Course Description

Introduction to the Undergraduate Information Technology Program

by

Author

Mohammad Hossein **Mohammadi**

Mohammadimh76.github.io

M.H.Mohammadimir2017@gmail.com

Github Repository

University: Islamic Azad University Najafabad Branch
Faculty: Computer Engineering
Field of Study: Information Technology (IT)

Licence

Curriculum and course descriptions for each lesson are provided in English, translated according to the Ministry of Science program.

Hello! Certain universities like TUM and ETH require you to submit syllabi for the courses you've completed during your undergraduate or graduate studies when applying. This document serves as a translation of the Ministry of Science file.
Please consult your respective file for each major. If you don't find a particular course in this list, kindly submit a request on **Github** to have it added. This ensures that your classmates' and friends' time isn't wasted unnecessarily.
Originally prepared for the Computer Engineering major, Information Technology branch, at Islamic Azad University, Najafabad Branch (**IAUN**), this document was inspired by my friend Hossein's GitHub repository. Some courses are sourced from Hossein's repository. If you're in the Computer Engineering major with a software specialization, you can refer to **Hossein's repository**.

Description

Introduction to the Undergraduate Information Technology Program at Islamic Azad University, Najaf Abad Branch

Welcome to the comprehensive guide for the Undergraduate Information Technology Program at Islamic Azad University, Najaf Abad Branch. This document is intended to be a detailed resource for students, faculty, and academic advisors, offering an in-depth exploration of the curriculum, courses, and educational goals of the Information Technology undergraduate program.

Structured to provide a strong foundation in both theoretical concepts and practical applications, the curriculum is divided into four main categories: Specialized Courses, Main Courses, Basic Courses, and General Courses. This approach ensures a well-rounded education, blending advanced technical knowledge with essential foundational principles.

- **Specialized Courses:** Dive deeply into cutting-edge topics and technologies within the Information Technology field, allowing students to specialize in areas of their interest.
- **Main Courses:** Cover essential principles and practices fundamental to Information Technology, providing the groundwork for both specialized study and broad understanding.
- **Basic Courses:** Focus on reinforcing students' understanding of fundamental scientific and mathematical concepts, crucial for success in technology-related fields.
- **General Courses:** Enhance students' broader skills such as critical thinking, ethics, and communication, essential for professional success.

Each course category is accompanied by detailed descriptions, including course objectives, covered topics, and recommended references, offering a clear roadmap for students. This meticulous documentation ensures that every student has the necessary information to navigate their academic and professional journey successfully.

This guide encapsulates the essence of the Information Technology undergraduate program at Islamic Azad University, Najaf Abad Branch, reflecting our commitment to providing a comprehensive curriculum that adapts to the evolving needs of the technology sector.

Contents

1

Specialized Courses

1.1. System Analysis and Design

The purpose of this course is to familiarize students with the concepts of analyzing and designing software systems. Throughout the course, students will explore various topics including different information systems, the software product life cycle, development methodologies, analysis, structured design, and project management concepts.

Name of Course	System Analysis and Design
Number of Credits	3
Number of Hours	48
Cross section	Bachelor's Degree
Needs	None
Prerequisites	Advanced Programming
Reference books	[1] L. D. Bentley and J. L. Whitten, Systems Analysis and Design for the Global Enterprise. 7th Edition, McGraw-Hill, 2007. [2] C. Larman, Applying UML and Patterns: An Introduction to Ohject-Oriented Analysis and Design and Iterative Development. Addison Wesley, 2004 [3] Roger S. Pressman, Software Engineering: a Practitioner's Approuch. McGraw Hill Inc.,7th Edition, 2011.

At the end of the module, students will be able to:

- Apply software development principles in software production.
- Select and use software development tools effectively.
- Document software products using the UML language.
- Analyze and design systems using the object-oriented method.
- Understand the importance of software testing and how to design tests.

Course Objectives:

1. Introduce software engineering and its challenges.
2. Explore various software development process models and their differences.

3. Learn software analysis and design methods.
4. Understand requirements engineering and system analysis.
5. Explore system design and software architecture.
6. Learn to build software effectively.
7. Get an introduction to software testing.
8. Gain introductory insights into project management and planning.

1.2. Principles of Database Design

Students will be able to apply essential concepts of relational database systems and systematically evaluate them. They'll develop expertise in using a database system, beginning from conceptual design to implementation and physical design. They'll proficiently formulate complex queries in SQL and grasp basic logical and physical optimization based on relational algebra. Additionally, they'll understand how to ensure the security of a database application, covering recovery, concurrency control, and authorization. The course also includes database design methodology, focusing on key topics such as data models, database system architecture, data definition and manipulation languages, database design methods, and the theory of data dependencies and relational normalization.

Name of Course	Principles of Database Design
Number of Credits	3
Number of Hours	48
Cross section	Bachelor's Degree
Needs	None
Prerequisites	Data Structures
Reference books	[1] R. Ramakrishnan and J. Gehrke, Database Management Systems, 3rd Edition. McGraw-Hill Inc., 2003. [2] H. Garcia-Molina, et al., Database Systems: The Complete Book 2nd Edition, Pearson Prentice Hall, 2009. [3] J.D. Ullman and J. Widom, A First Course in Database Systems. 3rd Edition, Pearson Prentice Hall Inc., 2008.

At the end of the module, students will be able to:

- Gain a general understanding of database systems and their architecture.
- Learn to create diagrams depicting existence and relationships.
- Convert diagrams of existence and relationships into corresponding relationships.
- Recognize functional dependencies and normalize relationships accordingly.
- Practice SQL through questions and answers.
- Understand how relationships and indexes are stored in a database system.

Course Objectives:

1. Basic concepts
    - Understand the relational model.
    - Learn relational algebra and arithmetic.
    - Master SQL.
    - Comprehend normalization principles.
2. Storage and indexing

- Explore data storage methods.
- Study indexing using tree structures.
- Understand indexing based on scrambling techniques.

3. Evaluate questions and answers

- Implement external sorting.
- Evaluate relational operators.
- Optimize queries effectively.

4. Transaction management

1.3. Fundamentals of Information Technology

In the first lesson, IT students will be introduced to the fundamental principles, definitions, concepts, applications, organizational and social effects, management concepts, foundations, and architecture of information technology. The comprehensive scope of Rabaneh's applications serves as the framework for discussions on information technology, with considerations of its societal, economic, and cultural impacts. As creators and advocates of new solutions in this field, computer and information technology engineers should stay informed about the latest concepts, achievements, and applications of this technology globally and in Iran. While this lesson provides a broad overview of these concepts, subsequent lessons will delve deeper into each topic, forming a structured progression.

Name of Course	Fundamentals of Information Technology
Number of Credits	3
Number of Hours	48
Cross section	Bachelor's Degree
Needs	None
Prerequisites	None
Reference books	[1] Linda Volonino, Efrain Turban, Information Technology for Management Improving Performance in the digital Economy. & Edition, Wiley, 2011. [2] Efraim Turban, Dorothy Leidner, Ephraim Mclean and James Wetherbe; information Technology. for Management Transforming Organizations in the Digital Economy, 5 Edition, John Wiley & Sons Inc, 2006 [3] E.Turban, R.K.Rainer, R.E.Potter, Introduction to Information Technology, Wiley, 3rd Edition, 2005. [4] Urs Birchler and Monika Butler, Information Economics, Routledge, 2007. [5] E.W.Martin, C.V.Brown, Managing Information Technology. Prentice Hall, 5th Edilion, 2004. [6] K.D.Willett, Information Assurance Architecture. CRC, 2008. [7] Thomas.H.Davenport and Laurence Prusak, Information Ecology: Mastering the Information and knowledge Environment. Oxford University Press, 1997.

Course Objectives:

1. Introduction
2. Understand the background, definition, principles, framework, and preconceptions of information technology.
3. Differentiate between data, information, and knowledge.
4. Explore Network Computing and its role in Fa Management within a Digital Economy-based organization.
5. Develop the ability to absorb Fa, assess electronic readiness, understand digital rankings, and identify digital gap criteria.
6. Study E-commerce, business intelligence, and data warehousing.
7. Explore Wireless and mobile computing, understanding pervasive, live, and current technologies.

8. Examine various business systems, including corporate, local, and international systems, their features, and integration.
9. Learn about support systems for management, supply chains, organizational resource planning, and client linkage.
10. Understand different types of Internet structures, their foundations, and Fa architecture.
11. Explore combined applications of today's value-added Fa.
12. Analyze the impacts, etiquette, and security considerations of information technology.
13. Discuss the Information society, e-government, e-services, and their foundations.
14. Explore National and International Information Technology landscapes.

1.4. Principles of Information Technology Strategic Management and Planning

The main objectives of this course are to familiarize students with theoretical and practical aspects of Fa-strategic studies in information technology management and planning within organizations. Students will gain knowledge of selecting the appropriate type of Wi-Fi strategic study for each organization based on its absorption capacity. They will learn to use suitable methods and produce solutions, transitioning with organizational methods using adaptable engineering patterns.
The second goal of the course is to understand the necessity of drawing architectural plans and updating them to progress from the current situation to desired conditions. Students will also learn to integrate solution systems that align with national projects such as e-government. Additionally, the course aims to improve students' attitudes and equip them with the ability to derive systemic solutions for organizational challenges through practical exercises.

Name of Course	Principles of Information Technology Strategic Management and Planning
Number of Credits	3
Number of Hours	48
Cross section	Bachelor's Degree
Needs	None
Prerequisites	None
Reference books	[1] Inge Hanschke, Strategic IT Management, Springer, 2010. [2] Danny Greefhorst, Erik Proper, Architecture Principles, Springer, 2011. [3]Martin Op't Land, Enterprise Architecture Creating Value by informed Governance, Springer, 2009. [4] Mario Godincz, The Art of Enterprise Information Architecture, IBM Press, 2010.

Course Objectives:

1. Introduction
    - Understand the fundamentals and principles of strategic management and planning.
    - Define the foundation, architecture, management, and planning of the Fa.
    - Grasp concepts such as organization, enterprise, mission, vision, plan, and project.
2. Introduction to the historical background of early organizational orthodox Romans
    - Explore Information Engineering.
    - Study Strategic Business Planning.
    - Examine Information Resource Management.
3. Basic Patterns of Strategic Planning and Management

- Learn about the Performance Measurement Reference Model.
- Apply models such as PEST, SWOT, SPACE QSPM.
- Understand the Zakman Strategic Framework and its applications.
- Explore Strategic Alignment Models.
- Study Mans's Integrated Architecture and Management Model and Nolan Organizational Life and Maturity Model.

4. Detailed Study of Family-Oriented Strategic Planning Patterns

- Explore Strategic Business Planning.
- Understand Strategic Planning of Information Systems.
- Examine Planning and Managing the Organization's Information Resources.
- Study Strategic Planning of the Organization's Information Technology.
- Learn about Organizational Architecture Planning.

