scenario-based case studies. Students will apply their gained knowledge in a series of practical assignments to create successful user interfaces.

Prerequisites: 06053113 - Visual Programming

Type of Course: Selective Requirements

6. Course Objectives and Outcomes (CLOs)

Students who successfully complete this course will be able to:

- **Explain** the principles of human-computer interaction and usability.
- **Apply** user-centered design methods to interface design.
- **Design** and prototype interactive user interfaces.
- **Evaluate** interfaces using appropriate usability methods.
- **Communicate** design decisions and findings to stakeholders.

This course supports student outcomes by developing:

Outcomes	Selected
1. Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions	
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline	✓
3. Communicate effectively in a variety of professional contexts.	✓
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.	
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline	✓
6. Apply computer science theory and software development fundamentals to produce computing-based solutions.	

7. Mapping between CLOs and SOs

CLO	SO2	SO3	SO5
CLO1	✓		
CLO2	✓		
CLO3	✓		
CLO4			✓
CLO5		✓	

8. List of Topics

The major topics covered in this course are:

Part I: Evaluation and Testing

Chapter 1 – Usability Testing

- What is Usability Testing?
- Planning and Conducting Usability Tests
- Analyzing and Reporting Usability Test Results
- Common Usability Testing Methods

Chapter 2 – Evaluation Methods

- Heuristic Evaluation
- Cognitive Walkthroughs
- A/B Testing
- Surveys and Questionnaires

Chapter 3 – Metrics and Analytics

- Key Performance Indicators for HCI
- Data Collection and Analysis Techniques

- Using Analytics to Improve Design
- Case Studies in Metrics and Analytics

Part II: Advanced Topics in HCI

Chapter 4 – Accessibility in HCI

- Principles of Accessible Design
- Legal and Ethical Considerations
- Designing for Diverse Populations
- Tools and Techniques for Accessibility Testing

Chapter 5 – Interaction Beyond the Screen

- Gesture-Based Interfaces
- Voice User Interfaces (VUIs)
- Virtual and Augmented Reality
- Wearable Technology

Chapter 6 – Emotional and Affective Computing

- Understanding Emotional Responses
- Designing for Positive User Experience
- Tools for Measuring User Emotions
- Case Studies in Affective Computing

Part III: HCI in Practice

Chapter 7 – HCI in Different Domains

- HCI in Education
- HCI in Healthcare
- HCI in Gaming
- HCI in E-commerce

Chapter 8 – The Future of HCI

- Emerging Trends and Technologies
- Ethical Considerations and Challenges
- The Role of AI and Machine Learning in HCI
- Preparing for the Future of Interaction Design

Conclusion

Chapter 9 – Recap and Next Steps

- Summary of Key Points
- How to Continue Learning HCI
- Resources for Further Study
- Building a Career in HCI

1. Course Number and Name

06043273 - Advanced Programming

2. Credits (Contact Hours/Week for Fall/Spring Semester)

4 (2 Lec , 2 Lab)

3. Course Coordinator

Assigned by the Computer Science Department each semester

4. Textbook and Supplemental Materials

Textbook: Introduction to Java Programming, Daniel Liang, 10th edition, Person Education, 2016.

Supplemental Materials:

- Java: How to Program, Deitel & Deitel,2014, 11th ed -- ISBN: 9780134751856.
- Java- The complete Reference, 11th edition, Herbert Schildt, McGraw Hill Education.
- Coding Basics for Beginners: The Smart Way to Approach the World of Computer Programming and the Fundamental Functions of the Most Popular Languages Such as Python, Java and C++, by Jefferson Sandyman | Jun 21, 2021.

5. Course Information

Catalog Description: Advanced OOP features such as servlet programming, Database connectivity, remote method invocation, and GUI. Other major topics in this course include network programming serialization, properties, multithreading, security, the collection classes and architecture.

Prerequisites: 06053113 - Visual Programming

Type of Course: Selective Requirements

6. Course Objectives and Outcomes (CLOs)

Students who successfully complete this course will be able to:

- **Describe** advanced programming constructs and language features.
- **Apply** design patterns and best practices in software construction.
- **Implement** applications using generics, collections, and concurrency.
- **Integrate** external frameworks and libraries into applications.
- **Develop** a robust application that meets given requirements.

