Pokhara University
Faculty of Science and Technology

Course No.: CMP 346 (3 Credits)	Full marks: 100
Course title: **Artificial Intelligence (3-1-3)**	Pass marks: 45
Nature of the course: Theory and Practical	Total Lectures: 45 hrs
Level: Bachelor	Program: BE (Computer)

1. Course Description

This course is designed to provide an in-depth introduction to the field of Artificial Intelligence (AI). It covers the fundamental concepts, methodologies, and applications of AI, including problem-solving by searching, knowledge representation, reasoning and machine learning. Students will also explore ethical issues.

2. General Objectives

- To provide the students with foundational principles and techniques of Artificial Intelligence.
- To develop the skills and ability in students to formulate AI-based solutions to real-world problems.
- To acquaint the students with the knowledge of various AI domains such as machine learning, expert systems, neural network and fuzzy systems.
- To provide the students with the knowledge to critically evaluate the ethical implications of AI and its impact on society.

3. Methods of Instruction

Lecture, Discussion, Readings, Practical works and Project works.

4. Contents in Detail

Specific Objectives	Contents

• Explain artificial intelligence, its approaches and its foundations. • Critically evaluate the ethical implications of AI and its impact on society.	**1. Introduction to Artificial Intelligence (4 hrs)** 1.1. Intelligence 1.1.1. Types of Intelligence 1.1.2. Components of Intelligence 1.2. Artificial Intelligence 1.2.1. Approaches of AI 1.2.1.1. Acting Humanly 1.2.1.2. Thinking Humanly 1.2.1.3. Thinking Rationally 1.2.1.4. Acting Rationally 1.2.2. Foundations of AI 1.2.3. History of AI 1.2.4. Risk and Benefits of AI 1.3. Ethics and Societal Implications 1.3.1. Ethical Implications of AI 1.3.2. AI and Society: Work and Automation, Employment, Privacy and Security 1.3.3. Governance and Regulation
● Design and implement intelligent agents.	**2. Intelligent Agents (5 hrs)** 2.1. Agents and Environments 2.2. Concept of Rationality 2.2.1. Performance Measures 2.2.2. Rationality and Rational Agent 2.3. Task environment and its properties 2.4. Structure of Agents 2.4.1. Agent programs 2.4.2. Types of agent programs 2.5. Learning Agents
● Formulate the real world problems and apply the search algorithms to solve them.	**3. Problem Solving and Search Algorithms (10 hrs)** 3.1. Problem Solving 3.1.1. Problem Solving Agents 3.1.2. Problem solving process 3.1.3. Production System 3.1.4. Well-defined and ill-defined problems 3.1.5. Problem formulation 3.2. Search Algorithms 3.2.1. Uninformed Search 3.2.1.1. Breadth- First Search 3.2.1.2. Depth-First Search 3.2.1.3. Iterative Deepening Search 3.2.2. Informed Search 3.2.2.1. Heuristics 3.2.2.2. Greedy Best-First Search 3.2.2.3. A* Search 3.3. Local Search and Optimization Problems 3.3.1. Hill-Climbing Search and its problems (Local maxima, plateaus, and ridges) 3.3.2. Simulated Annealing

	3.3.3. Genetic Algorithms 3.3.4. Gradient Descent 3.4. Adversarial Search and Game Playing 3.4.1. Minimax algorithm 3.4.2. Alpha-beta pruning 3.5. Constraint Satisfaction Problems 3.5.1. Representation of CSPs 3.5.1.1. Variables 3.5.1.2. Domains 3.5.1.3. Constraints 3.5.2. Search Algorithms for CSPs 3.5.2.1. Backtracking search 3.5.2.2. Constraint propagation 3.5.3. Optimization Technique: Min-Conflicts Heuristic1
● Represent the knowledge of a domain and apply inference rules to draw conclusions.	**4. Knowledge Representation and Reasoning (10 hrs)** 4.1. Propositional Logic 4.1.1. Syntax 4.1.2. Semantics 4.1.3. Inference in Propositional Logic 4.1.4. Conjunctive Normal Form (CNF) 4.1.5. Resolution Theorem Proving 4.1.6. Limitations of Propositional Logic 4.2. Predicate Logic 4.2.1. Syntax 4.2.2. Semantics 4.2.3. Inference in Predicate Logic 4.2.4. Resolution in Predicate Logic 4.3. Reasoning Under Uncertainty 4.3.1. Probabilistic Reasoning 4.3.1.1. Bayesian Networks 4.3.2. Probabilistic reasoning over time 4.3.2.1. Hidden Markov Models 4.4. Other Approaches to Knowledge Representation 4.4.1. Semantic Nets and Frames 4.4.2. Rule-based Representation 4.4.3. Ontological-Based Representation
● Develop and apply the machine learning algorithms to classify and cluster the data. ● Design an artificial neural network that can learn.	**5. Machine Learning (6 hrs)** 5.1. Definition and Evolution of Machine Learning 5.2. Learning by Analogy 5.3. Explanation-based learning 5.4. Supervised Learning Algorithms 5.4.1. Classification and Regression 5.4.2. Linear regression 5.4.3. K-Nearest Neighbour 5.5. Unsupervised Learning Algorithms 5.5.1. Clustering 5.5.2. K-means Clustering