5. Organizational FI Strategic Planning Bicycle

- Gain knowledge of the organization.
- Conduct alignment analysis.
- Analyze the current situation.
- Design optimal conditions.
- Develop a transition program.

6. Organizational Information Architecture Planning Process Model

- Develop policies.
- Create maps of existing conditions.
- Design optimal status maps.
- Measure the distance between the current and favorable situations.
- Compile a change statement and schedule a transition.
- Develop operational plans.

7. Example of Organizational Architecture Methods and Reference Models

- Explore FEAF and NIST.

8. Organizational Architecture in Iran and Introduction of Several Large Projects

- Examine Iranian Organizational Architecture Reference Models.
- Understand Iran's Position in the Rankings of Organizational Architecture Worldwide.
- Explore Several Large National Projects.

9. Status and Future of Applications of Fa's Strategic Studies

- Understand the role of Fa and the required integration of the e-government project.
- Explore Management of Organizational Architecture and Service-Oriented Organizational Architecture.
- Examine Applications of Some Standard Fa Service Management Templates.
- Study Quality Assurance of Information and Services and Organizational Knowledge Architecture.
- Understand the Koenigsberg Model.
- Gain a Landscape View of Architectural Studies.

- Explore Minimalist View of Architecture.
- Understand A Supermodel Look at Enterprise Architecture.

10. Documenting and Updating Strategic Study Plans and Computer Tools

- Learn about types and methods of drawing maps for each layer of architecture.
- Explore various tools for producing, maintaining, and updating maps.
- Understand how to select the minimum number of maps per layer.
- Study Fa-Strategic Studies Document Configuration Management.

1.5. Information Technology Projects Management

In this course, students become familiar with the principles and foundations of project management and control, and they learn how to apply techniques from this field in information technology projects.

Name of Course	Information Technology Projects Management
Number of Credits	3
Number of Hours	48
Cross section	Bachelor's Degree
Needs	None
Prerequisites	None
Reference books	[1] David L. Olson, Introduction to Information Systems Project Management with CD ROM Mandatory Package. McGraw-Hill, 2001. [2] Graham McLeod and Derek Smith, Managing Information Technology Project, Course Technology, 1996. [3] Chris Kemerer, Software Project Management: Readings and Cases, McGraw-Hill, 1997.

Course Objectives:

1. Challenges of Fa Management
2. Strategies for Fa Projects
3. Project Initiation and Requirement Definition
4. Formation of the Fa Project Team
5. Project Planning
6. Estimation Techniques
7. Project Execution and Control
8. Management of Hardware and Communication Projects
9. Software Project Management
10. Integrated Systems Management

1.6. Enterprise Applications Integration

Students will gain familiarity with techniques, tools, and skills for integrating enterprise application systems, providing solutions for linking legacy and new applications in network application environments. They will explore concepts such as data and process integration, the enterprise service gateway, its structure, and application, as well as various integration patterns in service architectures. Additionally, students will learn about incident-driven and grid integration, along with safety considerations in integration processes.

Name of Course	Enterprise Applications Integration
Number of Credits	3
Number of Hours	48
Cross section	Bachelor's Degree
Needs	None
Prerequisites	System Analysis and Design - Computer Networks
Reference books	[1] G.Schmutz, D.Liebhart,P.Welkenbach , Service-Oriented Architecture: An Integration Blueprint. Birmingham: Packt Publishing Ltd., 2010. [2] G. Hohpe and B. Woolf, Enterprise Integration Patterns. Addison-Wesley Inc., 2004. [3] David S. Linthicum, Enterprise Application Integration. Addison-Wesley Inc., 1999. [4] Amjad Umar, Enterprise and Inter-Enterprise Application Integration. NGE Solution, Inc., 2003.

Course Objectives:

1. Understand the basic concepts and necessities of integration, technologies, and architectures.
2. Identify the types, meanings, and levels of integration.
3. Explore types of layered architectures, their applications, advantages, and facilities.
4. Learn about different types of link middleware and their newer variants.
5. Study various integrated architectures.
6. Understand integration and Enterprise Service Bus (ESB).
7. Explore patterns of data integration, including service-oriented, event-driven, and grid integration.
8. Familiarize with basic technologies such as OSGi, JCA, JBI, SCA, and SDO.
9. Learn process modeling for integration.
10. Understand architectural integrity in integration.
11. Explore implementation scenarios of service, data, event, and network integration.
12. Study gateways and environment connection and integration.
13. Analyze styles and methods of integration.

14. Examine organizational architecture, productivity, and systems integrity.
15. Consider legal and security considerations in connection and integration.

1.7. Fundamentals of Secure Computing

During this course, students will first understand the necessity of establishing security and the various threats and attacks associated with it. They will delve into issues related to establishing and implementing security, including authentication methods, encryption, access control, and providing physical and environmental security. Additionally, students will familiarize themselves with security assessment standards and methodologies.

Name of Course	Fundamentals of Secure Computing
Number of Credits	3
Number of Hours	48
Cross section	Bachelor's Degree
Needs	Computer Networks
Prerequisites	Operating Systems
Reference books	[1] Ross J. Anderson, Security Engineering: A Guide to Building Dependable Distributed Systems. 2nd Edition, Wiley, 2008. [2] Matt Bishop, Introduction to Computer Security. Addison-Wesley, 2004.

Course Objectives:

1. Basic Concepts and Definitions
    - Understand basic concepts in security.
    - Identify types of threats and attacks.
    - Recognize vulnerabilities.
    - Explore security mechanisms.
    - Learn principles of Security Engineering.
2. Authentication
    - Understand authentication methods.
    - Learn about Identity Management Systems.
    - Discuss audit challenges vs. privacy.
3. Security Policy and Access Control Models
    - Study Discretionary Access Control (DAS).
    - Explore ACL Capability-based access control.
    - Understand hidden channels, information flow control, and mandatory access control (MAC).
    - Learn about Role-Based Access Control (RBAC).
4. Cryptography and Its Applications
    - Explore symmetric cryptography.
    - Understand asymmetric encryption and digital signatures.
    - Learn about hash functions.

- Study applications of cryptography in maintaining confidentiality, authentication, and non-repudiation.

5. Symmetric Cryptography
   - Understand the structure of the Feistel cipher.
   - Explore encryption algorithms based on the Feistel structure.
   - Learn about symmetric cryptographic standards.
   - Understand working modes in symmetric encryption.
6. Asymmetric Cryptography
   - Introduction and necessity of asymmetric cryptography.
   - Study cryptographic and asymmetric standards.
7. Digital Signature and Public Key Infrastructure
   - Understand simple digital signature algorithms.
   - Learn about RSA encryption.
   - Explore certificate management in Public Key Infrastructure (PKI).
8. Data Security Approaches
   - Explore transfer data security approaches.
   - Study processing data security approaches.
   - Understand stored data security approaches.
9. Physical and Environmental Security
   - Identify threats and physical attacks.
   - Explore physical and electronic barriers and locks.
   - Understand physical protection of communications.
   - Learn about sensors and alarm systems.
10. Security and Assurance Assessment
    - Study security assessment and testing methodologies.
    - Explore assessment and assurance standards.
11. Non-Technical Dimensions of Secure Computing
    - Discuss ethical dimensions in secure computing.
    - Explore legal dimensions of secure computing.
    - Understand social dimensions of secure computing.
12. Content Responsibility in Secure Computing

1.8. Engineering Economy

Engineering students will become familiar with the techniques of economic evaluation of industrial and service plans and projects in this course. They will learn how to apply these techniques and methods in their field of specialization.

Name of Course	Engineering Economy
Number of Credits	3
Number of Hours	48
Cross section	Bachelor's Degree
Needs	None
Prerequisites	None
Reference books	[1] William G. Sullivan, Elin M. Wicks, C. Patric Koclling, Engineering Economy. 15th edition, Prentice Hall, 2011. [2] L.T. A. Blank, A. J. Tarquin, Engineering Economy. 6h edition, McGraw-Hill, New York, 2005.

Course Objectives:

1. Understand the importance and necessity of engineering economics.
2. Learn how to draw liquidity flow and estimate parameters affecting liquidity flow.
3. Understand mathematical relations to determine economic parameters.
4. Use tables of factors in economic calculations.
5. Analyze liquidity flow with geometric slope and specific liquidity flow modes.
6. Differentiate between nominal and practical interest and their effects on the content.
7. Explore economic evaluation techniques such as IRR, B/C, NEUA, and NPV.
8. Analyze return on investment and projects with multiple rates of return.
9. Make economic decisions based on the decision horizon.
10. Learn methods of calculating depreciation, taxes, and tax savings.
11. Conduct economic studies considering taxes and depreciation.
12. Explore alternative studies in economic evaluations.
13. Perform sensitivity analysis in economic studies.
14. Investigate the effect of inflation in economic studies.
15. Apply examples and applications of economic studies.

1.9. Electronic Commerce

In this course, students will be introduced to the main concepts of business and IT applications in e-commerce.

Name of Course	Electronic Commerce
Number of Credits	3
Number of Hours	48
Cross section	Bachelor's Degree
Needs	None
Prerequisites	Computer Networks - Engineering Economy
Reference books	[1] R. Kalakota, A.B. Whinston, and T. Stone, Frontiers of Electronic Commerce, Addison-Wesley, 1996. [2] S. Solomon, Marketing. Student Edition, Prentice-Hall, 1996. [3] P. Koiler, and G. Amstrong, Principles of Marketing, & Edition, Prentice-Hall, 1998. [4] D. Kosiur, Understanding Electronic Commerce, Microsoft Press, 1997. [5] S.L. Huff, CASES in Electronic Comnrerce, 2 Edition. McGraw-Hill, 2002. [6] R. Reddy, and S. Reddy, Supply Chains to Virtual Integration, McGraw-Hill. 2002. [7] W. Raisch, The Marketplace: Strategies for Success in B2B eCommerce, McGraw-Hill, 2001.

Course Objectives:

1. Introduction to E-Commerce
2. Understanding the emergence of knowledge-based business
3. Exploring the value in a network economy
4. Understanding the concepts of factory and virtual organization
5. Exploring product development in the digital economy
6. Understanding marketing in the digital economy
7. Learning about product management and trading services
8. Exploring strategic planning and the trading process
9. Understanding security in e-commerce
10. Learning about e-commerce infrastructure
11. Exploring e-commerce software
12. Understanding search strategies
13. Exploring applications of software agents in business

2

Main Courses

2.1. Fundamentals of Computer Programming

The purpose of this course is to introduce students to the fundamentals of computers and programming. Initially, students will learn about the basic structure of a computer and the fundamental concepts of computing in hardware. They will also become familiar with organizing the components of a modern computer. Additionally, this course emphasizes programming in Java, focusing on writing engineering code with principles such as modularity, cleanliness, proper commenting, spacing, and the ability to implement pseudocode.

