COURSE SYLLABUS

CIVIL ENGINEERING FUNDAMENTALS

International Program

CIVIL ENGINEERING FUNDAMENTALS

5 ECTS

COURSE DESCRIPTION PROGRAMMING FOR CIVIL ENGINEERING FUNDAMENTALS

This course provides a comprehensive introduction to civil engineering, covering essential principles and practices related to the design, construction, and maintenance of infrastructure. Students will explore various sub-disciplines of civil engineering, including structural engineering, geotechnical engineering, transportation engineering, environmental engineering, and water resources engineering. The course emphasizes the application of engineering principles to solve real-world challenges in urban development, transportation networks, and environmental sustainability. By the end of the course, students will have a solid foundation in civil engineering concepts and the ability to apply these concepts through practical design projects.

COURSE OBJECTIVES

Upon successfully completing the course, students will be able to:

1. Civil Engineering Disciplines: Understand the various sub-disciplines within civil engineering and their roles in infrastructure development.
2. Structural Analysis: Analyze and design simple structural elements, including beams, columns, and trusses.
3. Material Properties: Understand the properties and behavior of construction materials, including concrete, steel, and soil.
4. Geotechnical Engineering Principles: Apply geotechnical principles in the analysis and design of foundations, retaining walls, and earth structures.
5. Transportation Engineering: Design basic transportation systems, including roads, highways, and traffic control devices.
6. Environmental and Water Resources Engineering: Address environmental challenges through sustainable design practices and manage water resources effectively.
7. Construction Management: Understand the fundamentals of project management, including planning, scheduling, and cost estimation.
8. Sustainability in Civil Engineering: Incorporate sustainable practices in civil engineering design and construction.
9. Civil Engineering Software: Utilize civil engineering software for design and analysis, including AutoCAD, SAP2000, and EPANET.
10. Capstone Project: Apply the knowledge and skills gained in the course to a real-world civil engineering design project.

STUDENT LEARNING OUTCOMES

Upon successful completion of this course, students will be able to:

1. Design and Analyze Civil Infrastructure: Design and analyze basic civil infrastructure, including buildings, bridges, roads, and water systems.

2. Apply Engineering Mechanics: Apply principles of engineering mechanics to solve problems related to structural stability and material strength.
3. Geotechnical Investigations: Conduct geotechnical investigations and assess soil properties for foundation design.
4. Transportation System Design: Develop efficient transportation systems considering safety, traffic flow, and environmental impact.
5. Water Resources Management: Design water distribution and treatment systems, including stormwater management and flood control.
6. Environmental Impact Assessment: Conduct environmental impact assessments and propose mitigation measures for civil engineering projects.
7. Construction Planning and Management: Develop project schedules, resource allocations, and cost estimates for construction projects.
8. Utilize Civil Engineering Software: Demonstrate proficiency in using software tools for civil engineering design, analysis, and project management.
9. Ethical Practices: Adhere to ethical standards in civil engineering, with an emphasis on public safety, sustainability, and professional responsibility.
10. Collaborative Project Work: Work effectively in teams to complete a comprehensive civil engineering design project.

LECTURE TOPICS

Week	Topics of Study
1	Introduction to Civil Engineering and Its Disciplines
2	Engineering Mechanics: Statics and Dynamics
3	Material Properties and Testing
4	Structural Analysis and Design Basics
5	Introduction to Geotechnical Engineering
6	Soil Mechanics and Foundation Design
7	Transportation Engineering: Roadway Design and Traffic Analysis
8	Midterm Examination
9	Environmental Engineering: Water and Wastewater Treatment
10	Water Resources Engineering: Hydrology and Hydraulic Design
11	Construction Planning and Management
12	Sustainable Practices in Civil Engineering
13	Civil Engineering Software Tools: AutoCAD, SAP2000, EPANET
14	Capstone Project Development
15	Final Review and Project Presentations
16	Final Test / Project: Python Script Demonstrating Comprehensive Understanding

TEACHING METHODS

The course will be delivered through a blend of lectures, hands-on lab sessions, group projects, and case studies. Core theoretical concepts and principles will be introduced during lectures, while lab sessions will focus on the practical application of these concepts in material testing, structural analysis, and software tools. Students will work on team-based civil engineering design projects in group projects, and real-world examples will be explored through case studies to illustrate civil engineering challenges. Field trips to ongoing civil engineering projects will provide opportunities to observe construction practices and challenges firsthand. Additionally, guest lectures from industry professionals will offer insights into current civil engineering practices. The course will also incorporate problem-based learning exercises to enhance critical thinking related to infrastructure design, along with simulations using software tools to model civil engineering systems and projects.

ASSESSMENT AND GRADING:

- Class Participation and Engagement: 10%
- Homework Assignments and Quizzes: 20%
- Midterm Examination: 20%
- Capstone Project: 30%
- Project Reports and Presentations: 20%

TEXTBOOK:

"Civil Engineering: A Modern Approach" by Lester A. Hoel, Nicholas J. Garber

REFERENCES:

1. "Structural Analysis" by R.C. Hibbeler
2. "Principles of Geotechnical Engineering" by Braja M. Das
3. "Traffic and Highway Engineering" by Nicholas J. Garber, Lester A. Hoel
4. "Environmental Engineering" by Howard S. Peavy, Donald R. Rowe, George Tchobanoglous

ADDITIONAL RESOURCES:

Supplementary readings, design software tutorials, and online resources will be provided to support learning.

GRADING SCALE:

- A: 90-100%
- B: 80-89%
- C: 70-79%
- D: 60-69%
- F: Below 60%

COURSE POLICIES:

- Regular attendance and active participation are expected.
- Late submissions may incur grade penalties unless prior arrangements are made.
- Academic integrity is strictly enforced.

Undiknas University reserves the right to changes. The syllabus is introductory, and the latest version will be distributed in the beginning of the lectures. Holiday schedules may vary depending on the public holidays.

COURSE SYLLABUS

COMPUTER AND COMMUNICATION SYSTEMS ENGINEERING

International Program

5 ECTS

COURSE DESCRIPTION FOR COMPUTER AND COMMUNICATION SYSTEMS ENGINEERING

This course covers the fundamental concepts and techniques in the design, implementation, and analysis of computer and communication systems. It integrates key aspects of computer networks, digital communication systems, wireless technologies, and information theory. The course focuses on the interaction between hardware and software components in communication systems, as well as the protocols and algorithms that enable efficient data exchange. Students will explore modern communication technologies, network architectures, security measures, and practical applications of computer and communication systems in the real world.