This course supports student outcomes by developing:

Outcomes	Selected
1. Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions	
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline	✓
3. Communicate effectively in a variety of professional contexts.	
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.	

Outcomes	Selected
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline	
6. Apply computer science theory and software development fundamentals to produce computing-based solutions.	✓

7. Mapping between CLOs and SOs

CLO	SO2	SO6
CLO1		✓
CLO2	✓	
CLO3	✓	
CLO4		✓
CLO5	✓	

8. List of Topics

The major topics covered in this course are:

- Introduction to Java
- Java data types
- Console input and output
- Control structures
- Methods
- Arrays
- Classes and objects
- Method overloading
- Inheritance
- Aggregation and composition
- Method overriding
- Abstract classes and methods
- Interfaces
- Exception handling
- File processing
- Generics
- Collections
- Multithreading
- Graphical user interfaces (GUI)
- Object serialization
- XML processing
- Database commands and connectivity
- Application programming interfaces (APIs)
- Remote method invocation (RMI)

1. Course Number and Name

06033172 - Programming Mathematics

2. Credits (Contact Hours/Week for Fall/Spring Semester)

3 (3 Lec , 0 Lab)

3. Course Coordinator

Assigned by the Computer Science Department each semester

4. Textbook and Supplemental Materials

Textbook: Stormy Attaway, 'MATLAB: A Practical Introduction to Programming and Problem Solving, ELSEVIER, Inc. Printed in Canada,2018.

Supplemental Materials:

- Boufadene Mourad, Modeling and Control of AC Machine Using MATLAB®/SIMULINK', CRC Press, 2018
- MATLAB, The Language for Technical Computing: Computation, Visualization, and Programming', Math Works Inc., 2007.

5. Course Information

Catalog Description: Introduction to MATLAB infrastructure. Working with linear algebra. Arrays and matrices. Graphics: plotting, images and GUI. Use of Symbolic Math toolbox. Flow control. Data structures. Scripts, functions and calculus. Manipulating images. Manipulating sounds, solving equations.

Prerequisites: 06052112 - Object-oriented Methods

Type of Course: Selective Requirements

6. Course Objectives and Outcomes (CLOs)

Students who successfully complete this course will be able to:

- **Explain** the mathematical foundations used in computational problem solving.
- **Apply** numerical and mathematical methods to programming problems.
- **Implement** mathematical algorithms and matrix operations in programs.
- **Analyze** the accuracy and efficiency of mathematical computations.
- **Develop** programs that solve mathematical problems correctly.

This course supports student outcomes by developing:

Outcomes	Selected
1. Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions	✓
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline	✓
3. Communicate effectively in a variety of professional contexts.	
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.	

Outcomes	Selected
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline	
6. Apply computer science theory and software development fundamentals to produce computing-based solutions.	✓

7. Mapping between CLOs and SOs

CLO	SO1	SO2	SO6
CLO1	✓		
CLO2	✓		
CLO3			✓
CLO4	✓		
CLO5		✓	

8. List of Topics

The major topics covered in this course are:

- Introduction to MATLAB
- MATLAB interface and windows
- MATLAB built-in functions
- Data types and type casting
- Mathematical functions
- Trigonometric functions
- Statistical and data-analysis functions
- Random-number generation
- MATLAB computational limits
- Special values and built-in functions
- Matrix creation and manipulation
- Extracting data from matrices
- Special matrices
- Solving linear equations
- Matrix inverse method
- Cramer's rule
- Creating and running M-file programs
- User input and formatted output
- Graphical input techniques
- MATLAB cell mode
- Functions with single and multiple inputs and outputs
- User-defined toolboxes
- Anonymous functions
- Function handles

- User-defined functions
- Conditional statements
- if, elseif, and else structures
- switch and case structures
- for loops
- while loops
- Creating and labeling two-dimensional plots
- Customizing plot appearance
- Creating subplots
- Creating three-dimensional plots
- Interactive plotting tools

1. Course Number and Name

06053273 - Expert Systems

2. Credits (Contact Hours/Week for Fall/Spring Semester)

3 (3 Lec , 0 Lab)

3. Course Coordinator

Assigned by the Computer Science Department each semester

4. Textbook and Supplemental Materials

Textbook: Expert Systems: Principles and Applications by Stuart Russell and Peter Norvig (Latest Edition).