Name of Course	Fundamentals of Computer Programming
Number of Credits	3
Number of Hours	48
Cross section	Bachelor's Degree
Needs	None
Prerequisites	None
Reference books	[1] P. Deitel and 11. Deitel, Java: How to Program. 6th Edition, Prentice Hall, 2009.

At the end of the module, students will be able to:

- Familiarize themselves with the basics of computer and programming.
- Gain proficiency in programming in Java.
- Develop the ability to write and implement pseudocode.

Course Objectives:

1. Understand basic concepts.
2. Perform calculations on the computer.
3. Learn programming basics.
4. Master output/input formatting.

5. Gain familiarity with algorithms, flowcharts, and pseudocodes.
6. Understand loops.
7. Learn about functions.
8. Gain familiarity with program testing and troubleshooting.
9. Understand arrays.
10. Learn about characters and strings.
11. Understand structures.
12. Perform inputs and outputs with files.

2.2. Advanced Computer Programming

The purpose of this lesson is to explore various approaches to developing high-quality programs. Students will first learn about the top-down design method for problem-solving. Subsequently, they will be introduced to the concepts of object-oriented programming as a means of managing complexity in medium and large-sized applications. Throughout the lesson, emphasis will be placed on ensuring proper program functionality through testing and debugging techniques such as unit tests, assertions, and pre- and post-conditions. The focus of the lesson will be on methodologies rather than the advantages of any particular programming language. This lesson can be taught using any commonly used object-oriented programming language, such as Java or C++.

Name of Course	Advanced Computer Programming
Number of Credits	3
Number of Hours	48
Cross section	Bachelor's Degree
Needs	None
Prerequisites	Fundamentals of Computer Programming
Reference books	[1] H.M. Deitet and P.J. Deitel, C++ How to Program. Sth ed., Prentice-Hall Inc., 2011. [2] P. Deitel and H. Dcitel, Java: How to Program. 9th Edition, Prentice Hall Inc., 2011.

At the end of the module, students will be able to:

- Solve problems using the top-down design method.
- Manage the complexity of programming problems by defining appropriate classes.
- Utilize appropriate abstraction methods such as inheritance and polymorphism.
- Employ essential features of programming language libraries.
- Demonstrate the ability to use necessary methods for testing and debugging programs.

Course Objectives:

1. Provide an overview of the basics of programming.
2. Teach top-down design principles.
3. Introduce the basic concepts of objectives.
4. Cover the basic structures of object-oriented programming.
5. Discuss inheritance and polymorphism.
6. Explain memory management.
7. Explore general programming concepts.
8. Teach error handling techniques.
9. Introduce input/output libraries.

10. Cover standard data structure libraries.
11. Explain how to create a graphical user interface.
12. Discuss text processing and string manipulation.
13. Provide an introduction to concurrent programming.
14. Teach how to test and debug programs effectively.

2.3. Discrete Mathematics

The purpose of this course is to introduce students to the concepts, structures, and techniques of discrete mathematics, which are widely used in computer science and engineering. The course aims to develop basic skills, including understanding and constructing precise mathematical proofs, as well as fostering creative thinking in problem-solving. Students will become familiar with foundational topics in number theory, logic, combinations, and graph theory. Moreover, the course aims to provide the mathematical prerequisites required for many other courses offered in various computer engineering majors.

Name of Course	Discrete Mathematics
Number of Credits	3
Number of Hours	48
Cross section	Bachelor's Degree
Needs	General Mathematics I (Calculus I) - Fundamentals of Computer Programming
Prerequisites	Does no have
Reference books	[1] K. H. Rosen, Discrete Mathematics and its Applications. 6th Edition, McGraw-Hill Inc., 2007.

At the end of the module, students will be able to:

- Apply mathematical reasoning and argumentation methods to solve problems effectively.
- Utilize combination methods and counting techniques to solve problems related to discrete mathematics.
- Analyze and work with graphs and trees, understanding their properties and applications.

Course Objectives:

1. Understand the fundamentals of Mathematical Logic.
2. Learn about the Theory of Functions and Sets.
3. Study Number Theory.
4. Understand the concept of Induction.
5. Explore Counting techniques.
6. Learn about Reciprocal Relations.
7. Study Relationships and Partial Orders.
8. Understand Algebra Bull.
9. Explore Graph Theory.
10. Study Trees.
11. Develop proficiency in using the elementary vocabulary of discrete mathematics and performing logic, algebraic, and algorithmic calculations.

12. Solve combinatorial problems.
13. Model and solve problems using Graph Theory.
14. Perform quantitative analysis of the efficiency of algorithms.

2.4. Signal and Systems

Students are acquainted with describing signals and analyzing linear and time-invariant systems in the domains of time and frequency.

Name of Course	Signal and Systems
Number of Credits	3
Number of Hours	48
Cross section	Bachelor's Degree
Needs	None
Prerequisites	Engineering Mathematics
Reference books	[1] A.V. Oppenheim, A. S. Willsky, and S.II. Nawab, Signals and Systems. 2nd ed. Prentice-Hall, 1996. [2] R. E. Zicmer, W.11. Tránter, and D. R. Fannin, Signals and Systems , Continuous and Discrete. 4th., Prentice-Hall, 1998. [3] S. Haykin and B. Van Veen, Signals and Systems. 2nd ed. Wiley, 2003.

Course Objectives:

1. Introduction: Introduce mathematical concepts and tools of signal processing and systems analysis. Define, describe properties, and classify signals.
2. Continuous and Discrete-Time Signals: Understand mathematical representations of signals including impulse, step, sinusoidal, exponential, power, and energy signals.
3. Linear and Time-Invariant Systems: Study systems with impulse response, convolution concept, properties of impulse response, and description using differential equations.
4. Fourier Series: Learn about intermittent signal representation using Fourier series, its convergence, properties, and its application in calculating the response of LTI systems.
5. Fourier Transform: Understand continuous-time signal representation using Fourier transform, its definition, convergence, properties, and analysis of systems described by differential equations with constant coefficients.
6. Discrete Fourier Transform: Study the discrete-time signal representation using Discrete Fourier Transform, its definition, convergence, properties, and analysis of systems described by differential equations with constant coefficients.
7. Sampling: Understand the concepts of ideal and non-ideal sampling, aliasing phenomenon, and signal reconstruction.
8. Laplace Transform: Learn about Laplace transform, its definition, convergence, properties, and its application in converting and analyzing continuous-time LTI systems.
9. Z-Transform: Understand Z-transform, its definition, convergence, properties, and its application in converting and analyzing discrete-time LTI systems.

2.5. Data Structures

Theoretical and practical exercises are designed to deepen the knowledge imparted, enabling students to design, analyze, and implement data structures and algorithms independently, or sensibly incorporate them into their programs. Students understand how the choice of a data structure impacts the performance and usability of applications, such as web browsing, searching, computer databases, data analysis, and text processing. Specific topics covered include lists, stacks, queues, sets, maps, priority queues, trees, and graphs, along with general algorithmic techniques like sorting, searching, and other data transformations. Upon completion of the course, students are equipped to design and develop efficient programs for a wide range of applications using these tools.

Name of Course	Data Structures
Number of Credits	3
Number of Hours	48
Cross section	Bachelor's Degree
Needs	General Mathematics I (Calculus I) - Fundamentals of Computer Programming
Prerequisites	Does no have
Reference books	[1] T. Cormen, C. Leiscrson, and R. Rivest. Introduction to Algorithms. McGraw-Hill Inc., 2001.

At the end of the module, students will be able to:

- Utilize existing data structures effectively and appropriately.
- Design diverse data structures as required.
- Design and implement various algorithms for accessing and processing data.
- Analyze the performance of data structures and algorithms in terms of time and space complexity.

Course Objectives:

1. Algorithm Analysis Methods:
    - Growth function analysis.
    - Counting steps.
    - Recursive relations and solving methods including conjecture and induction.
    - Iteration with substitution and the use of the Master theorem.
    - Breakdown analysis.
2. Types of Lists:
    - One-way, two-way, general, hash, and stack lists.
    - Various operations on lists.
    - Utilization of actual and dummy pointers.

- Implementation of various problems using lists (manipulating mathematical expressions, garbage collection, integrated sorting).

3. Trees:
   - Basic definitions.
   - Phrase trees.
   - Different tree implementations.
   - Tree induction.
   - Navigation and structural induction on trees, including binary trees.
   - Various operations on phrase trees.
   - Conversion between different versions of trees.
   - Binary tree search.
4. Hashing:
   - Interlocking method in global and open chaining.
5. Sorting and Searching:
   - Sorting and statistical ranking including the Bottom-Up approach, Decision Tree, and Linear Sorting (Counting, Radix, and Bucket).
   - Quick Sorting.
   - Heap Sorting.
   - Merge Sorting.
   - Statistical Sorting.
   - External Sorting.
6. Advanced Data Structures:
   - Disjoint Sets.
   - Red-Black Trees.
   - Order-Statistic Tree.
   - Interval Tree.
   - B-Tree.

2.6. Technical English

The aim of this course is to enhance students' proficiency in reading and comprehending English texts related to computer engineering and information technology, as well as to understand scientific lectures in this field to some extent. The course focuses on improving students' skills through consistent practice in reading, writing, and listening to scientific lectures in English.

Name of Course	Specialized English
Number of Credits	2
Number of Hours	32
Cross section	Bachelor's Degree
Needs	None
Prerequisites	General Language
Reference books	[1] Selected short articles on Computer Engineering and Information Technology (from different authors) [2] TED group scientific lectures [3] EE Times [4] IEEE Spectrum Magazine

At the end of the module, students will be able to:

- Read technical texts in computer engineering with medium complexity fluently.
- Understand the meaning of the text relatively well while reading.
- Comprehend technical lectures in the field of computer engineering to a reasonable extent.
- Write simple technical texts at an appropriate pace.

Course Objectives:

1. Browser Security
2. Cloud Computing
3. Network Security
4. Introduction to FPGA (Field-Programmable Gate Arrays)
5. Introduction to Linux
6. Introduction to Service-Oriented Architecture
7. Data Communication Channels
8. BitTorrent Protocol
9. Search Engine Optimization (SEO)
10. Operating Systems
11. Programming Languages

2.7. Electric Circuits

The objectives of the course include familiarizing students with the principles, theorems, and methods for analyzing electrical circuits. Additionally, the course aims to equip students with the ability to analyze the dynamic behavior of circuits as energy systems.

Name of Course	Electric Circuits
Number of Credits	3
Number of Hours	48
Cross section	Bachelor's Degree
Needs	None
Prerequisites	Differential Equations
Reference books	[1] L. O. Chua, C. A. Desoer, and E. S. Kuh, Linear and Nonlinear Circuits. McGraw Hill, 1987. [2] J. W. Nilson, Electric Circuits. 4" Edition, Addison Wesicy, 1995. [3] RJ. Smith and R. C. Dorf, Circuits,Devices, and Systems, sih Edition, John Wiley, 1992.