COURSE OBJECTIVES

Upon successful completion of this course, students will be able to:

1. Understand the fundamental principles of computer and communication systems engineering.
2. Analyze and design digital communication systems and protocols.
3. Understand computer networking concepts and protocols, including LANs, WANs, and wireless networks.
4. Design and implement secure and efficient communication systems.
5. Apply knowledge of modulation, coding, and information theory to improve communication system performance.
6. Evaluate the performance of communication systems using simulation and analysis tools.
7. Understand the role of hardware and software in modern communication systems.
8. Solve real-world engineering problems related to communication networks and computer systems.

STUDENT LEARNING OUTCOMES

Upon successful completion of this course, students will be able to:

1. Digital Communication Systems: Design and analyze digital communication systems, including modulation, multiplexing, and encoding techniques.
2. Computer Networks: Understand the architecture and operation of computer networks, including network protocols, routing, and switching.
3. Wireless Communication: Design and evaluate wireless communication systems, including cellular networks, Wi-Fi, and satellite communication.
4. Network Security: Implement security protocols and mechanisms to safeguard communication systems and networks from attacks and vulnerabilities.
5. Network Simulation and Analysis: Use simulation tools to model and evaluate the performance of

communication networks and systems.

6. Information Theory: Apply principles of information theory to optimize communication systems for data transmission efficiency.
7. Communication Hardware: Understand the role of communication hardware, including antennas, modems, and routers, in data transmission.
8. System Integration: Design integrated systems that combine computer and communication technologies for specific engineering applications.

LECTURE TOPICS

Week	Topics of Study
1	Introduction to Computer and Communication Systems Engineering
2	Communication Channels and Signal Transmission
3	Digital Communication Systems: Modulation and Demodulation
4	Information Theory and Data Compression
5	Error Detection and Correction Techniques
6	Network Architectures and Protocols: OSI and TCP/IP Models
7	Midterm Exam: Digital Communication Systems and Network Protocols
8	Wireless Communication Systems: Design and Performance
9	Computer Networking: LANs, WANs, and Wireless Networks
10	Routing, Switching, and Network Management
11	Network Security: Protocols, Cryptography, and Firewalls
12	Internet of Things (IoT) and Machine-to-Machine Communication
13	Case Studies: Modern Communication Technologies (5G, Optical Networks, Satellite Communication)
14	Simulation and Modeling of Communication Networks
15	Final Project Design and Implementation
16	Final Exam and Project Presentations

TEACHING METHODS

The teaching methods for this course include lectures that provide theoretical knowledge of communication systems and computer networks, complemented by hands-on laboratory sessions where students design and analyze network systems. Team-based projects focus on implementing real-world communication solutions, while case studies explore cutting-edge technologies such as

5G, IoT, and satellite systems. Guest lectures from industry experts offer insights into current trends and challenges in the field.

ASSESSMENT AND GRADING:

- Class Participation and Engagement: 10%
- Lab Assignments: 20%
- Midterm Exam: 20%
- Group Project (Communication System Design): 30%
- Quizzes: 10%
- Final Exam and Project Presentation: 10%

TEXTBOOK:

- "Computer Networks" by Andrew S. Tanenbaum (Latest Edition)

REFERENCES:

- "Digital and Analog Communication Systems" by Leon W. Couch
- "Data and Computer Communications" by William Stallings
- "Wireless Communications: Principles and Practice" by Theodore S. Rappaport
- "Communication Systems Engineering" by John G. Proakis and Masoud Salehi

ADDITIONAL RESOURCES:

- Online simulations and modeling tools for communication systems.
- Research papers and industry reports on emerging communication technologies.
- Technical documentation and standards for computer networks and protocols.

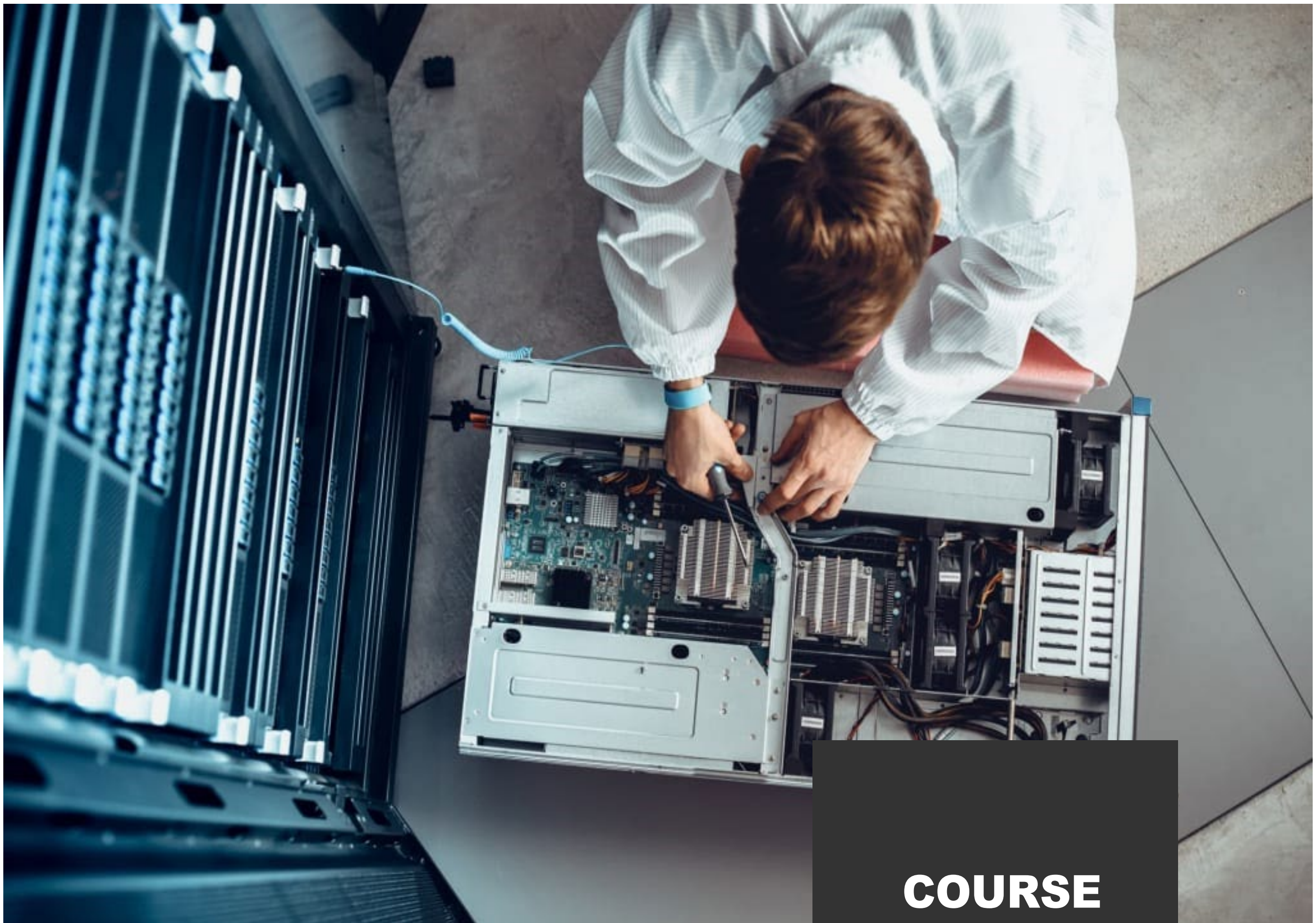
GRADING SCALE:

- A: 90-100%
- B: 80-89%
- C: 70-79%
- D: 60-69%
- F: Below 60%

COURSE POLICIES:

Attendance and Participation: Regular attendance and active participation are required.
Late Submissions: Late assignments will be penalized unless prior approval is obtained.
Academic Integrity: Any form of cheating or plagiarism will result in severe penalties.
Collaboration: Group work is encouraged for projects, but individual assessments must be completed independently.

Undiknas University reserves the right to changes. The syllabus is introductory, and the latest version will be distributed in the beginning of the lectures. Holiday schedules may vary depending on the public holidays.



COURSE SYLLABUS

COMPUTER ENGINEERING

International Program

5 ECTS

COURSE DESCRIPTION COMPUTER ENGINEERING

This course introduces students to the core concepts and methodologies that form the foundation of computer engineering. Covering both hardware and software aspects, the course emphasizes the integration of these components in designing and developing computer systems. Key topics include digital logic design, computer architecture, microprocessors, operating systems, and programming. Students will engage in both theoretical study and practical lab sessions to gain a comprehensive understanding of how computer systems are built, how they operate, and how they are programmed. By the end of the course, students will be equipped to understand the basic principles of computer engineering and apply them to real-world problems.

COURSE OBJECTIVES

Upon successfully completing the course, students will be able to:

1. Digital Logic Mastery: Understand and apply the principles of digital logic design, including Boolean algebra, logic gates, and combinational/sequential circuits.
2. Microprocessor Fundamentals: Gain proficiency in the architecture, operation, and programming of microprocessors.
3. Computer Architecture: Comprehend the structure and function of computer systems, including CPU, memory, I/O devices, and data buses.
4. Programming Proficiency: Develop skills in programming languages commonly used in computer engineering, such as C/C++ and assembly language.
5. Operating Systems Basics: Understand the role and functions of operating systems, including process management, memory management, and file systems.
6. Hardware-Software Integration: Explore the interaction between hardware and software, and the role of firmware in embedded systems.
7. Lab Skills Development: Acquire practical experience in digital circuit design, microprocessor programming, and system integration through lab exercises.
8. Problem-Solving in Computer Engineering: Apply computer engineering principles to design and troubleshoot simple computer systems.

STUDENT LEARNING OUTCOMES

Upon successful completion of this course, students will be able to:

1. Design Digital Circuits: Analyze and design basic digital circuits using combinational and sequential logic.
2. Program Microprocessors: Write and debug programs for microprocessors using assembly language and high-level programming languages.

3. **Understand Computer Organization**: Explain the organization and operation of different components within a computer system.
4. **Develop Embedded Systems**: Design and implement simple embedded systems by integrating hardware and software components.
5. **Use Simulation Tools**: Utilize simulation software to model and test digital circuits and computer systems.
6. **Collaborate on Projects**: Work effectively in teams to design, implement, and troubleshoot computer engineering projects.
7. **Document Engineering Processes**: Prepare detailed documentation for digital designs, microprocessor programs, and system configurations.
8. **Adhere to Ethical Practices**: Demonstrate awareness of ethical and professional responsibilities in computer engineering.

LECTURE TOPICS

Week	Topics of Study
1	Introduction to Problem Solving and Computer Programming
2	Understanding Flowcharting and Logical Constructs
3	Boolean Algebra and Logic Operations
4	Variables and Data Types
5	Decision Statements and Control Structures
6	Repetition and Looping Techniques
7	Working with Arrays/Lists
8	Middle test
9	Modules and Functions for Code Reusability
10	Introduction to Classes and Objects
11	Procedural vs. Object-Oriented Programming Paradigms
12	Introduction to IDLE Python Programming Environment
13	Hands-on Python Script Development
14	Debugging Techniques
15	Final Review
16	Final Test / Project: Python Script Demonstrating Comprehensive Understanding

TEACHING METHODS

The course will be delivered through a combination of lectures, hands-on lab sessions, group projects, case studies, and problem-based learning. Additional methods include flipped classrooms, simulation exercises, guest lectures from industry professionals, and online resources to enhance learning. Interactive technologies and peer teaching will also be incorporated to reinforce concepts and promote engagement.

ASSESSMENT AND GRADING:

- Class Participation and Engagement: 10%
- Homework Assignments and Quizzes: 20%
- Midterm Examination: 20%
- Final Project (Python Script): 30%
- Problem-Solving Experience Reports: 20%

TEXTBOOK:

- "Computer Organization and Design: The Hardware/Software Interface" by David A. Patterson and John L. Hennessy

REFERENCES:

1. "Digital Design and Computer Architecture" by David Harris and Sarah Harris
2. "Microprocessor Architecture, Programming, and Applications with the 8085" by Ramesh Gaonkar

ADDITIONAL RESOURCES:

- Supplementary readings, lab manuals, and online tutorials will be provided to support the course.

GRADING SCALE:

- A: 90-100%
- B: 80-89%
- C: 70-79%
- D: 60-69%
- F: Below 60%

COURSE POLICIES:

- Regular attendance and active participation are expected.
- Late submissions may incur grade penalties unless prior arrangements are made.
- Academic integrity is strictly enforced.

Undiknas University reserves the right to changes. The syllabus is introductory, and the latest version will be distributed in the beginning of the lectures. Holiday schedules may vary depending on the public holidays.