Supplemental Materials:

- Expert Systems: A Practical Guide by Dan W. Patterson (Latest Edition)
- Expert Systems: Techniques and Applications by S.J. Green and L.C. Oxborrow (Latest Edition)
- Expert Systems: Concepts and Applications by John Durkin (Latest Edition).

5. Course Information

Catalog Description: This course provides an in-depth exploration of expert systems, focusing on the development, implementation, and application of intelligent systems capable of mimicking human expertise. Topics include knowledge representation, rule-based systems, inference engines, uncertainty handling, and real-world applications.

Prerequisites: 06013176 - Artificial Intelligence

Type of Course: Selective Requirements

6. Course Objectives and Outcomes (CLOs)

Students who successfully complete this course will be able to:

- **Explain** the architecture and components of expert systems.
- **Represent** domain knowledge using appropriate formalisms.
- **Apply** inference and reasoning techniques in rule-based systems.
- **Analyze** knowledge acquisition and system validation issues.
- **Develop** a simple expert system for a given problem domain.

This course supports student outcomes by developing:

Outcomes	Selected
1. Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions	✓
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline	✓
3. Communicate effectively in a variety of professional contexts.	
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.	

Outcomes	Selected
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline	
6. Apply computer science theory and software development fundamentals to produce computing-based solutions.	✓

7. Mapping between CLOs and SOs

CLO	SO1	SO2	SO6
CLO1			✓
CLO2	✓		
CLO3			✓
CLO4	✓		
CLO5		✓	

8. List of Topics

The major topics covered in this course are:

- Introduction to expert systems
- Types of expert systems
- Applications and societal impacts of expert systems
- Knowledge representation
- Semantic networks
- Frames and scripts
- Rule-based knowledge representation
- Rule-based systems
- Structure and syntax of rules
- Forward chaining
- Backward chaining
- Inference engines
- Uncertainty in expert systems
- Sources of uncertainty in knowledge
- Fuzzy logic and its applications
- Bayesian networks
- Probabilistic reasoning
- Knowledge acquisition and elicitation
- Strategies for acquiring expert knowledge
- Knowledge-elicitation techniques
- Tools and methodologies for knowledge acquisition
- Expert-system development tools
- Comparison and evaluation of development tools
- Expert-system development case studies
- Ethical considerations in expert-system development

- Integration of machine learning and expert systems
- Hybrid intelligent systems
- Emerging trends in expert-systems research
- Expert-system project development and presentation

1. Course Number and Name

06053259 - Image Processing

2. Credits (Contact Hours/Week for Fall/Spring Semester)

3 (3 Lec , 0 Lab)

3. Course Coordinator

Prof. Mohamed Hashem Saeed Almourish

4. Textbook and Supplemental Materials

Textbook:

- DEY, S. (2021). Image Processing Masterclass with Python: 50+ Solutions and Techniques Solving Complex Digital Image Processing Challenges Using Numpy, Scipy, Pytorch and Keras (English Edition). BPB PUBN.

Supplemental Materials:

- Anil K. Jain (1989). Fundamentals of Digital Image Processing. Prentice-Hall International.
- Zahid Hussain (1991). Digital Image Processing. Ellis Horwood.
- Jan Teuber (1993). Digital Image Processing. Prentice Hall.
- John C. Russ (2006). The Image Processing Handbook. CRC Press.
- Rafael C. Gonzalez, Richard E. Woods and Steven L. Eddins (2009). Digital Image Processing Using MATLAB. Gates mark Publishing.
- S. Dey (2021). Image Processing Masterclass with Python: 50+ Solutions and Techniques Solving Complex Digital Image Processing Challenges Using Numpy, Scipy, Pytorch and Keras. BPB Publications.

5. Course Information

Catalog Description: This is an introductory course to the fundamentals of digital image processing. It emphasizes general principles of image processing, rather than specific applications. The student will cover topics such as image acquisition and display, properties of the human visual system, color representations, sampling and quantization, point operations, linear image filtering and correlation, transforms and sub-band decompositions, and nonlinear filtering, image compression using various methods.

Prerequisites: 06052157 - Computer Graphics

Type of Course: Selective Requirements

6. Course Objectives and Outcomes (CLOs)

Students who successfully complete this course will be able to:

- **Explain** the fundamentals of digital image representation and processing.
- **Apply** spatial and frequency-domain image enhancement techniques.
- **Implement** filtering, segmentation, and feature-extraction algorithms.
- **Analyze** the effect of image processing operations on image quality.
- **Develop** an image processing application for a given task.