Course Objectives:

1. Rules and Definitions:
    - Understanding compact and complex circuits
    - Applying voltage and current laws
    - Recognizing ideal and real circuit elements (resistor, capacitor, inductor, voltage sources, and current sources)
    - Exploring two-terminal elements (dependent voltage and current sources, transistor model, and operational amplifier)
    - Understanding power and energy concepts
    - Identifying active and passive elements
    - Analyzing waveform types (step, pulse, impulse, and sinusoidal)
2. General Circuit Analysis:
    - Understanding linearity and time invariance concepts
    - Analyzing zero-state and zero-input responses
    - Studying transient and steady-state responses
    - Analyzing responses in the time and frequency domains
    - Applying node and mesh analysis methods
3. Time Domain Circuit Analysis:
    - Analyzing simple circuits
    - Understanding ordinary circuits
    - Exploring step and impulse response concepts
    - Studying second-order and higher-order circuits
4. Convolution Theorem and Its Applications:

- Understanding convolution integral
- Analyzing zero-input response of linear circuits

5. Frequency Domain Circuit Analysis:
    - Applying Laplace transform
    - Using Laplace transforms in electrical circuit analysis
    - Understanding Fourier series
    - Analyzing sinusoidal steady-state response
    - Exploring network function concept and its relation to impulse response
    - Understanding frequency response
6. Network Theorems and Their Applications:
    - Application of superposition theorem
    - Utilizing Norton's theorem
    - Understanding maximum power transfer theorem
7. Familiarity with SPICE Simulator Software and Its Use in Electrical Circuit Analysis

2.8. Operating Systems Lab

This course aims to cultivate a solid understanding and proficiency in designing and implementing operating systems, focusing on open-source platforms. Students will explore fundamental concepts of operating system design, including file management, process management, kernel-level programming, process synchronization in a Unix-like operating system, and the installation and administration of open-source operating systems. The course also includes laboratory sessions and practical exercises to reinforce learning.

Name of Course	Operating Systems Lab
Number of Credits	1
Number of Hours	16
Cross section	Bachelor's Degree
Needs	Operating Systems
Prerequisites	None
Reference books	[1] M. K. Dalheimer, T. Dawson, L. Kaufman, M. Welsh, Running Linux. O'Reilly, 2002 [2] K. Wall, M. Watson, and M. Whitis, Line Programming Unleashed. Sams PublishersInc., 1999.

At the end of the module, students will be able to:

- Install and administer an open-source operating system
- Implement fundamental concepts of operating system design within an open-source operating system environment
- Perform kernel-level programming tasks

Course Objectives:

1. Introduction to Linux: Overview of Linux, including its History, Versions, and POSIX standards. Introduction to the Graphical User Interface (GUI).
2. Linux Installation: Disk partitioning, Boot loading, Application menu usage, System configuration.
3. Linux File System: Understanding the Linux file system structure and operations.
4. Standard and Advanced Shell: Utilizing standard and advanced shell commands for efficient navigation and file manipulation.
5. Process and Thread Management: Understanding process creation, termination, and manipulation. Introduction to threads and their management.
6. CPU Scheduling in Linux: Exploring CPU scheduling algorithms and their implementation in Linux.
7. Synchronization and Deadlocks: Managing synchronization and avoiding deadlocks in Linux-based systems.

8. Kernel Programming and System Services: Introduction to kernel programming techniques and development of system services within the Linux environment.

2.9. Logic Circuits

This course encompasses the design and implementation of digital logic circuits, covering both combination and sequential logic circuits. By the end of this course, students will be proficient in recognizing and applying the following concepts, ideas, and tools:

1. Logic Level Models: Understanding Boolean algebra, finite state machines, arithmetic circuits, and hardware description languages.
2. Logic Gates and Memory: Familiarity with CMOS gates, flip-flops, memory arrays, and programmable logic devices.
3. Design Tools: Utilizing both manual and computerized tools for the design, optimization, and testing of logic circuits.
4. Design Criteria: Considering factors such as area, speed, power consumption, and testability when designing logic circuits.

Name of Course	Logic Circuits
Number of Credits	3
Number of Hours	48
Cross section	Bachelor's Degree
Needs	Discrete Mathematics
Prerequisites	None
Reference books	[1] S. Brown and Z. Vrancsic, Fundamentals of Digital Logic with Verilog Design. 3 Edition, McGraw-Ilill, 2009. [2] C.H. Roth and L.L. Kinncy, Fundamentals of Logic Design. 5° Edition, 2005. [3] J. Wakerly. Digital Design, Principles and Practices. 4" Edition. 2005. [4] Victor P. Nelson, II. Troy Nagle, Bill D. Carroll, and David Irwin, Digital Logic Circuit Analysis and Design. Prentice Hall, 1995.

At the end of the module, students will be able to:

- Understand the basic concepts of digital systems and circuits
- Design digital systems effectively
- Analyze digital systems comprehensively
- Model digital systems accurately

Course Objectives:

1. Introduction and basic concepts
    - Understanding the history and applications of digital systems
    - Differentiating between digital and analog systems
    - Overview of digital circuits based on MOS transistors
2. Numerical systems

- Understanding number theory and representation
- Performing computations in digital systems
- Handling carry and overflow concepts
- Understanding LCD display systems

3. Boolean Algebra

   - Grasping the principles of Boolean Algebra
   - Understanding functions, operators, and logic gates
   - Applying Boolean algebra in various logical functions
   - Learning hardware description languages (VHDL or Verilog) at the structural level

4. Analysis and design of combinational logic systems

   - Simplifying combinational circuits using Boolean algebra
   - Optimizing circuits using Carnot table and Queen-McCluskey algorithm
   - Understanding glitch and hazard race concepts
   - Implementing circuits with two-story designs
   - Designing circuits for coding, decoding, division, etc.
   - Utilizing universal kits and programmable circuits

5. Analysis and design of sequential logic systems

   - Introducing memory elements such as latches and flip-flops
   - Analyzing sequential circuits using excitation tables and state diagrams
   - Designing sequential circuits with different flip-flop types
   - Understanding counters, registers, and shifters
   - Introduction to standard sequential chips

6. Basics of asynchronous circuit design

2.10. Logic Circuits and Computer Architecture Laboratory

In logic circuit and computer architecture courses, students gain hands-on experience with concepts like digital circuit design, analysis, and debugging. They learn to design the logical and arithmetic components of a processor, including memory units, input/output interfaces, and the data path and control mechanisms typical in processors. Students also use descriptive software languages like Verilog and VHDL to simulate, synthesize, and implement these designs, typically on FPGA boards, working at both gate and RTL abstraction levels.

Name of Course	Logic Circuits and Computer Architecture Laboratory
Number of Credits	1
Number of Hours	16
Cross section	Bachelor's Degree
Needs	Computer Architecture
Prerequisites	Logic Circuits
Reference books	[1] S. Brown and Z. Vranesic, Fundamentals of Digital Logic with Verilog Design. McGraw-Hill, 2003. [2] B. Parhami, Computer Arithmetic - Algorithms and Hardware Designs, Oxford Univ.Press, 2000. [3] D.A. Patterson and J. L. llennessy, Computer Organization and Design: The Hardware, Software Interfacc, 4 Edition, Morgan Kaufman Publisher Inc., 2010. [4] J. L. Hennessy and D.A. Patterson, Computer Architecture, A Quantitative Approach,Prentice-Hall, 4th cdition. [5] D.M, Harris, Digital Design and Computer Architecture, 24 Edition, Morgan Kaufman Publisher Inc., 2012.

At the end of the module, students will be able to:

- Design digital systems using SSL chips and debug them effectively.
- Gain familiarity with designing various components of a typical processor and its peripheral units.

Course Objectives:

1. Use schematic design software to simulate sequential and hybrid digital circuits within the FPGA environment.
2. Utilize simulation software based on hardware description languages to verify synthesis results of various structures.
3. Design hybrid circuits using SSI chips and analyze their Verilog equivalents with VIDI.
4. Understand the performance of counters and registers, and implement their Verilog equivalents using VHDL.
5. Design RAM, ROM, and dual-port memory units.

6. Design adders, multipliers, and analyze their size and speed. Analyze and implement their Verilog or VHDL equivalents.
7. Implement synchronous transmission in one-way and two-way channels.
8. Implement arbitration mechanisms for crossings.
9. Implement and evaluate pipeline acceleration.
10. Develop a binary divider using complementary logic.
11. Design and implement a desktop computer system.
12. Design memory hierarchy structures.
13. Measure CPI, IPC, and MIPS parameters in a typical processor.

2.11. Computer Networks

This course explores the fundamentals of designing, implementing, and optimizing computer networks. Students will delve into network architecture, services, and the OSI model. Emphasis is placed on Internet networks and the TCP/IP model, with a focus on application layer protocols. Additionally, the course covers the transport, network, and data link layers of network communication.

Name of Course	Computer Networks
Number of Credits	3
Number of Hours	48
Cross section	Bachelor's Degree
Needs	None
Prerequisites	Operating Systems
Reference books	[1] James F. Kurose and Keith W. Ross, Computer Networking: 4 Top-DowwApproach. 5th edition, Addison-Wesley Inc., 2009. [2] Alberto Leon-Garcia and Indra Widjaja, Communication Networks. 20 edition.McGraw-Hill Inc., 2003.

Course Objectives:

1. Understand various computer network services (examples, definitions, service quality, protocol fundamentals).
2. Explore components of the Internet network (network edge, core, Client-Server model, access networks, packet switching vs. circuit switching).
3. Examine the layered architecture of computer networks (OSI reference model, TCP/IP model, multiplexing, protocols).
4. Study the Application layer (web applications, HTTP, FTP, SMTP, DNS, peer-to-peer applications, socket programming).
5. Analyze the Transport layer (services, UDP, TCP, error control, congestion control).
6. Investigate the Network layer (routing, forwarding, router architecture, traffic management, IP protocols, routing algorithms, routing protocols like RIP, OSPF, BGP).
7. Understand the Data Link layer and Local Area Networks (services, error detection/correction, media access control, LAN basics).

2.12. Microprocessors and Assembly Language

This course equips students with fundamental knowledge of microprocessors and microcontrollers, enabling them to design systems based on these technologies. A significant focus is placed on the 8086/8088 microprocessor, chosen for its simplicity compared to more advanced processors, providing essential insights into microprocessor design concepts. Additionally, students will receive a brief overview of the more advanced Pentium microprocessor, highlighting differences such as addressing methods and operating states.