COURSE SYLLABUS

COMPUTER NETWORKS AND SYSTEMS

International Program

COMPUTER NETWORKS AND SYSTEMS

5 ECTS

COURSE DESCRIPTION FOR COMPUTER NETWORKS AND SYSTEMS

This course provides a comprehensive introduction to the concepts, protocols, and technologies that form the foundation of computer networks and systems. Students will learn about the architecture, design, and operational principles of networks, including the Internet and local area networks (LANs). Topics such as data communication, networking protocols (TCP/IP), routing, switching, wireless technologies, network security, and cloud computing will be explored. Practical labs will focus on network setup, configuration, and troubleshooting using industry-standard tools and simulators.

COURSE OBJECTIVES

Upon successful completion of the course, students will be able to:

1. Understand the basic concepts and architectures of computer networks and systems.
2. Explain the functions and protocols of the OSI and TCP/IP models.
3. Analyze and design small to medium-sized networks using routing and switching techniques.
4. Configure, troubleshoot, and manage network devices (routers, switches, firewalls) and protocols.
5. Explore network security concepts, including encryption, firewalls, and intrusion detection.
6. Understand wireless networking technologies and protocols.
7. Examine the impact of cloud computing, virtualization, and software-defined networking (SDN) on modern networks.
8. Evaluate the performance of network systems and optimize them for efficiency and scalability.

STUDENT LEARNING OUTCOMES

1. Upon successful completion of this course, students will be able to:
2. Network Models & Protocols: Understand the OSI and TCP/IP models and their associated protocols (e.g., HTTP, FTP, DNS, ARP).
3. Data Communication: Explain the fundamentals of data communication, including signal transmission, encoding, and error detection/correction methods.
4. Routing & Switching: Design and implement IP addressing schemes, static and dynamic routing protocols (RIP, OSPF), and VLANs.
5. Network Configuration: Configure network devices (routers, switches, firewalls) using command-line interfaces (CLI) and network simulators (e.g., Cisco Packet Tracer, GNS3).
6. Network Security: Implement security measures such as encryption, VPNs, firewalls, and intrusion prevention systems (IPS) to secure networks.
7. Wireless Networks: Analyze and implement wireless network protocols (Wi-Fi, Bluetooth, 5G), and

manage their security.

8. Cloud Networking: Explore cloud networking concepts, including Infrastructure as a Service (IaaS) and software-defined networking (SDN).
9. Network Troubleshooting: Diagnose and resolve network issues using tools like Wireshark, Ping, Traceroute, and NetFlow.
10. Performance Optimization: Analyze network performance metrics (latency, throughput, jitter) and optimize network infrastructure for scalability and efficiency.
11. Ethical & Legal Issues: Recognize ethical and legal issues related to data privacy, security, and network management.

LECTURE TOPICS

Week	Topics of Study
1	Introduction to Computer Networks and System Architecture
2	OSI and TCP/IP Models: Layers and Protocols
3	Data Communication: Transmission, Encoding, Error Detection
4	Local Area Networks (LAN) and Ethernet Technologies
5	IP Addressing and Subnetting
6	Routing Protocols: Static Routing, RIP, OSPF
7	Switching: VLANs, STP, and Trunking
8	Midterm Exam
9	Network Security: Firewalls, Encryption, VPNs, and IPS
10	Wireless Networks: Wi-Fi, Bluetooth, and Cellular Technologies
11	Introduction to Cloud Networking and Virtualization
12	Software-Defined Networking (SDN) and Network Function Virtualization (NFV)
13	Network Troubleshooting and Performance Optimization
14	Network Design and Case Studies
15	Final Review and Presentation of Network Projects
16	Final Exam / Project Submission

TEACHING METHODS

The course includes lectures that explain theoretical networking concepts and real-world applications, complemented by hands-on labs for practical experience in network setup, configuration, and management using tools such as routers, switches, firewalls, and simulators like Packet Tracer and Wireshark. Students will also engage in group projects that involve collaborative network design and troubleshooting, simulating real-world networking scenarios. Case studies will analyze existing networks and their solutions to business challenges, while guest lectures will provide insights from industry experts on modern networking trends and challenges. Problem-based learning will encourage students to tackle problem-solving scenarios, applying critical thinking and technical solutions, and online resources will offer additional support through e-learning platforms, virtual labs, and networking forums for further study.

ASSESSMENT AND GRADING:

- Class Participation and Engagement: 10%
- Lab Exercises and Homework Assignments: 25%
- Midterm Examination: 20%
- Final Project (Network Design & Implementation): 30%
- Quizzes and Case Studies: 15%

TEXTBOOK:

- "Computer Networking: A Top-Down Approach" by James F. Kurose and Keith W. Ross

REFERENCES:

1. "Data Communications and Networking" by Behrouz A. Forouzan
2. "Cisco CCNA Routing and Switching Study Guide" by Todd Lammle
3. "Network Security Essentials" by William Stallings

ADDITIONAL RESOURCES:

- Supplementary readings, online simulators, lab exercises, and interactive resources will be provided to support learning.
- Online platforms such as Cisco NetAcad and Wireshark for hands-on practice.

GRADING SCALE:

- A: 90-100%
- B: 80-89%
- C: 70-79%
- D: 60-69%
- F: Below 60%

COURSE POLICIES:

- Attendance and Participation: Regular attendance and active participation in discussions, labs, and group projects are expected.
- Late Submissions: Late assignments may incur penalties unless prior arrangements are made.
- Academic Integrity: Plagiarism or cheating will not be tolerated and will result in disciplinary action.
- Collaboration: While group work is encouraged for labs and projects, individual assignments and exams must be completed independently.

Undiknas University reserves the right to change. The syllabus is introductory, and the latest version will be distributed in the beginning of the lectures. Holiday schedules may vary depending on the public holidays.

COURSE SYLLABUS

DATA SCIENCE

International Program

DATA SCIENCE

5 ECTS

COURSE DESCRIPTION DATA SCIENCE

This course provides an in-depth exploration of data science, covering the theoretical foundations and practical applications of analyzing large datasets to gain insights and inform decision-making. Students will learn techniques in data collection, cleaning, processing, and visualization, as well as machine learning, statistical analysis, and predictive modeling. The course emphasizes the use of tools such as Python, R, SQL, and cloud-based technologies to work with data across multiple domains, including business, healthcare, and engineering.