This course supports student outcomes by developing:

Outcomes	Selected
1. Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions	✓
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline	✓
3. Communicate effectively in a variety of professional contexts.	
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.	
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline	
6. Apply computer science theory and software development fundamentals to produce computing-based solutions.	✓

7. Mapping between CLOs and SOs

CLO	SO1	SO2	SO6
CLO1			✓
CLO2	✓		
CLO3		✓	
CLO4	✓		
CLO5		✓	

8. List of Topics

The major topics covered in this course are:

Chapter 1: Introduction to Digital Image Processing

- Introduction to digital image processing
- Human visual system
- Image acquisition and display
- Image Storage

Chapter 2: Image Formats and Color Representations

- RGB, Gray and Binary Images formats
- Colour representations

Chapter 3: Image Sampling and Quantization

- Sampling and quantization (analog to digital conversion)
- Image Enhancement fundamentals

Chapter 4: Intensity Transformation and Histogram Processing

- Point Operations
- Image Enhancement, Intensity Transformation
- Histogram stretching and shrinking
- Histogram equalization

Chapter 5: Arithmetic Operations

- Addition, Subtraction, Division, Multiplication

Chapter 6: Image Filtering and Edge Detection

- Linear image filtering and correlation
- Edge Detection Filters (Canny, Prewitt, Laplacian, Sobel)
- Nonlinear filtering

Chapter 7: Image Restoration

- Image restoration overview
- System model
- Noise
- Mean Filters
- Order-Statistic Filters

Chapter 8: Image Compression

- Image compression using various methods

Chapter 9: MATLAB and Python Fundamentals

- Introduction to MATLAB, Reading Images, Displaying Images
- Python Fundamentals

1. Course Number and Name

06083141 - Wireless Networks

2. Credits (Contact Hours/Week for Fall/Spring Semester)

3 (3 Lec , 0 Lab)

3. Course Coordinator

Assigned by the Computer Science Department each semester

4. Textbook and Supplemental Materials

Textbook:

- Sparks, T. ," Wireless Communication Networks and Systems", 5th Edition, Published 2022 by STATES ACADEMIC Press, ISBN 9781639895700.

Supplemental Materials:

- Wireless and Mobile Network Architectures, Yi-Bing Lin, Imrich Chlamtac, John Wiley & Sons, 2000.
- Data Communications and Networking: Behrouz A. Forouzan, McGrawHill, Third Edition.

5. Course Information

Catalog Description: Introduction to mobile and wireless networks. Designing computer networks to support computer mobility. Mobile network architecture. Wireless technologies and protocols. Wireless LAN standards. Models for indoor and outdoor mobile networks. Systems issues such as performance. Quality of service guarantees, reliability, and security in mobile computing environment. Hardware and access protocols for mobile networks. Mobile application protocols.

Prerequisites: 06052140- Computer Networks

Type of Course: Selective Requirements

6. Course Objectives and Outcomes (CLOs)

Students who successfully complete this course will be able to:

- **Explain** the principles and architecture of wireless networks.
- **Analyze** wireless transmission and medium access techniques.
- **Apply** cellular and wireless LAN concepts to network scenarios.
- **Describe** mobility management and wireless network protocols.
- **Evaluate** wireless network performance and security considerations.

This course supports student outcomes by developing:

Outcomes	Selected
1. Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions	✓
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline	✓
3. Communicate effectively in a variety of professional contexts.	

Outcomes	Selected
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.	
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline	
6. Apply computer science theory and software development fundamentals to produce computing-based solutions.	✓

7. Mapping between CLOs and SOs

CLO	SO1	SO2	SO6
CLO1			✓
CLO2	✓		
CLO3		✓	
CLO4			✓
CLO5	✓		

8. List of Topics

The major topics covered in this course are:

- Introduction to wireless local area networks (WLANs)
- Antennas and radio-wave propagation
- Signal-encoding techniques
- Spread-spectrum techniques
- Cellular wireless networks
- Mobile IP
- Wireless access protocols
- Wireless LAN technologies
- Wireless networking laboratory activities
- Wi-Fi technologies
- IEEE 802.11 wireless LAN standard
- IEEE 802.11 physical layer
- Other IEEE 802.11 standards
- IEEE 802.11a standard
- IEEE 802.11b standard
- Bluetooth technology
- IEEE 802.15 wireless personal area network standard

1. Course Number and Name

06034259 - Software Projects Management

2. Credits (Contact Hours/Week for Fall/Spring Semester)

4 (2 Lec , 2 Lab)

3. Course Coordinator

Dr. Ahmed Khairat

4. Textbook and Supplemental Materials

Textbook: Kathy Schwalbe, Information Technology Project Management, Cengage Learning. (9th Ed), 2018.