The course also introduces the AVR family microcontrollers, offering students hands-on experience with embedded systems concepts. Topics covered include memory types, programmable parallel ports, hardware and software interrupts, timer usage for timing events, generation of analog signals like PWM, analog-to-digital converters, and various serial communication protocols.

To facilitate learning assembly language, the course introduces assembly instructions for the 8084 microprocessor and the AVR family. Students will also become familiar with assemblers and integrated development environments (IDEs) to aid in their understanding and practice.

Name of Course	Microprocessors and Assembly Language
Number of Credits	3
Number of Hours	48
Cross section	Bachelor's Degree
Needs	None
Prerequisites	Computer Architecture
Reference books	[1] John Uffenbeck, The 8086/8088 Family: Design, Programming, and Interfacing, Prentice Hall, 3rd Edition, 2001. [2] ATmegal6 microcontroller datashcct. [3] James L. Antonakos, The Pentium Microprocessor, Prentice-Hall, 1998. [4] Holzner Steven Advanced Assembly Language, Prentice-Hall, 1995. [5] Intel Corporation, Intel Pentium Developer's Manual, Volume 3, 1995. [6] NASM Development Team, NASM-Netwide Assembler User Manual, 2012. [7] Richard H. Barnett, Sarah Cox, Larry O'Cull, Embedded C Programming and the Atmel AVR, Delnmar Cengage Learning Publishing, 2011. [8] CodeVision AVR C compiler, User manual, 2003. [9] AVR Assembler, Atmel, 2004. [10] Atmel Studio, Atmel. [11] Winavr User Guide.

At the end of the module, students will be able to:

- Explain the architecture and organization of a microprocessor.
- Develop assembly programs with proper structure and clear explanations.
- Implement methods to connect and manage different types of main memory and input-output devices with the microprocessor.
- Manage communication between the microprocessor and input-output devices.

- Utilize microcontroller components (e.g., timers, analog-to-digital converters, serial communication methods) in embedded systems applications.
- Employ integrated development environments (IDEs) as software tools for developing microprocessor-based systems and microcontrollers across various applications.

2.13. Microprocessors Laboratory

The microprocessor course laboratory aims to familiarize students with practical aspects of microprocessor systems. This includes tasks such as setting up reset circuits, generating clock signals, programming parallel ports, interfacing keyboards, 7-segment displays, and character providers, as well as communication with different types of memory.
Students will also learn to work with internal and external keys, prioritize them, utilize timer counters for event counting, timing, and generating PWM signals. They will gain experience with analog comparators and analog-to-digital converters for interfacing with analog signals, as well as serial communication protocols like SPI, USART, and I2C.
Additionally, students will practice using assemblers and compilers for microprocessors and microcontrollers. Finally, they will simulate microprocessor-based projects or microcontrollers using simulation environments like Proteus, and will design, construct, and test relevant circuit boards as part of their final project in the laboratory.

Name of Course	Microprocessor Laboratory
Number of Credits	1
Number of Hours	16
Cross section	Bachelor's Degree
Needs	None
Prerequisites	Microprocessors and Assembly Language
Reference books	[1] John Uffenbeck, The 8086/8088 Family: Design, Programming, and Interfacing, Prentice Hall, 3rd Edition, 2001. [2] ATmega16 microcontroller datasheet. [3] CodeVision AVR C compiler, User manual. [4] AVR Assembler, Atmel. [5] Atmel Studio, Atmel. [6] WinAVR user manual.

At the end of the module, students are able to:

- Gain practical experience and proficiency in working with microprocessors.
- Design both hardware and software components and assemble systems utilizing microprocessors and microcontrollers.

Course Objectives:

1. Introduce students to simulation software like Proteus for testing, and PCB design software such as Altium for schematic and PCB design. Students will complete a simple electronic project involving designing and simulating a flashing circuit with two LEDs (unstable multivibrator circuit), creating schematic and PCB designs, assembling components on the PCB, soldering, and testing.
2. Familiarize students with microcontroller assemblers and compilers such as Atmel Studio and CodeVision.

3. Construct reset circuits, program cables, program fuse bits, and generate clock signals for microcontrollers.
4. Develop programs for reset interrupts, set stack pointers, work with ports, implement software delay generation using polling, and utilize watchdog timers.
5. Implement external interrupts and utilize power-saving modes.
6. Read values from ports connected to a four-bit DIP switch, convert the read values to BCD, further convert BCD digits to 7-segment displays, and display the results.
7. Design and interface with a matrix keyboard to display numbers read from the keyboard using 7-segment displays.
8. Connect and interface LCDs with microcontrollers to display information received from the keyboard.
9. Read and write data from EEPROM microcontroller memory.
10. Utilize counter timers for various tasks, including flashing LEDs alternatively and creating a digital frequency meter.
11. Use timer 2 in PWM mode to control LED brightness or motor speed.
12. Interface with a microcontroller's analog comparator to control LED states based on voltage levels.
13. Measure temperature using an analog-to-digital converter and display the results on an LCD.
14. Establish communication between microcontrollers and computers via USART interface for data exchange.
15. Implement communication between microcontrollers using the SPI interface or perform data read/write operations in SD RAM.

2.14. Digital Systems Design

Understanding automated methods for designing and debugging digital circuits and systems involves utilizing tools for the automatic design of integrated circuits.

Name of Course	Digital Systems Design
Number of Credits	3
Number of Hours	48
Cross section	Bachelor's Degree
Needs	None
Prerequisites	Computer Architecture
Reference books	[1] S. Palnitkar, Verilog HDL: A Guide to Digital Design and Synthesis. SunSoft Press, 2nd ed. 2003. [2] V. A. Pedroni, Circuit Design with VHDL, MIT Press, 2011. [3] C. Maxfield, The Design Warrior's Guide to FPGAs: Devices, Tools and Flows.Elsevier Pub., 2004.

At the end of the module, students will be able to:

- Describe and design complex circuits and hardware systems using hardware description languages, with a focus on mastering complexity control skills.
- Utilize digital circuit design tools effectively.
- Gain familiarity with programmable chips such as FPGA and CPLD, understanding their internal architecture and leveraging their features for professional design purposes.

Course Objectives:

1. Introduction and Basic Concepts
    - Explore the history and evolution of digital systems.
    - Analyze the growth trends in the digital systems design industry.
    - Introduce automated hardware design tools and languages.
    - Compare ASIC and FPGA design cycles.
    - Discuss different hardware design styles.
    - Understand abstract levels of hardware design.
2. Hardware Description Languages
    - Examine the need for hardware description languages over schematic methods.
    - Identify key features of hardware description languages.
    - Explore concurrency as a distinguishing feature of hardware description languages.
    - Compare conventional hardware description languages.
    - Dive into Verilog/VHDL language features.
    - Compare Verilog/VHDL with other hardware description languages.

- Learn hardware simulation methods.

3. VHDL/Verilog Descriptive Language Training

   - Understand delay models in the target language.
   - Explore types of language data.
   - Implement hardware description methods at different levels (behavioral, data flow, and structural).
   - Utilize specific features of the chosen description language.
   - Design test benches.
   - Design sequential and functional blocks.
   - Implement parametric designs using generics.
   - Manage complexity in large hardware designs.
   - Employ organizing techniques.
   - Understand top-down and bottom-up design methods.
   - Describe state machines and coding methods.
   - Design pipelines and describe stability transfer.

4. Hardware Synthesis

   - Understand concepts of behavioral, logical, and physical synthesis.
   - Outline steps for logical synthesis.
   - Address considerations in describing a synthesis constraint subset.
   - Perform simulation and testing post-synthesis.
   - Design based on constraints.
   - Utilize static timing analysis (STA) methods and Slack parameter.
   - Optimize design criteria for area, speed, and power consumption using tools.
   - Overview high-speed and low-power circuit design techniques.
   - Overview test circuit design techniques.

2.15. Design of Algorithms

The aim of this course is to instruct students in the analysis and design of algorithms. Throughout the course, students will learn how to systematically analyze problems and devise a range of algorithms to address them. They will explore algorithmic solutions for each problem type, analyze and compare these solutions in terms of computational complexity, and ultimately select the most suitable algorithm for a given engineering application based on the problem's size and input characteristics. The course will cover fundamental algorithms for solving practical and common problems, providing students with a solid foundation in algorithmic thinking and problem-solving techniques.

Name of Course	Design of Algorithms
Number of Credits	3
Number of Hours	48
Cross section	Bachelor's Degree
Needs	None
Prerequisites	Data Structures
Reference books	[1] T. Cormen, C. Leiserson, and R. Rivest. Introduction to Algorithms. McGraw-HillInc., 2001.

At the end of the module, students will be able to:

- Develop a general understanding of algorithmic problem-solving methods.
- Gain proficiency in identifying and proving NP-complete problems.
- Acquire familiarity with basic graph algorithms.
- Calculate the time complexity of algorithms.
- Comprehend common and essential algorithms, comparing their solutions based on complexity, and determine suitable applications for each.
- Utilize existing library functions for standard algorithms effectively.

Course Objectives:

1. Explore various problem-solving techniques.
2. Study dynamic programming algorithms.
3. Analyze knapsack problems and their solutions.
4. Understand greedy algorithms and their applications.
5. Solve interval scheduling problems.
6. Learn breakpoint analysis techniques.
7. Master advanced data structures.
8. Understand disjoint sets and their applications.
9. Investigate algorithms for finding shortest paths between all pairs of graph nodes.
10. Study maximum flow algorithms.
11. Explore string matching algorithms.

2.16. Artificial Intelligence and Expert Systems

Artificial intelligence stands as a cornerstone in computer science, aspiring to replicate human intelligence and exceptional capabilities. This field encompasses a broad spectrum of sub-disciplines, ranging from general concepts to highly specialized topics.

This course delves into machine learning, covering a range of concepts from fundamental principles to specific applications like chess playing, automated theorem proving, and disease diagnosis. While introducing the foundational concepts of artificial intelligence, including various search methods and knowledge representation techniques, the course also provides a brief overview of some sub-branches within the field.

Name of Course	Artificial Intelligence Expert Systems
Number of Credits	3
Number of Hours	48
Cross section	Bachelor's Degree
Needs	None
Prerequisites	Data Structures
Reference books	[1] S. Russel and P. Norvig Artificial Intelligence: A Modern Approach. 3rd edition, Prentice Hall, 2010. [2] C. S. Krishnamoorthy and S. Rajecy, Artificial Intelligence and Expert Systems for Engineers. CRC Press, 1996.

At the end of the module, students will be able to:

- Understand the structure and functioning of intelligent agents.
- Identify solutions to artificial intelligence problems using search methods.
- Comprehend the concept of initiative in problem-solving within artificial intelligence.
- Acquire knowledge of factors influencing knowledge-based systems.
- Gain familiarity with first-order logic as a language for knowledge representation in knowledge-based agents.
- Understand planning methodologies.
- Learn approaches for solving artificial intelligence problems in uncertain environments.
- Grasp the concept of learning through observations.
- Familiarize themselves with the concept of robots, perception, inference, and execution in robotic systems.