COURSE OBJECTIVES

Upon successful completion of the course, students will be able to:

1. Understand the core concepts of data science and its role in solving real-world problems.
2. Apply statistical and machine learning methods to analyze and interpret data.
3. Utilize programming languages (Python, R) and tools (SQL, Pandas, NumPy, Scikit-learn) to manipulate and analyze large datasets.
4. Build and evaluate predictive models using machine learning techniques.
5. Design and implement data visualization techniques to communicate insights effectively.
6. Perform data cleaning and preprocessing for structured and unstructured data.
7. Implement big data solutions using cloud technologies (e.g., AWS, Google Cloud).
8. Understand ethical implications and challenges in data science, including privacy, security, and bias.
9. Work collaboratively on data-driven projects and effectively communicate results to stakeholders.

STUDENT LEARNING OUTCOMES

Upon successful completion of this course, students will be able to:

1. Data Collection & Cleaning: Collect and preprocess data from various sources, ensuring accuracy and completeness for analysis.
2. Statistical Analysis: Perform statistical analysis using Python or R to uncover patterns, correlations, and trends in data.
3. Machine Learning Models: Develop and train machine learning models such as regression, classification, clustering, and neural networks.
4. Data Visualization: Use tools like Matplotlib, Seaborn, and Tableau to visualize data insights clearly and inform decision-making.
5. Big Data Analytics: Apply big data technologies (Hadoop, Spark) to process and analyze massive datasets.

6. **Data Ethics & Privacy:** Identify and address ethical issues related to data use, including data privacy, fairness, and transparency.
7. **Predictive Analytics:** Develop predictive models for forecasting future trends and outcomes in various fields.
8. **Collaborative Data Projects:** Work effectively in teams to tackle complex data science problems, from data acquisition to presentation of results.
9. **Cloud Computing in Data Science:** Deploy data science projects using cloud-based platforms and frameworks for scalable solutions.

LECTURE TOPICS

Week	Topics of Study
1	Introduction to Data Science and its Applications
2	Data Collection and Cleaning: Working with Structured and Unstructured Data
3	Introduction to Python and R for Data Analysis
4	Exploratory Data Analysis and Data Visualization Techniques
5	Introduction to Statistical Inference and Hypothesis Testing
6	Machine Learning: Supervised Learning (Regression and Classification)
7	Machine Learning: Unsupervised Learning (Clustering, Dimensionality Reduction)
8	Midterm Exam
9	Predictive Modeling and Time Series Analysis
10	Introduction to Big Data: Hadoop, Spark, and NoSQL Databases
11	Data Science in the Cloud: AWS, Google Cloud, and Azure
12	Data Ethics: Bias, Fairness, and Privacy in Data Science
13	Hands-on Project: Building a Machine Learning Model
14	Deep Learning: Neural Networks and AI Applications
15	Final Review and Presentation of Data Science Projects
16	Final Exam / Project Submission

TEACHING METHODS

The teaching methods for this course include lectures that cover core concepts of data science and provide step-by-step walkthroughs of tools and techniques. Hands-on labs offer practical experience with Python, R, SQL, and data visualization tools, while group projects encourage collaboration to solve real-world data science problems. Case studies are used to analyze successful data-driven projects across various industries, and students have access to online resources such as interactive coding platforms, datasets, and forums to support their learning. Guest lectures by industry experts provide insights into the latest trends and innovations in data science, and problem-based learning engages students with data problems that require critical thinking and technical solutions.

ASSESSMENT AND GRADING:

- Class Participation and Engagement: 10%
- Homework Assignments and Quizzes: 20%
- Midterm Examination: 20%
- Final Project (Data Science Solution): 30%
- Lab Reports and Case Studies: 20%

TEXTBOOK:

- "Python for Data Analysis" by Wes McKinney

REFERENCES:

1. " Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" by Aurélien Géron
2. "R for Data Science" by Hadley Wickham and Garrett Grolemund
3. "Introduction to Statistical Learning" by Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani

ADDITIONAL RESOURCES:

- Supplementary readings, coding exercises, and interactive online resources (e.g., Kaggle, Jupyter notebooks) will be provided.

GRADING SCALE:

- A: 90-100%
- B: 80-89%
- C: 70-79%
- D: 60-69%
- F: Below 60%

COURSE POLICIES:

- Attendance and Participation: Regular attendance and active participation in discussions and labs are expected.
- Late Submissions: Assignments submitted late may be penalized unless prior arrangements are made.
- Academic Integrity: Plagiarism and other forms of academic dishonesty will result in disciplinary action.
- Collaboration: While collaboration is encouraged for projects, individual assignments and exams must be completed independently.

Undiknas University reserves the right to changes. The syllabus is introductory, and the latest version will be distributed in the beginning of the lectures. Holiday schedules may vary depending on the public holidays.

COURSE SYLLABUS

ENGINEERING PROJECT MANAGEMENT

International Program

ENGINEERING PROJECT MANAGEMENT

5 ECTS

COURSE DESCRIPTION ENGINEERING PROJECT MANAGEMENT

This course provides an in-depth understanding of the principles, methodologies, and tools required to manage engineering projects effectively. It covers the entire project lifecycle, from initiation and planning to execution, monitoring, control, and closure. Students will learn how to apply project management techniques to engineering projects, considering constraints such as time, cost, quality, risk, and resource allocation. The course also addresses the unique challenges associated with managing engineering teams and projects in a global and multidisciplinary environment.

COURSE OBJECTIVES

Upon successfully completing the course, students will be able to:

1. Project Planning and Scheduling: Develop comprehensive project plans and schedules using tools like Gantt charts and Critical Path Method (CPM).
2. Resource Management: Allocate and manage resources effectively, including personnel, materials, and finances.
3. Risk Management: Identify, analyze, and mitigate risks associated with engineering projects.
4. Quality Management: Implement quality assurance and control measures to ensure project deliverables meet required standards.
5. Project Execution and Monitoring: Oversee project execution, monitor progress, and make adjustments as needed to stay on track.
6. Stakeholder Management: Engage and manage relationships with stakeholders, including clients, team members, and other project participants.
7. Leadership and Team Management: Lead and motivate engineering teams to achieve project goals.
8. Project Documentation and Reporting: Prepare detailed project documentation and reports, including project charters, status reports, and post-project evaluations.
9. Ethical and Professional Standards: Adhere to ethical and professional standards in project management practices.

STUDENT LEARNING OUTCOMES

Upon successful completion of this course, students will be able to:

1. Apply Project Management Tools and Techniques: Utilize tools such as MS Project, Primavera, or other project management software to plan, execute, and monitor engineering projects.
2. Develop Project Management Plans: Create detailed project management plans, including scope, time, cost, and risk management plans.
3. Conduct Risk Assessments: Perform risk assessments and develop mitigation strategies for engineering projects.