Supplemental Materials:

- Information Technology Project Management (8th Ed.), 2016, by Schwalbe, K., Course Technology.

Online / Open-Access Resources:

- Global Project Management: Communication, Collaboration and Management', 2007, by JeanCarlo Binder. Gower Publishing

5. Course Information

Catalog Description: This course covers Management activities, Project Planning and scheduling. Scope Management, Managing People, Time Management, Software Cost Estimation, Quality Management, Software Measurement and Metrics, Process Improvement, Configuration Management.

Prerequisites: 06032250 - Software Engineering Basics

Type of Course: Selective Requirements

6. Course Objectives and Outcomes (CLOs)

Students who successfully complete this course will be able to:

- **Explain** the processes and knowledge areas of software project management.
- **Apply** estimation, planning, and scheduling techniques to a project.
- **Manage** risk, quality, and change throughout the project life cycle.
- **Function** effectively as a member or leader of a project team.
- **Communicate** project status and decisions to stakeholders.

This course supports student outcomes by developing:

Outcomes	Selected
1. Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions	
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline	✓
3. Communicate effectively in a variety of professional contexts.	✓
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.	✓

Outcomes	Selected
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline	✓
6. Apply computer science theory and software development fundamentals to produce computing-based solutions.	

7. Mapping between CLOs and SOs

CLO	SO2	SO3	SO4	SO5
CLO1	✓			
CLO2	✓			
CLO3			✓	
CLO4				✓
CLO5		✓		

8. List of Topics

The major topics covered in this course are:

- **Chapter 1 – Introduction to Project Management**
  - What is a project? (Project attributes and constraints)
  - What is project management? (Stakeholders, knowledge areas, tools, and techniques)
  - Program and project portfolio management
  - The role of the project manager (Job description, skills, and the PMI Talent Triangle)
  - The project management profession (History, PMI, certification, ethics)
  - Project management software and careers
- **Chapter 2 – The Project Management and Information Technology Context**
  - A systems view of project management and the three-sphere model
  - Understanding organizations (The four frames, organizational structures, and culture)
  - Focusing on stakeholder needs and top management commitment
  - Project and product life cycles
  - The nature of IT projects and characteristics of team members
  - Recent trends: Globalization, outsourcing, virtual teams, and Agile
- **Chapter 3 – The Project Management Process Groups**
  - Mapping the five process groups (Initiating, Planning, Executing, Monitoring & Controlling, Closing) to knowledge areas
  - Developing an IT project management methodology
  - Case study: JWD Consulting project (Predictive approach from pre-initiation to closing)
  - Case study: JWD Consulting project (Agile approach with Scrum roles, artifacts, and ceremonies)
  - Templates by process group
- **Chapter 4 – Project Integration Management**
  - Strategic planning and project selection (Methods including NPV, ROI, payback analysis, weighted scoring, and balanced scorecard)

- Developing a project charter and project management plan
- Directing, managing, and monitoring project work
- Managing project knowledge
- Performing integrated change control and change control systems
- Closing projects or phases