Course Objectives:

1. Provide an introduction to artificial intelligence and its historical context.
2. Explore the concept of intelligent agents and their role in problem-solving.
3. Develop problem-solving skills using search algorithms.

4. Differentiate between various types of smart and non-smart search strategies.
5. Solve constraint satisfaction problems using search techniques.
6. Understand and apply techniques for minimum and maximum search.
7. Study logic-based factors, including resolution, propositional logic, and forward and backward chaining.
8. Analyze first-order logic and its inference types.
9. Examine systems based on logical inference.
10. Address uncertainties in problem-solving and explore systems based on possible inference.
11. Study decision-making systems in artificial intelligence.
12. Learn methods for building knowledge bases and different approaches to knowledge representation.

2.17. Computer Architecture

The goal of this course is to introduce computer engineering students to processor architecture and organization, covering instructional architecture and the internal structure of processors. Students will also gain proficiency in computer arithmetic used in general-purpose processors, including number representation and basic arithmetic operations like addition, subtraction, multiplication, and division in various numerical systems. Additionally, the course will cover memory hierarchy in processing systems.
As one of the objectives is to model and test different architectures, the use of hardware description languages like Verilog is recommended. Basic concepts of Verilog will be recalled and taught during the course to facilitate this objective.

Name of Course	Computer Architecture
Number of Credits	3
Number of Hours	48
Cross section	Bachelor's Degree
Needs	None
Prerequisites	Logic Circuits
Reference books	[1] D. A. Patterson and J. L. Hennessy, Computer Organization and Design: The Hardware/Software Interface. 4 Edition, Morgan Kaufmann Publishers Inc., 2010.

At the end of the module, students will be able to:

- Understand different low-order/predestined architectures.
- Analyze processor performance.
- Design and implement processors.
- Implement computer arithmetic algorithms in processors.
- Design peripherals and establish their connection to processors.
- Gain proficiency in the Or Bello language and simulate fundamental structures of computer architecture using it.

Course Objectives:

1. Provide an introduction to the course.
2. Introduce basic concepts of computer architecture and organization.
3. Familiarize students with the Verilog hardware modeling language.
4. Explore processor design principles.
5. Introduce the pipeline mechanism in processor architecture.
6. Discuss memory hierarchy in computing systems.
7. Cover computer arithmetic algorithms.
8. Study CPU peripherals and their integration with processors.
9. Introduce multicore processors and their architecture.

2.18. Operating Systems

This course aims to cultivate a comprehensive understanding of the interplay between application software and hardware, along with resource management methods and algorithms for computer engineering students. It delves into robust techniques for managing low-level computer systems, emphasizing the design of such systems while considering hardware and software characteristics and limitations. Additionally, the course focuses on enhancing program quality. It combines theoretical concepts with practical applications to provide a holistic learning experience.

Name of Course	Operating Systems
Number of Credits	3
Number of Hours	48
Cross section	Bachelor's Degree
Needs	None
Prerequisites	Logic Circuits - Computer Architecture
Reference books	[1] P. Silberschatz, B. Galvin, and G. Gagne, Operating System Concepts. 81h Edition,John Wiley Inc., 2010.

At the end of the module, students will be able to:

- Identify various types of computer systems and their respective applications.
- Design, construct, and administer software systems effectively.
- Analyze the factors contributing to the decline in efficiency of computer systems and implement solutions to address these issues.
- Develop resource management policies tailored to system conditions.

Course Objectives:

1. Provide an introduction to operating system structures.
2. Cover process management concepts.
3. Explore thread management.
4. Study CPU scheduling algorithms.
5. Understand synchronization of processes.
6. Address deadlock management strategies.
7. Learn about main memory management techniques.
8. Discuss secondary memory management.
9. Study input-output management in operating systems.
10. Examine mass memory structures and their management.

2.19. The Theory of Formal Languages and Automata

This course delves into the theoretical aspects of computer engineering, exploring the relationship between problems and languages. It covers various computational models, their capabilities, formal expression of models and grammars, computational properties, and applications. Additionally, the course addresses concepts of computability, decision-making, and Church and Turing theses on algorithms. It provides foundational knowledge essential for compiler courses, algorithm design, computational theory, and courses related to formal description and modeling of computer systems.

Name of Course	The Theory of Formal Languages and Automata
Number of Credits	3
Number of Hours	48
Cross section	Bachelor's Degree
Needs	None
Prerequisites	Data Structures
Reference books	[1] P. Linz, An introduction to formal languages and automata, 5th Edition, Joncs and Barlett Publishers, 2011. [2] M. Sipser, Introduction to the theory of computation. 2nd Edition, PWS Publishing Company, 2006.

At the end of the module, students will be able to:

- Gain basic knowledge for identifying problems that are solvable and undecidable.
- Develop the ability to write grammars to solve various problems in different languages.
- Design machines capable of distinguishing language strings from different classes.

Course Objectives:

1. Introduction to foundational topics: propositional logic, set theory including Russell's paradox, finite and infinite sets, languages, grammars, and uncertainty theory.
2. Study of regular languages: deterministic finite automata, nondeterministic finite automata, conversion from nondeterministic to deterministic finite automata, minimization of deterministic finite automata, regular expressions, linear right-hand grammars, linear left-hand grammars, regular grammars, closure properties of regular languages, decision problems for regular languages, irregular languages, pumping lemma for regular languages.
3. Investigation of context-free languages: context-free grammars, context-free languages, parse trees, rightmost derivation, leftmost derivation, derivation trees, ambiguous grammars, unambiguous grammars, inherently ambiguous languages, context-free languages closure properties, Chomsky normal form,

membership problem, CYK algorithm, pushdown automata, equivalence of pushdown automata and context-free grammars, deterministic pushdown automata, deterministic context-free languages, non-context-free languages, pumping lemma for context-free languages, closure properties and decision problems for context-free languages.

4. Study of context-sensitive languages, linear-bounded automata, and context-sensitive grammars.
5. Exploration of recursively enumerable languages, Turing machines, types of Turing machines, recursively enumerable grammars.
6. Understanding the Chomsky hierarchy of formal languages.
7. Introduction to computability theory.

2.20. Computer Networks Laboratory

This course aims to establish a comprehensive understanding and practical experience of the fundamental concepts of computer networks. To achieve this goal, students will be introduced to the tools and equipment essential for setting up computer networks. The course will include hands-on experiments that reinforce theoretical topics covered in lectures. By engaging in practical exercises, students will gain a deeper understanding of computer network concepts and develop essential skills for designing and managing computer networks.

Name of Course	Computer Networks Laboratory
Number of Credits	1
Number of Hours	16
Cross section	Bachelor's Degree
Needs	Computer Networks
Prerequisites	None
Reference books	[1] S. Panwar, S. Mao, J. Ryoo , Y. Li , TCP/IP Essentials: A Lab-Based Approach. Cambridge University Press, 2004.

At the end of the module, students will be able to:

- Utilize troubleshooting tools effectively to diagnose network issues.
- Employ packet capturing tools to analyze network traffic.
- Utilize packet crafting tools for network testing and simulation.
- Configure network equipment to meet specific requirements.
- Set up a Local Area Network (LAN) with multiple Virtual LANs (VLANs).
- Establish connections between multiple LANs using both dynamic and static routing protocols.
- Compile and present test results in the form of technical reports.

Course Objectives:

1. Provide an overview of computer networking principles.
2. Introduce troubleshooting tools such as Ping, Traceroute, and Arpping for network diagnostics.
3. Familiarize students with packet capturing tools like TCPDump and Wireshark for network traffic analysis.
4. Enable students to create and send third- and fourth-layer packets using Packet Generator tools and modify field values with the Stay tool.
5. Guide students in setting up Client-Server connections.
6. Teach initial configuration of VLANs and Trunk configuration for switches and routers.
7. Instruct students in static and dynamic routing configuration, including OSPF and RIP protocols.

8. Assist students in setting up a Domain Name Server (DNS).
9. Guide students in configuring a DHCP server.

2.21. Engineering Mathematics

The class will cover fundamental mathematical methods for engineers, encompassing topics such as differentiation and integration, Taylor's expansion, resolution of linear systems, matrix formalism, partial differential equations, as well as Laplace, Fourier, and Legendre transforms. Additionally, it will include elements of statistics and probability. The focus of the class will primarily be on practical exercises and problem-solving, providing students with hands-on experience to reinforce theoretical concepts.

Name of Course	Engineering Mathematics
Number of Credits	3
Number of Hours	48
Cross section	Bachelor's Degree
Needs	None
Prerequisites	Calculus II - Differential Equations
Reference books	[1] E. Krcyszig, Advanced Engineering Mathematics. 10 cd., Wiley, 2011. [2] C.R. Wylie, Advanced Engineering Mathematics, 6h ed., McGraw-Hill, 1995.

Course Objectives:

1. Fourier Series and Fourier Transform: Understand the definition of Fourier series, O's formula, half-range expansions, induced Fourier integral by Toussaint, and applications of Fourier analysis in engineering problem-solving. Explore Gibbs' theorem, Fourier analysis constraints, convergence of series, and Fourier transform. Analyze symmetric properties of Fourier transform and Fourier series, and introduce the discrete Fourier transform.
2. Partial Differential Equations: Study the univariate wave equation, variable separation method, D'Alembert's solution for wave equation, heat conduction equation, wave equation, Laplace's equation in Cartesian, spherical, and polar coordinates, and classification of partial differential equations into elliptic, parabolic, and hyperbolic types. Learn to solve partial differential equations using the Fourier integral and derive Telegraph equations.
3. Analytic and Mapping Functions: Explore the concepts of limit and continuity, derivative of complex functions, exponential and logarithmic functions, hyperbolic and inverse trigonometric functions. Understand coherence mapping of Dirichlet boundary conditions, energy conservation, and its application in electrical capacitance calculations.
4. Complex Integrals in the Complex Plane: Study the Cauchy integral theorem, calculation of complex integrals using indefinite integrals, Cauchy's formula, Taylor and Maclaurin series, residue theorem, calculation of improper integrals using residues, and evaluation of real trigonometric integrals.

2.22. Principle of Compiler Design

Designing and constructing compilers is a foundational aspect of computer science. While the methods for building compilers may vary, they can be applied to create interpreters and translators for diverse languages and architectures. Following an initial overview of compiler components and grammar types, the course delves into various translation stages including lexical analysis, syntax analysis, semantic analysis, and code generation.

Name of Course	Principle of Compiler Design
Number of Credits	3
Number of Hours	48
Cross section	Bachelor's Degree
Needs	None
Prerequisites	Data Structures
Reference books	[1] Alfred V. Aho, Ravi Sethi, and Jeffrey D. Ullman, Compilers: Principles, Techniques, and Tools. Second Edition, Boston: Addison-Wesley, 2007.