4. **Manage Project Budgets:** Prepare and manage project budgets, including cost estimation, forecasting, and control.
5. **Implement Quality Control Measures:** Apply quality management principles to ensure project outputs meet technical specifications and customer expectations.
6. **Lead Engineering Teams:** Demonstrate leadership skills in managing and directing engineering teams to achieve project objectives.
7. **Communicate Effectively:** Produce clear and concise project documentation and communicate project status to stakeholders effectively.
8. **Evaluate Project Outcomes:** Assess the success of completed projects and identify lessons learned for future improvement.

LECTURE TOPICS

Week	Topics of Study
1	Introduction to Project Management in Engineering
2	Project Lifecycle and Phases
3	Project Scope and Objectives Definition
4	Work Breakdown Structure (WBS) and Task Allocation
5	Project Scheduling: Gantt Charts, CPM, and PERT
6	Resource Allocation and Budgeting
7	Risk Management and Contingency Planning
8	Midterm Exam
9	Quality Assurance and Control in Engineering Projects
10	Team Dynamics and Leadership in Engineering Projects
11	Stakeholder Management and Communication
12	Project Monitoring and Control Techniques
13	Project Documentation and Reporting
14	Ethical Issues in Engineering Project Management
15	Final Review
16	Final Exam / Project Presentation: Comprehensive Engineering Project Plan

TEACHING METHODS

The course include lectures to introduce and explain key concepts and methodologies, supplemented by case studies that analyze real-world engineering projects to apply theoretical knowledge. Students collaborate on group projects to gain hands-on experience with engineering project management plans, and engage in problem-based learning activities focused on project management scenarios. Insights from industry professionals are provided through guest speakers, while simulations replicate real-life project management challenges. The flipped classroom approach requires preparation through pre-class readings and videos, allowing for in-depth discussions during class.

ASSESSMENT AND GRADING:

- Class Participation and Engagement: 10%
- Homework Assignments and Quizzes: 20%
- Midterm Examination: 20%
- Final Project (Comprehensive Project Plan): 30%
- Case Study Reports: 20%

TEXTBOOK:

- "Project Management: A Systems Approach to Planning, Scheduling, and Controlling" by Harold Kerzner

REFERENCES:

1. " Engineering Project Management" by Nigel J. Smith
2. "A Guide to the Project Management Body of Knowledge (PMBOK Guide)" by Project Management Institute (PMI)

ADDITIONAL RESOURCES:

Supplementary readings, case studies, and online resources will be provided to support learning.

GRADING SCALE:

- A: 90-100%
- B: 80-89%
- C: 70-79%
- D: 60-69%
- F: Below 60%

COURSE POLICIES:

- Regular attendance and active participation are expected.
- Late submissions may incur grade penalties unless prior arrangements are made.
- Academic integrity is strictly enforced.

Undiknas University reserves the right to change. The syllabus is introductory, and the latest version will be distributed at the beginning of the lectures. Holiday schedules may vary depending on the public holidays.

COURSE SYLLABUS

INTRODUCTION TO ENVIRONMENTAL ENGINEERING

International Program

INTRODUCTION TO ENVIRONMENTAL ENGINEERING

5 ECTS

COURSE DESCRIPTION ENVIRONMENTAL ENGINEERING

This course provides an introduction to the principles and practices of environmental engineering. Students will explore the fundamental concepts related to water and air quality, waste management, pollution control, and environmental sustainability. The course covers the scientific, technical, and regulatory aspects of environmental engineering, focusing on how to design and implement systems to protect human health and the environment. Through lectures, case studies, laboratory experiments, and fieldwork, students will gain a comprehensive understanding of environmental challenges, and the engineering solutions used to address them.

COURSE OBJECTIVES

Upon successfully completing the course, students will be able to:

1. Environmental Engineering Principles: Understand the basic principles of environmental engineering, including water and air quality management, waste treatment, and pollution control.
2. Water and Air Quality Analysis: Analyze water and air quality using standard methods and interpret the results in the context of environmental regulations.
3. Sustainable Engineering Practices: Apply the concepts of sustainability and green engineering to environmental problem-solving.
4. Pollution Control Techniques: Design and evaluate pollution control systems for air, water, and soil.
5. Environmental Regulations: Understand and apply key environmental regulations and standards that govern engineering practices.
6. Waste Management: Develop strategies for the management and treatment of solid and hazardous waste.
7. Environmental Impact Assessment: Conduct basic environmental impact assessments (EIA) and develop mitigation strategies.
8. Field and Laboratory Techniques: Utilize laboratory and field techniques to monitor environmental parameters and assess the effectiveness of engineering interventions.
9. Problem-Solving Experience: Address real-world environmental challenges through engineering design and problem-solving approaches.

STUDENT LEARNING OUTCOMES

Upon successful completion of this course, students will be able to:

1. Assess Environmental Quality: Conduct and interpret water, air, and soil quality assessments using appropriate analytical methods.

2. **Design Environmental Systems**: Design systems for water treatment, air pollution control, and waste management.
3. **Apply Sustainable Practices**: Implement sustainable practices in environmental engineering projects.
4. **Understand Environmental Legislation**: Navigate environmental laws and regulations relevant to engineering projects.
5. **Perform Environmental Impact Assessments**: Conduct EIAs and propose mitigation measures for development projects.
6. **Collaborate on Environmental Solutions**: Work effectively in teams to solve complex environmental engineering problems.
7. **Document and Present Findings**: Prepare detailed reports and presentations on environmental engineering projects.
8. **Conduct Fieldwork**: Apply fieldwork techniques to gather data and monitor environmental conditions.
9. **Promote Environmental Stewardship**: Demonstrate a commitment to environmental stewardship and ethical engineering practices.

LECTURE TOPICS

Week	Topics of Study
1	Introduction to Environmental Engineering and Sustainability
2	Water Quality: Parameters, Standards, and Testing Methods
3	Wastewater Treatment: Processes and Design
4	Air Quality: Pollutants, Monitoring, and Control Techniques
5	Solid Waste Management: Collection, Treatment, and Disposal
6	Hazardous Waste Management and Remediation Technologies
7	Environmental Chemistry and Microbiology
8	Midterm Test
9	Environmental Impact Assessment (EIA) and Risk Analysis
10	Sustainable Engineering and Green Technologies
11	Renewable Energy Systems and Environmental Impacts
12	Climate Change: Mitigation and Adaptation Strategies
13	Field Techniques in Environmental Monitoring
14	Case Studies in Environmental Engineering
15	Final Review and Project Presentations
16	Final Test / Project Presentation

TEACHING METHODS

The course will be delivered through lectures, laboratory sessions, field trips, and case studies. Students will engage in problem-based learning activities, group projects, and interactive discussions. Guest speakers from industry and government agencies will provide insights into real-world environmental engineering challenges. Online resources and simulation tools will also be used to enhance learning.