- **Chapter 5 – Project Scope Management**
  - Planning scope management and collecting requirements
  - Defining scope and creating the Work Breakdown Structure (WBS)
  - Approaches to developing WBS (Guidelines, analogy, top-down/bottom-up, mind mapping)
  - Validating and controlling scope
  - Suggestions for improving user input and reducing changing requirements
- **Chapter 6 – Project Schedule Management**
  - The importance of project schedules
  - Defining, sequencing activities, and estimating activity durations
  - Developing the schedule using Gantt charts and Critical Path Method (CPM)
  - Critical chain scheduling and PERT
  - Controlling the schedule and reality checks
- **Chapter 7 – Project Cost Management**
  - Basic principles of cost management
  - Planning cost management and estimating costs (Types of estimates, tools, typical problems)
  - Determining the budget and controlling costs
  - Earned Value Management (EVM)
  - Project portfolio management
- **Chapter 8 – Project Quality Management**
  - Planning, managing, and controlling quality
  - Tools and techniques for quality control (Statistical sampling, Six Sigma, testing)
  - Modern quality management (Deming, Juran, Crosby, Ishikawa, Taguchi, Feigenbaum)
  - Improving IT project quality (Leadership, cost of quality, maturity models like CMMI)
- **Chapter 9 – Project Resource Management**
  - Planning, acquiring, and developing the project team
  - Managing the project team and understanding team dynamics
  - Theories on motivation and team performance
  - Control and conflict resolution in teams
- **Chapter 10 – Project Communications Management**
  - The importance of communication in projects
  - Planning communications management
  - Managing and monitoring communications
  - Techniques for effective communication and technology choices
- **Chapter 11 – Project Risk Management**
  - Risk management planning and risk identification
  - Performing qualitative and quantitative risk analysis
  - Planning risk responses and implementing them
  - Monitoring risks throughout the project
- **Chapter 12 – Project Procurement Management**
  - Planning procurement and conducting procurements
  - Contract types and their uses
  - Controlling procurement and closing procurements
  - Procurement statements of work

- **Chapter 13 – Project Stakeholder Management**
  - Identifying project stakeholders
  - Planning and managing stakeholder engagement
  - Monitoring stakeholder engagement
  - Power/interest grids and their use in stakeholder management

1. Course Number and Name

06014171 - Selected Topics (1) "**C++ programming**"

2. Credits (Contact Hours/Week for Fall/Spring Semester)

3 (3 Lec , 0 Lab) – Fall-2025

3. Course Coordinator

Assigned by the Computer Science Department each semester (**Dr. Ahmed Khairat**)

4. Textbook and Supplemental Materials

Textbook: C++ Programming: From Problem Analysis to Program, D S Malik, CEGAGE Learning, 2011.

Supplemental Materials:

- C How to Program: (8th Edition) by Harvey Deitel, Paul J. Deitel, Paperback, 1008 Pages, Published 2015.

Online / Open-Access Resources:

- https://www.w3schools.com/

5. Course Information

Catalog Description: Introduction to problem-solving and programming using C++. Topics include fundamental programming concepts, input/output, control structures, functions, arrays, strings, structures, object-oriented programming, inheritance, polymorphism, pointers, and dynamic memory allocation.

Prerequisites: Departmental approval

Type of Course: Selective Requirements

6. Course Objectives and Outcomes (CLOs)

Students who successfully complete this course will be able to:

- **Explain and apply** fundamental C++ concepts, including program structure, data types, variables, constants, operators, expressions, type conversion, namespaces, and standard input/output.
- **Analyze** computing problems and **develop** structured algorithms using selection statements, repetition structures, and user-defined functions.
- **Design, implement, test, and debug** C++ programs using functions, arrays, strings, structures, and user-defined data types.
- **Develop** object-oriented C++ solutions using classes, encapsulation, constructors, destructors, inheritance, composition, and polymorphism.
- **Implement and evaluate** advanced C++ solutions using pointers, dynamic memory allocation, virtual functions, and abstract classes.

This course supports student outcomes by developing:

Outcomes	Selected
1. Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions	✓
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline	✓
3. Communicate effectively in a variety of professional contexts.	
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.	
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline	
6. Apply computer science theory and software development fundamentals to produce computing-based solutions.	✓

Note: Student Outcomes (SOs) are assigned per topic each semester; the mapping shown is representative.

7. Mapping between CLOs and SOs

CLO	SO1	SO2	SO6
CLO1	✓		✓
CLO2	✓	✓	✓
CLO3	✓	✓	✓
CLO4		✓	✓
CLO5	✓	✓	✓

8. List of Topics

The major topics covered in this course are:

- **Chapter 1 – An Overview of Computers and Programming Languages**
  - A brief history of computers and programming languages
  - Understanding the problem analysis and program design process
  - The structure of a C++ program and its execution process
  - Introduction to object-oriented programming (OOP)
- **Chapter 2 – Basic Elements of C++**
  - Basic syntax, data types, and variables
  - Constants and the const modifier
  - Arithmetic operators and expressions
  - Type conversion and the string type