At the end of the module, students will be able to:

- Understand compiler implementation and various implementation techniques.
- Interpret the execution of commands in programming languages.
- Develop skills in optimizing programs and debugging programming errors.
- Apply automated tools in compiler development.

Course Objectives:

1. Introduction to Compiler Design
2. Overview of Languages and Grammars
3. Lexical Analysis and Error Correction
4. Syntax Analysis
5. Top-Down Parsing Methods
6. Bottom-Up Parsing Methods
7. Operator Precedence Parsing
8. Simple Precedence Parsing
9. LR Parsing (LR(1), LALR(1), SLR(1), CLR(1))
10. Semantic Analysis
11. Symbol Table Management
12. Runtime Memory Allocation Techniques
13. Code Generation
14. Code Optimization and Payment
15. Automated Compiler Generation

2.23. Methods of Research and Presentation

The aim of this course is to introduce students to the principles and methodologies of research, as well as to develop skills in preparing various types of written presentations and delivering oral presentations. Topics covered include the implementation of different speech formats and the use of relevant tools in research.

Name of Course	Method of Research and Presentation
Number of Credits	2
Number of Hours	32
Cross section	Bachelor's Degree
Needs	None
Prerequisites	Technical English
Software	Word, PowerPoint, OneNote, Project, EndNote, .

At the end of the module, students will be able to:

- Select appropriate research topics. - Conduct research effectively. - Present research findings in the form of reports and articles. - Deliver technical presentations adhering to necessary principles. - Utilize relevant software for conducting research and presentations.

Course Objectives:

1. Introduction to Research: Understanding basic definitions and concepts.
2. Topic Selection and Title Formulation: Learning to define research topics and choose appropriate titles.
3. Research Planning: Developing a structured research plan.
4. Literature Review: Reviewing relevant literature, conducting studies, and taking notes.
5. Experimental Methodology: Understanding methods and essential considerations for conducting experimental research.
6. Report Writing: Learning important principles for writing engineering reports and preparing final design reports.
7. Written Presentations: Understanding specific considerations for different types of written presentations.
8. Speech Presentations: Learning principles for delivering effective speech presentations.
9. Presentation Techniques: Exploring special tips for different types of speech presentations.

3

Basic Course

3.1. General Mathematics I (Calculus I)

This calculus course encompasses the differentiation and integration of single-variable functions, culminating in a brief exploration of infinite series. Calculus serves as a cornerstone in various scientific fields, including physics, engineering, and economics.

Name of Course	General Mathematics I (Calculus I)
Number of Credits	3
Number of Hours	48
Cross section	Bachelor's Degree
Needs	None
Prerequisites	None
Reference books	[1] James Stewart, Single Variable Calculus: Concepts and Contexts. 4th edition, Cengage Learning, 2009. [2] George Simmons, Calculus with Analytic Geometry, 2nd edition, McGraw-Hill Science/Engineering/Math, 1996. [3] Tom Apostol, Calculus, Vol. 1: One-Variable Calculus, with an Introduction to Linear Algebra, Wiley: 2nd edition, 1991.

Course Objectives:

1. Cartesian and Polar Coordinates: Understanding coordinate systems.
2. Complex Numbers: Introduction to complex numbers, operations, and geometric representation.
3. Algebraic Functions: Study of algebraic functions.
4. Differentiation: Calculating derivatives, inverse functions, trigonometric derivatives, mean value theorem.
5. Applications of Differentiation: Geometric and physical applications, polar coordinates, root approximation.

6. Integration: Definite and indefinite integrals, fundamental theorems, integration techniques, approximate methods.
7. Applications of Integration: Area, volume, curve length, torque, center of gravity, work.
8. Exponential and Logarithmic Functions: Study of exponential, logarithmic, and hyperbolic functions and their derivatives.
9. Integration Techniques: Methods like substitution and partial fractions.
10. Sequences and Series: Understanding numerical series, power series, Taylor theorem, and Taylor expansion remainder.

3.2. General Mathematics II (Calculus II)

Calculus II serves as a progression from Calculus I, focusing on integration techniques and infinite series. It is tailored for students pursuing degrees in science, mathematics, computer science, and those preparing for specific graduate studies.

Name of Course	General Mathematics II (Calculus II)
Number of Credits	3
Number of Hours	48
Cross section	Bachelor's Degree
Needs	None
Prerequisites	Calculus I
Reference books	[1] James Stewart, Multivariable Calculus. Cengage Leaming, 7th edition, 2011. [2] Tom Apostol, Calculus, Vol. 2: Multi-Variable Calculus and Linear Algebra with Applications to Differential Equations and Probability. Wiley, 1969. [3] George Simmons, Calculus with Analytic Geometry, 2nd edition, McGraw-Hill Science/Engineering/Math, 1996. [4] Ron Larson and Bruce Edwards, Calculus Multivariable, Cengage Learning; 9th edition. 2009.

Course Objectives:

1. Exploring parametric equations
2. Understanding spatial coordinates
3. Analyzing vectors in space and various vector multiplications
4. Studying 3x3 matrices, solving three-dimensional linear equation systems, inverse matrices, linear independence, bases in $\mathbb{R}^3$ and $\mathbb{R}^2$, linear transformations, determinants of 3x3 matrices, eigenvalues, and eigenvectors
5. Investigating second-order lines, planes, and their equations
6. Understanding vectors and their derivative functions, velocity, acceleration, curvature, and normal vectors of curves
7. Exploring multivariable functions, including general and partial derivatives, tangent planes, gradient vectors, the chain rule for partial derivatives, and differentials
8. Learning about double and triple integrals and their applications in physics and engineering, including techniques for changing variables in multiple integrals (e.g., cylindrical and spherical coordinates)
9. Understanding vector fields, line integrals, surface integrals, divergence, curl, the divergence theorem, Stokes' theorem, Green's theorem, and applications in physics and engineering

3.3. Differential Equations

First-order ordinary differential equations and initial value problems; higher-order differential equations; vector spaces, matrices, determinants, eigenvectors, and eigenvalues; applications to systems of first-order equations; Laplace transforms.

Topics covered include:

- Solving first-order ordinary differential equations and initial value problems.
- Handling higher-order differential equations.
- Exploring vector spaces, matrices, determinants, eigenvectors, and eigenvalues.
- Applying these concepts to systems of first-order equations.
- Utilizing Laplace transforms.

Name of Course	Differential Equations
Number of Credits	3
Number of Hours	48
Cross section	Bachelor's Degree
Needs	None
Prerequisites	Calculus I
Reference books	[1] Yunus Cengel and William Palm, Differential Equations for Engineers and Scientists. McGraw-Hill Science Engineering Math, First edition, 2012.

Course Objectives:

1. Understanding the nature of differential equations and how to solve them.
2. Exploring families of curves and vertical solutions.
3. Understanding separable equations.
4. Solving first-order linear differential equations, including homogeneous equations.
5. Solving second-order linear equations, including homogeneous equations with constant coefficients. Methods include the method of indeterminate coefficients and the method of variation of parameters.
6. Applying second-order equations in physics and mechanics.
7. Solving differential equations using series, Bessel functions, and gamma functions.
8. Studying Legendre polynomials.
9. Introduction to systems of differential equations.
10. Understanding Laplace transform and its application in solving differential equations.

3.4. Engineering Statistics and Probability

After successfully completing the module, participants will:

1. Gain familiarity with essential concepts of discrete and continuous probability spaces and stochastic processes, enabling them to deduce these concepts independently to a large extent.
2. Master calculation rules for determining and estimating probabilities, expected values, and variances.
3. Develop the ability to map real-world problems to abstract probability spaces.
4. Acquire the skills to easily apply simple statistical tests.

Name of Course	Engineering Statistics and Probability
Number of Credits	3
Number of Hours	48
Cross section	Bachelor's Degree
Needs	None
Prerequisites	Calculus II
Reference books	[1] Alberto Lcon-Garcia, Probability, Statistics, and Random Processes for Electrical Engineering, Prentice Hall, 3rd edition, 2008. [2] Ronald E. Walpole, Raymond H. Myers, Sharon L. Myers, and Keying E.Ye. Probability and Statistics for Engineers and Scientists. Pearson, 9th edition, 2011.

Course Objectives:

1. Understanding the concepts of data sets, samples, and their tabular representation, including mean, mode, median, and variance.
2. Applying conversion and combination techniques for related probabilities and theorems.
3. Exploring continuous and discrete stochastic variables.
4. Calculating mean, median, and variance for binomial distributions, Poisson distributions, hypergeometric distributions, exponential distributions, and normal distributions.
5. Analyzing common distributions of stochastic variables, understanding correlation, and independence of variables.
6. Studying conditional distributions.
7. Exploring characteristic and moment generating functions.
8. Understanding the central limit theorem.
9. Applying Chebyshev's inequality.
10. Analyzing functions of random variables.
11. Relating the topics to statistics.

3.5. Physics I (Heat and Mechanics)

The course delves into the principles and applications of classical mechanics, covering topics such as harmonic motion, physical systems, and thermodynamics, with a focus on problem-solving. Additionally, students will conduct basic laboratory experiments to reinforce theoretical concepts in physics and their practical application in classical mechanics, including harmonic motion, physical systems, and thermodynamics.

Name of Course	Physics I
Number of Credits	3
Number of Hours	48
Cross section	Bachelor's Degree
Needs	None
Prerequisites	Calculus I
Reference books	[1] D. Halliday, R. Resnick, and J. Walker, Frondamentals of Physics. 9"edition, Wiley,2010.

Course Objectives:

1. Equilibrium: Understanding equilibrium conditions influenced by forces and torques, and relevant laws.
2. Motion in one and two dimensions: Exploring speed and acceleration, different types of motion, projectile motion, and gravitational effects.
3. Work and Energy: Introduction to work, kinetic energy, elastic potential energy, stable and unstable equilibrium, internal work, internal potential energy, power and velocity, and conservation of mechanical energy.
4. Impulse and Momentum: Studying the law of momentum conservation and laws of collision.
5. Rotation: Understanding angular velocity, angular acceleration, rotation with constant and variable angular acceleration, relationship between angular velocity and linear velocity, torque, rotational inertia, rotational kinetic energy, angular momentum conservation, and rolling motion around fixed and moving axes.
6. Temperature, Heat, and the First Law of Thermodynamics: Exploring temperature, heat, the first law of thermodynamics, the zeroth law of thermodynamics, and temperature measurement.
7. Kinetic Theory of Gases: Understanding ideal gases, transitional kinetic energy, mean free path, degrees of freedom, and molar specific heat.
8. Entropy and the Second Law of Thermodynamics: Exploring irreversible processes, change in entropy, and the second law of thermodynamics.