ASSESSMENT AND GRADING:

- Class Participation and Engagement: 10%
- Homework Assignments and Quizzes: 20%
- Lab Reports and Fieldwork: 20%
- Midterm Examination: 20%
- Final Project and Presentation: 30%

TEXTBOOK:

- "Introduction to Environmental Engineering" by Mackenzie L. Davis and David A. Cornwell

REFERENCES:

1. "Environmental Engineering: Fundamentals, Sustainability, Design" by James R. Mihelcic and Julie B. Zimmerman
2. "Principles of Environmental Engineering & Science" by Mackenzie L. Davis and Susan J. Masten

ADDITIONAL RESOURCES:

- Supplementary readings, environmental datasets, and online simulation tools will be provided to support learning.

GRADING SCALE:

- A: 90-100%
- B: 80-89%
- C: 70-79%
- D: 60-69%
- F: Below 60%

COURSE POLICIES:

- Regular attendance and active participation are expected.
- Late submissions may incur grade penalties unless prior arrangements are made.
- Academic integrity is strictly enforced.

Undiknas University reserves the right to change. The syllabus is introductory, and the latest version will be distributed in the beginning of the lectures. Holiday schedules may vary depending on the public holidays.

COURSE SYLLABUS

RENEWABLE ENERGY ENGINEERING

International Program

PROGRAMMING FOR RENEWABLE ENERGY ENGINEERING

5 ECTS

COURSE DESCRIPTION PROGRAMMING FOR RENEWABLE ENERGY ENGINEERING

The Renewable Energy Engineering course explores the principles, technologies, and applications of renewable energy sources. It covers the technical aspects of various renewable energy systems, including solar, wind, hydro, geothermal, and bioenergy. Students will learn about energy conversion technologies, system design, and integration into existing energy grids. The course emphasizes the importance of sustainability, environmental impact, and the economic aspects of renewable energy projects. Hands-on labs and projects provide practical experience in designing and implementing renewable energy systems.

COURSE OBJECTIVES

Upon successful completion of this course, students will be able to:

1. Understand the fundamental concepts and technologies of renewable energy sources.
2. Analyze and compare different renewable energy technologies based on efficiency, cost, and environmental impact.
3. Design and evaluate renewable energy systems and their integration into power grids.
4. Develop technical skills in the use of renewable energy simulation and modeling tools.
5. Assess the economic and environmental benefits of renewable energy projects.
6. Understand the policy, regulatory, and societal factors influencing the adoption of renewable energy.
7. Design and implement small-scale renewable energy projects and systems.

STUDENT LEARNING OUTCOMES

Upon successful completion of this course, students will be able to:

1. Renewable Energy Technologies: Describe the operating principles and applications of major renewable energy technologies, including solar photovoltaic, wind turbines, hydroelectric systems, geothermal energy, and biomass.
2. System Design: Design renewable energy systems, including the selection of appropriate technologies and components for specific applications.
3. Energy Conversion: Understand the processes of energy conversion, storage, and transmission in renewable energy systems.
4. Simulation and Modeling: Use simulation tools to model the performance of renewable energy systems and analyze their efficiency and output.
5. Economic Analysis: Evaluate the economic feasibility of renewable energy projects, including cost-benefit analysis and financial modeling.
6. Environmental Impact: Assess the environmental impact of renewable energy technologies and their role in reducing carbon emissions and promoting sustainability.
7. Policy and Regulation: Understand the policy frameworks, regulations, and incentives related to renewable energy development.
8. Project Implementation: Plan, design, and implement small-scale renewable energy projects, including system integration and performance assessment.

LECTURE TOPICS

Week	Topics of Study
1	Introduction to Renewable Energy and Sustainability
2	Solar Energy: Photovoltaic Systems and Thermal Applications
3	Wind Energy: Turbines, Design, and Efficiency
4	Hydroelectric Power: Types, Design, and Environmental Impact
5	Geothermal Energy: Systems and Applications
6	Biomass and Bioenergy: Technologies and Conversion Processes
7	Midterm Exam: Renewable Energy Technologies and System Design
8	Energy Storage: Batteries, Flywheels, and Pumped Hydro Storage
9	Integration of Renewable Energy into Power Grids
10	Economic and Financial Aspects of Renewable Energy Projects
11	Environmental and Social Impacts of Renewable Energy
12	Policy, Regulation, and Incentives for Renewable Energy
13	Simulation and Modeling of Renewable Energy Systems
14	Case Studies: Successful Renewable Energy Projects
15	Final Project Design and Implementation
16	Final Exam and Project Presentations

TEACHING METHODS

The course employs a variety of teaching methods to ensure a comprehensive understanding of renewable energy engineering. Lectures provide the foundational knowledge of renewable energy technologies and their principles. Hands-on labs offer practical experience with renewable energy systems, allowing students to design, test, and analyze these systems. Project work involves designing and implementing small-scale renewable energy systems, fostering practical skills and team collaboration. Case studies are used to examine real-world renewable energy projects, helping students understand practical applications and challenges. Guest lectures from industry experts provide insights into current trends and innovations in renewable energy. Simulation tools are utilized to model and analyze renewable energy systems, enhancing students' ability to evaluate system performance. Field trips to renewable energy facilities offer practical exposure to operational systems and technologies.

ASSESSMENT AND GRADING:

- Class Participation and Engagement: 10%
- Lab Assignments: 20%
- Midterm Exam: 20%
- Group Project (Renewable Energy System Design): 30%
- Quizzes: 10%
- Final Project Presentation: 10%

TEXTBOOK:

- "Renewable Energy: Power for a Sustainable Future" by Godfrey Boyle (Latest Edition)

REFERENCES:

- "Fundamentals of Renewable Energy Processes" by Aldo V. da Rosa
- "Introduction to Renewable Energy" by Vaughn C. Nelson and Kenneth L. Starcher
- "Renewable Energy Systems: A Smart Energy Systems Approach to the Future" by Henrik Lund
- "The Physics of Solar Energy" by C. A. Balanis

ADDITIONAL RESOURCES:

- Simulation software for renewable energy system modeling.
- Online databases and resources for renewable energy research.
- Technical reports and white papers on current trends and technologies in renewable energy.