- **Chapter 3 – Input/Output**
  - Standard input/output using cin and cout
  - Formatting output with manipulators
  - Using the iomanip library
- **Chapter 4 – Control Structures I (Selection)**
  - Relational and logical operators
  - if and if...else selection statements
  - switch structures
- **Chapter 5 – Control Structures II (Repetition)**
  - while, for, and do...while loops
  - Nested control structures and loop techniques
  - break and continue statements
- **Chapter 6 – User-Defined Functions I**
  - Function definition, prototype, and calls
  - Parameters, return types, and function overloading
  - Scope of identifiers (global, local)
- **Chapter 7 – Namespaces, the Class String, and User-Defined Simple Data Types**
  - Creating and using namespaces
  - User-defined simple data types (typedef, enum)
- **Chapter 8 – Arrays and Strings**
  - Declaring, initializing, and using arrays
  - Parallel arrays and two-dimensional arrays
  - C-strings and the string class
- **Chapter 9 – Records (Structs)**
  - Defining and using structures (struct)
  - Arrays of structures and nested structures
- **Chapter 10 – Classes and Data Abstraction**
  - Introduction to classes and object-oriented design
  - Data abstraction, information hiding, and encapsulation
  - Constructors and the destructor
- **Chapter 11 – Inheritance and Composition**
  - Inheritance: derived classes and access specifiers
  - Inheritance hierarchy and polymorphism
  - Composition and its relationship to objects
- **Chapter 12 – Pointers, Classes, Virtual Functions, and Abstract Classes**
  - Pointer concepts and dynamic memory allocation (new, delete)
  - Classes and dynamic memory management
  - Virtual functions, abstract classes, and polymorphism

1. Course Number and Name

06054272 - Selected Topics (2)

2. Credits (Contact Hours/Week for Fall/Spring Semester)

3 (3 Lec , 0 Lab)

3. Course Coordinator

Assigned by the Computer Science Department each semester

4. Textbook and Supplemental Materials

Textbook: Howard E. Poston III "Python for Cybersecurity", Wiley, 2022.

Supplemental Materials:

- Recent research papers, technical articles, and online resources relevant to the topic.
- Lecture notes and materials provided by the instructor.

5. Course Information

Catalog Description: This course introduces the concepts and practices of cybersecurity, with a focus on the Python programming language. Students will learn the basics of Python programming, how it's applied in cryptography and network scanning, password cracking techniques, malware design, and web-based APIs and RESTful transactions. The course blends theoretical knowledge with practical application to provide a comprehensive understanding of the subject matter.

Prerequisites: Departmental approval

Type of Course: Selective Requirements

6. Course Objectives and Outcomes (CLOs)

Students who successfully complete this course will be able to:

- **Explain** the advanced concepts and principles of the selected topic.
- **Apply** advanced methods and techniques relevant to the topic.
- **Analyze** complex problems within the scope of the selected topic.
- **Develop** advanced solutions or projects related to the topic.
- **Communicate** results and findings related to the selected topic.

This course supports student outcomes by developing:

Outcomes	Selected
1. Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions	✓
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline	✓
3. Communicate effectively in a variety of professional contexts.	
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.	

Outcomes	Selected
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline	
6. Apply computer science theory and software development fundamentals to produce computing-based solutions.	✓

Note: Student Outcomes (SOs) are assigned per topic each semester; the mapping shown is representative.

7. Mapping between CLOs and SOs

CLO	SO1	SO2	SO6
CLO1	✓		
CLO2			✓
CLO3	✓		
CLO4		✓	
CLO5		✓	

8. List of Topics

The major topics covered in this course are:

- Introduction to Python
- Python strings, lists, tuples, dictionaries, and sets
- Boolean data type and expressions
- Conditional statements
- for loops
- while loops
- Python operators
- String formatting
- String methods
- User-defined functions
- Data visualization
- Importing modules and libraries
- Python libraries for cybersecurity
- Caesar cipher and the ASCII table
- ROT13 implementation in Python
- Using Python encryption libraries
- Password-security assessment
- Dictionary-attack concepts and prevention
- Brute-force attack concepts and mitigation
- Network scanning for authorized security assessment
- Using APIs for cybersecurity applications
- Malware concepts, analysis, and defensive design
- Cybersecurity programming project