3.6. Physics II (Electricity and Magnetism)

Electric charge and Coulomb's law describe how charged particles interact with each other. The electric field is a concept used to understand the force experienced by a charged object due to other charges in its vicinity. Gauss' law is a principle used to calculate the electric field created by a distribution of charges.
Electric potential energy and potential are concepts used to describe the energy associated with electric charges in an electric field. These concepts help in understanding how work is done when moving charges in an electric field.
The electric properties of materials refer to how different materials behave in response to electric fields. This includes concepts such as conductivity, resistivity, and dielectric properties.
Capacitance and capacitors are devices used to store electric charge. They are essential components in electronic circuits for storing and releasing energy.
DC circuits involve the study of electric circuits with direct current (DC), where the flow of electric charge is constant and unidirectional. This includes analyzing the behavior of resistors, capacitors, and inductors in DC circuits.
The magnetic field is a field that surrounds a magnet or a current-carrying wire. It exerts a force on other magnets or moving charges. Ampere's law relates the magnetic field to the current flowing through a conductor.
Faraday's law of induction describes how a changing magnetic field induces an electromotive force (EMF) in a nearby conductor. This phenomenon is the basis for generating electricity in generators and transformers.
The magnetic properties of materials refer to how materials respond to magnetic fields. This includes concepts such as magnetization, permeability, and susceptibility.
Inductance is a property of a circuit element that opposes changes in current. It is a key factor in the behavior of circuits with changing currents, such as in AC circuits.
AC circuits involve the study of electric circuits with alternating current (AC), where the direction of the electric current reverses periodically. This includes analyzing the behavior of resistors, capacitors, and inductors in AC circuits.
Ampere's law relates the magnetic field around a closed loop to the electric current passing through the loop, providing a fundamental principle for understanding magnetism and its interaction with electric currents.

Name of Course	Physics II
Number of Credits	3
Number of Hours	48
Cross section	Bachelor's Degree
Needs	None
Prerequisites	Calculus I
Reference books	[1] D. Halliday, R. Resnick, and J. Walker, Fundamentals of Physics. 9th edition, Wiley,2010.

Course Objectives:

1. Electric Charge: Understanding electric charge, Coulomb's law, and the conservation of electric charge.
2. Electric Field: Exploring concepts such as electric field lines, the electric field of a point charge, the electric field of a charged line, the motion of a point charge in an electric field, and periodic motion in an electric field.
3. Gauss's Law: Understanding electric flux, the flux of an electric field, the relationship between Gauss's law and Coulomb's law, and applications of Gauss's law.
4. Electric Potential: Exploring electric potential energy, electric potential, potential due to a point charge, potential due to a group of charged particles, potential of a continuous distribution, and calculating potential using methods like the method of images.
5. Capacitance: Understanding capacitance, calculating energy stored in an electric field, capacitors with dielectrics, and the relationship between dielectrics and Gauss's law.
6. Electric Current and Electrical Resistance: Exploring concepts of current, resistance, Ohm's law, and power in electrical circuits.
7. Magnetic Fields: Understanding magnetic field strength, Hall effect, magnetic force on a moving charge, magnetic force on a current-carrying wire, torque on a current loop, and the magnetic field generated by a current using Ampere's law. Exploring magnetic dipoles and their properties.
8. Magnetic Field from Electric Current: Calculating magnetic fields from current using Ampere's law, magnetic field torque, and magnetic dipoles.
9. Induction: Understanding Faraday's law, induction, energy transfer, electric fields induced by induction, inductors, RL circuits, energy stored in magnetic fields, and mutual induction.
10. Electromagnetic Oscillations and Alternating Current: Exploring LC oscillations, alternating current circuits, power in RLC circuits, and Maxwell's equations including displacement current.
11. Electromagnetic Waves: Understanding moving electromagnetic waves, energy transfer, and the Poynting vector.
12. Wave-Particle Duality: Familiarity with the duality of waves and particles, including topics such as light interference, light diffraction, relativity, and radiation from hydrogen atoms.

3.7. Computer Workshop

In the laboratory sessions of the Computer Basics course, students will engage in hands-on activities and practical exercises to reinforce their understanding of fundamental computer concepts. These sessions aim to provide students with practical skills that complement the theoretical knowledge gained in lectures. The laboratory activities will cover various topics, including:

1. Operating Systems: Students will become familiar with different operating systems such as Windows, macOS, and Linux. They will learn how to navigate the user interface, manage files and folders, and perform basic system configurations.
2. Word Processing: Practical exercises will focus on word processing software such as Microsoft Word or Google Docs. Students will learn essential formatting techniques, document editing, and how to create professional-looking documents.
3. Spreadsheet Applications: Students will explore spreadsheet software like Microsoft Excel or Google Sheets. They will learn how to input data, perform calculations, create charts and graphs, and utilize basic functions and formulas.
4. Presentation Software: Practical sessions will involve using presentation software such as Microsoft PowerPoint or Google Slides. Students will learn how to create visually appealing presentations, add multimedia elements, and deliver effective presentations.
5. Internet Skills: Students will develop internet skills such as web browsing, searching for information, and evaluating online sources. They will also learn about internet safety, privacy, and best practices for online communication.
6. Email Management: Practical exercises will focus on email management using platforms like Gmail or Microsoft Outlook. Students will learn how to compose, send, receive, and organize emails efficiently.
7. Basic Troubleshooting: Students will be introduced to basic troubleshooting techniques for common computer issues. They will learn how to identify and resolve hardware and software problems effectively.

Throughout the laboratory sessions, emphasis will be placed on hands-on learning, problem-solving, and collaboration. By actively engaging in practical exercises, students will develop essential computer skills that are valuable in both academic and professional settings.

Name of Course	Computer Workshop
Number of Credits	1
Number of Hours	16
Cross section	Bachelor's Degree
Needs	None
Prerequisites	Fundamentals of Computer Programming

Course Objectives:

1. Understanding the principles of workshop safety and health, including the proper use of tools and equipment.
2. Introduction to basic practical concepts in computer work.
3. Introduction to basic practical concepts in computer hardware.
4. Introduction to basic practical concepts in computer software.
5. Understanding basic concepts of information security in computer systems.
6. Understanding basic concepts of software, hardware, and e-commerce system security.
7. Learning macro writing for software management and computer systems.
8. Connecting, using, and programming various software in a smart work environment, including wireless communication protocols such as Bluetooth, Zigbee, and LAN.
9. Practical experience with connections, network cards, and microcontrollers in a network environment.
10. Writing simple web applications and designing web pages.

3.8. Physics Laboratory II (Electricity and Magnetism)

In the Physics II course, students will engage in a series of practical experiments aimed at reinforcing theoretical concepts and developing hands-on skills. These experiments cover various topics within the realm of physics, including:

Course Objectives:

1. Electric Charge and Coulomb's Law: Students will perform experiments to understand electric charge and Coulomb's law. They will measure electric charges, study the forces between charged objects, and verify the inverse square law of electrostatic forces.
2. Electric Fields: Practical exercises will focus on electric fields, including mapping electric field lines, determining the electric field strength due to point charges, and investigating the behavior of charged particles in electric fields.
3. Gauss's Law: Students will explore Gauss's law through experimental setups. They will measure electric flux, verify Gauss's law for different charge distributions, and apply Gauss's law to calculate electric fields in various scenarios.
4. Electric Potential: Experiments related to electric potential will involve measuring potential differences, calculating electric potential energy, and understanding the relationship between electric potential and electric field.
5. Capacitance and Capacitors: Practical sessions will cover capacitance and capacitors. Students will measure capacitance, investigate factors affecting capacitance, and explore the behavior of capacitors in electric circuits.
6. Direct Current Circuits: Students will work with direct current (DC) circuits, analyzing the behavior of resistors, capacitors, and inductors in DC circuits. They will apply Ohm's law, Kirchhoff's laws, and principles of energy conservation in circuit analysis.
7. Magnetic Fields: Practical experiments will focus on magnetic fields, including measuring magnetic field strength, studying the behavior of magnets and magnetic materials, and investigating the magnetic field around current-carrying conductors.
8. Faraday's Law of Induction: Students will explore Faraday's law through experimental setups involving electromagnetic induction. They will observe induced currents, measure induced voltages, and investigate factors affecting electromagnetic induction.
9. Alternating Current Circuits: Experiments in alternating current (AC) circuits will involve analyzing the behavior of resistors, capacitors, and inductors in AC circuits. Students will study resonance phenomena, power in AC circuits, and the effects of frequency and impedance.
10. Electromagnetic Waves: Practical exercises will explore electromagnetic waves, including measuring wave properties, studying wave interference and diffraction, and investigating the properties of electromagnetic radiation.

These practical experiments provide students with hands-on experience and reinforce key concepts covered in the Physics II course, helping them develop critical thinking skills and a deeper understanding of physics principles.

Name of Course	Physics Laboratory II
Number of Credits	1
Number of Hours	16
Cross section	Bachelor's Degree
Needs	None
Prerequisites	Physics II
Reference books	[1] D. Halliday, R. Resnick, and J. Walker, Fundamentals of Physics. 9th edition, Wiley,2010.

Course Objectives:

1. Understanding and applying Ohm's and Kirchhoff's laws in direct current circuits. Introduction to the use of meters (ammeters, voltmeters, and ohmmeters).
2. Exploring direct current measurement bridges for accurate measurements.
3. Investigating the charging and discharging of capacitors in RC circuits with step input.
4. Analyzing the steady-state response of RC circuits to sinusoidal input.
5. Analyzing the steady-state response of RL circuits to sinusoidal input.
6. Exploring magnetism and electric current, including the study of Lenz's and Faraday's laws. Introduction to oscilloscope usage.
7. Understanding the principles underlying the operation of transformers.
8. Investigating the effect of ferromagnetic materials on magnetic fields.
9. Hands-on experience with DC generators and dynamos.

4

General Courses

4.1. Islamic Ideology I
Focused on Islamic Laws and Regulations.

4.2. Islamic Ideology II
Based on Rational and Philosophical proofs of God and other Islamic Concepts.

4.3. Islamic Revolution of Iran
Studies focused on Islamic Revolution happened in 1979 and the upcoming events. Also, what leads to this revolution.

4.4. The History of Islamic Culture and Civilization
Focused on Iran History and Islamic Civilization in Arab world.

4.5. Thematic Interpretation of Qu'ran
Interpretation of Qu’ran.

4.6. General Persian
Mainly focused on Persian literature, academic, and formal writing.

4.7. General English Language
Focused on four English Skills: Writing, Speaking, Listening and Reading.

4.8. Physical Education
General Excreting and Aerobic and Core Muscle Exercises.

4.9. Sports 1
Specialized in one of the following sports: Football, Volleyball, Basketball, Marathon Running, Table Tennis or Badminton.

4.10. Family and Population Knowledge

Mainly Focused on Marriage and Sexual Education.