GRADING SCALE:

- A: 90-100%
- B: 80-89%
- C: 70-79%
- D: 60-69%
- F: Below 60%

COURSE POLICIES:

- Attendance and Participation: Regular attendance and active participation are required.
- Late Submissions: Late assignments will be penalized unless prior arrangements are made.
- Academic Integrity: Any form of cheating or plagiarism will result in disciplinary actions.
- Collaboration: Teamwork is encouraged for projects, but individual assignments must be completed independently.

Undiknas University reserves the right to changes. The syllabus is introductory, and the latest version will be distributed in the beginning of the lectures. Holiday schedules may vary depending on the public holidays.

COURSE SYLLABUS

SOFTWARE ENGINEERING

International Program

SOFTWARE ENGINEERING

5 ECTS

COURSE DESCRIPTION FOR SOFTWARE ENGINEERING

The Software Engineering course introduces students to the principles, methods, and tools used in the design, development, testing, and maintenance of software systems. The course covers the software development lifecycle (SDLC), project management, requirements engineering, software design patterns, testing methodologies, and best practices for software documentation. Emphasis is placed on the agile development process and collaborative team projects that simulate real-world software engineering environments. Students will gain practical experience using tools for version control, bug tracking, and continuous integration.

COURSE OBJECTIVES

Upon successful completion of this course, students will be able to:

1. Understand the fundamental concepts and methodologies of software engineering.
2. Analyze and gather software requirements from stakeholders.
3. Design software solutions using architectural patterns, UML diagrams, and design principles.
4. Develop software using modern programming languages and version control tools.
5. Apply testing techniques, including unit testing, integration testing, and system testing.
6. Manage software projects using agile methodologies and track progress through sprints and releases.
7. Understand software maintenance, refactoring, and the importance of technical debt.
8. Collaborate effectively in teams to deliver software projects.
9. Document software designs, requirements, and processes according to industry standards.
10. Comprehend ethical and professional responsibilities in software engineering.

STUDENT LEARNING OUTCOMES

Upon successful completion of this course, students will be able to:

1. **Software Development Lifecycle (SDLC):** Understand and apply various stages of the SDLC, including planning, design, implementation, testing, deployment, and maintenance.
2. **Requirements Engineering:** Conduct stakeholder analysis, gather software requirements, and create requirement specification documents.
3. **Software Design:** Create software architectures using UML diagrams, design patterns (e.g., MVC, Singleton), and software design principles.
4. **Coding Practices:** Write clean, maintainable, and efficient code using modern software development frameworks.
5. **Agile Development:** Use agile principles, such as Scrum or Kanban, for project management and sprint planning.

6. **Version Control:** Implement version control systems, including Git, for collaborative software development.
7. **Software Testing:** Develop and execute tests (unit, integration, system) to ensure software reliability and quality.
8. **Team Collaboration:** Work effectively in a team to design, implement, and deliver a complete software project.
9. **Software Documentation:** Prepare technical documentation, including user manuals, system architecture, and developer guides.
10. **Ethical Software Engineering:** Demonstrate awareness of ethical considerations, such as data privacy, security, and intellectual property.

LECTURE TOPICS

Week	Topics of Study
1	Introduction to Software Engineering & SDLC
2	Requirements Engineering: Elicitation and Documentation
3	Software Design: UML, Design Patterns, and Architectures
4	Agile Methodologies: Scrum, Kanban, and Sprint Planning
5	Version Control Systems: Git and GitHub
6	Midterm Exam: Design and Requirements Analysis
7	Software Development Frameworks: Frontend and Backend Technologies
8	Software Testing: Unit, Integration, and System Testing
9	Software Quality Assurance and Continuous Integration
10	Software Maintenance and Refactoring Techniques
11	Collaborative Development in Agile Teams
12	Project Management: Tools and Best Practices
13	Ethics in Software Engineering: Data Privacy and Security
14	Case Studies: Successful Software Projects and Failures
15	Final Review and Software Project Presentations
16	Final Project Submission and Examination

TEACHING METHODS

Teaching methods for this course include lectures to explore theoretical aspects of software engineering and its real-world applications. Hands-on labs provide practical experience in software development, testing, and the use of collaborative tools. Students will engage in group projects to develop, document, and deliver working software applications. Case studies will involve analyzing real-world examples of successful and failed software projects. Guest lectures by industry professionals will offer insights into current trends in software engineering. Agile simulations will involve participating in sprints and collaborative workflows to mimic real-world development environments, while peer reviews will encourage students to conduct code reviews and project evaluations, fostering a collaborative learning environment.

ASSESSMENT AND GRADING:

- Class Participation and Engagement: 10%
- Lab Assignments: 20%
- Midterm Exam (Design and Requirements): 15%
- Group Project (Software Development): 30%
- Quizzes: 10%
- Final Project and Presentation: 15%

TEXTBOOK:

- "Software Engineering" by Ian Sommerville (Latest Edition)

REFERENCES:

1. "The Pragmatic Programmer" by Andrew Hunt and David Thomas
2. "Design Patterns: Elements of Reusable Object-Oriented Software" by Erich Gamma et al.
3. "Clean Code: A Handbook of Agile Software Craftsmanship" by Robert C. Martin
4. "Agile Software Development with Scrum" by Ken Schwaber and Mike Beedle

ADDITIONAL RESOURCES:

- GitHub repositories for collaborative development and version control.
- Online tutorials and coding exercises (e.g., LeetCode, HackerRank).
- Access to development platforms, such as Visual Studio Code, Git, and CI/CD tools.

GRADING SCALE:

- A: 90-100%
- B: 80-89%
- C: 70-79%
- D: 60-69%
- F: Below 60%

COURSE POLICIES:

- Attendance and Participation: Regular attendance and active participation in labs, discussions, and group projects are required.
- Late Submissions: Late assignments will incur a penalty unless previously arranged with the instructor.
- Academic Integrity: Plagiarism, cheating, and any form of academic dishonesty will result in disciplinary action.
- Collaboration: Collaboration is encouraged on group projects, but individual assignments must reflect each student’s work.

Undiknas University reserves the right to changes. The syllabus is introductory, and the latest version will be distributed in the beginning of the lectures. Holiday schedules may vary depending on the public holidays.