	5.6. Artificial Neural Network 5.6.1. Biological Inspiration 5.6.2. Basic Components of ANN 5.6.3. Training Neural Networks 5.6.3.1. Forward Propagation 5.6.3.2. Loss Function 5.6.3.3. Backward Propagation 5.6.3.4. Learning Rate 5.6.4. Single-Layer Perceptron 5.6.5. Multi-Layer Perceptron
● Apply the fuzzy logic for reasoning in an expert system.	**6. Fuzzy Logic (5 hrs)** 6.1. Classical vs Fuzzy Logic 6.2. Fuzzy Sets and Membership Functions 6.3. Fuzzy Operations 6.4. Fuzzy Rule-Based Systems 6.5. Fuzzification and Defuzzification 6.6. Fuzzy Inference System: Mamdani
● Design and develop an expert system to solve a real-world problem.	**7. Expert System (5 hrs)** 7.1. Definition and History of Expert System 7.2. Architecture of Expert Systems 7.3. Knowledge Representation in expert system 7.3.1. Logic based representation 7.3.2. Rule-based system 7.3.3. Semantic networks 7.3.4. Ontology-based Systems 7.3.5. Frame-based Systems 7.4. Inference Mechanisms 7.4.1. Forward Chaining 7.4.2. Backward Chaining 7.5. Knowledge Acquisition and Learning 7.6. Applications of Expert Systems

5. Practical Works

Laboratory work of 45 hours per group of maximum 24 students should cover implementation of the following lab works:

SN	Implementation Description
1	Implement search algorithms (e.g., BFS, DFS, A*)
2	a.Create a program with facts about a different domain (e.g., a book collection, family tree) and perform queries to extract information.

	b.Extend the family tree with new rules to find relationships between grandparents, siblings, or cousins. Enhance the program with additional rules and variables to solve a given set of queries. Include comments explaining the logic. c.Develop a program that solves a specific problem related to a domain of interest (e.g., a recommendation system) and prepare a brief report explaining the design and logic. d.Implement Semantic Network in for Knowledge Representation and Querying (For example, a simple hierarchy might involve concepts such as "Animal", "Bird", and "Mammal" with relationships such as is_a, has_parts, or can_fly) e.Implement a Frame-Based Representation with Inheritance and Queries.
3	Develop a simple expert system using rule-based reasoning/fuzzy logic.
4	Implement and evaluate classification algorithms (e.g. linear regression and k-NN) and clustering algorithms (e.g. k-means)
5	Build and train single layer and multi-layer perceptron.
6	Implement the fuzzy logic for reasoning in an expert system.

Students should submit a project work that uses all the knowledge obtained from this course to solve any problem chosen by themselves. The marks for the practical evaluation must be based on the project work submitted by students.

6. List of Tutorials

The various tutorial activities that suit your course should cover all the content of the course to give students a space to engage more actively with the course content in the presence of the instructor. Students should submit tutorials as assignments or class works to the instructor for evaluation. The following tutorial activities of 15 hours per group of maximum 24 students should be conducted to cover the content of this course:

A. Discussion-based Tutorials: (3 hrs)
   a. Evolution of Artificial Intelligence (Class discussion).
   b. AI and Society: Employment, Privacy and Security (Class discussion).

B. Problem solving-based Tutorials: (6 hrs)
   a. Apply a search technique to solve a real world problem.
   b. Design and develop a simple expert system that solves a real world problem.

C. Review and Question/Answer-based Tutorials: (6 hrs)
   a. A detailed case study on any one fuzzy system. (Class Presentation)
   b. Students ask questions within the course content, assignments and review key course content in preparation for tests or exams.

7. Evaluation System and Students' Responsibilities

Evaluation System

The internal evaluation of a student may consist of assignments, attendance, internal assessment, lab reports, project works etc. The internal evaluation scheme for this course is as follows:

Internal Evaluation	**Weight**	**Marks**	**External Evaluation**	**Marks**
Theory		30	Semester-End examination	50
Attendance & Class Participation	10%			
Assignments	20%			
Presentations/Quizzes	10%			
Internal Assessment	60%			
Practical		20		
Attendance & Class Participation	10%			
Lab Report/Project Report	20%			
Practical Exam/Project Work	40%			
Viva	30%			
Total Internal		50		
Full Marks: 50 + 50 = 100				

Student Responsibilities

Each student must secure at least 45% marks separately in internal assessment and practical evaluation with 80% attendance in the class in order to appear in the Semester End Examination. Failing to get such a score will be given NOT QUALIFIED (NQ) to appear for the Semester-End Examinations. Students are advised to attend all the classes, formal exam, test, etc. and complete all the assignments within the specified time period. Students are required to complete all the requirements defined for the completion of the course.

8. Prescribed Books and References

Text Books

1. Russell, S. J., & Norvig, P. (2022). *Artificial intelligence: a modern approach.* Pearson.

References

1. Jurafsky, D., & Martin, J. H. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Pearson.
2. Bird S., Klein E., & Loper E. (2009). *Natural Language Processing with Python*. O'Reilly Media.

3. Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Pokhara University
Faculty of Science and Technology

Course No.: CMM 344 (3 Credits)	Full marks: 100
Course title: Digital Signal Analysis and Processing (3-1-2)	Pass marks: 45
Nature of the course: Theory and Practical	Total Lectures: 45 hrs
Level: Bachelor	Program: BE (CE)

1. Course Description

This course covers theory and methods for digital signal analysis processing including basic principles governing the analysis and design of discrete-time systems as signal processing devices. Major parts of the course will concentrate on signal analysis using Fourier transforms, linear system analysis and filter design. The discrete Fourier transform and its properties as well as the relationship between continuous and discrete time transforms will be studied. The course presents an analysis on how discrete time, linear shift invariant systems can be characterized using linear difference equations and the impulse response and show how tools such as the z-transform and discrete Fourier transform can be used in the design and analysis of such systems. An introduction to Fast Fourier Transform and the design and implementation of digital filters is presented towards the end of the course.

2. General Objectives

- To provide fundamental knowledge of digital signal processing techniques and applications.
- To introduce basic techniques in designing and implementing digital signal processing systems.
- To learn basic methods of spectral analysis.
- To teach students to design digital filters

3. Methods of Instruction

Lecture, Discussion, Readings, Practical works and Project works.

4. Contents in Detail

Specific Objectives	Contents

• Learn about the basics of discrete time signals and its classification • Understand the concept of different types of discrete time systems • Analyze the signals in time domain and frequency domain • Comprehend the concept of discrete linear convolution and its properties	**1. Discrete Signals and systems: (8 hrs)** 1.1. Discrete time signal, basic signal 1.2. Energy signal, power signal 1.3. Periodicity of discrete time signal 1.4. Transformation of independent variable 1.5. Discrete time Fourier Series and properties 1.6. Discrete time Fourier Transform and properties 1.7. Discrete time system properties 1.8. Linear Time Invariant (LTI) system, convolution sum, properties of LTI system 1.9. Frequency response of LTI system 1.10. Sampling of continuous time signal, spectral properties of sampled signal
• Comprehend the concept of Z-Transform, ROC and properties of Z-transform • Learn to perform forward and inverse Z-transform • Understand the condition for causality, non-causality and stability with respect to ROC	**2. Z-transform (6 hrs)** 2.1. Definition, convergence of Z-transform and region of convergence 2.2. Properties of Z-transform (linearity, time shift, multiplication by exponential sequence, differentiation, time reversal, convolution, multiplication) 2.3. Inverse z-transform by long division and partial fraction expansion.
• Grasp the concept of plotting Magnitude and Phase Response of LTI system • Understand the concept of poles and zeros and their relationship to causality	**3. Analysis of LTI system in frequency domain (5 hrs)** 3.1. Frequency response of LTI system, response to complex exponential 3.2. Linear constant co-efficient difference equation and corresponding system function 3.3. Relationship of frequency response to pole-zero of system 3.4. Linear phase of LTI system and its relationship to causality.
• Learn to represent FIR and IIR filters in direct, cascade and lattice structure. • Understand the effect of Quantization and limit cycles	**4. Discrete filter structures (6 hrs)** 4.1. FIR filter, Structures for FIR filter (direct form, cascade, lattice) 4.2. IIR filter, Structures for IIR filter (direct form I, direct form II, cascade, lattice, lattice ladder) 4.3. Quantization effect (truncation, rounding), limit cycles and scaling
• Grasp the concept of designing analog Butterworth and Chebyshev filters. • Comprehend how analog IIR filter is converted to digital IIR filter • Understand the idea of converting LPF prototype filter to LP, HP, BP and BS filters • Learn the property of Butterworth, Chebyshev and Elliptical filters and their differences.	**5. IIR Filter Design (7 hrs)** 5.1. Filter design by impulse invariance method 5.2. Filter design using bilinear transformation method 5.3. Design of digital low pass Butterworth filter 5.4. Properties of Chebyshev filter, properties of elliptic filter, properties of Bessel filter, spectral transformation

• Understand Gibbs phenomena • Grasp the concept of designing FIR filters using Windowing and frequency sampling methods. • Understand the concept of designing optimum FIR filters	**6. FIR Filter Design (7 hrs)** 6.1. Filter design by window method, commonly used windows (rectangular window, Hanning window, hamming window) 6.2. Filter design by Kaiser window 6.3. Filter design by frequency sampling method 6.4. Filter design using optimum approximation, Remez exchange algorithm
• Understand the concept of Discrete Fourier Transform and its properties • Grasp the concept of circular convolution • Learn about FFT, DIT, DIF algorithms. • Comprehend the computational complexity of FFT algorithm	**7. Discrete Fourier Transform (6 hrs)** 7.1. Discrete Fourier transform (DFT) representation, properties of DFT (linearity, time shift, frequency shift, conjugation and conjugate symmetry, duality, convolution, multiplication), circular convolution 7.2. Fast Fourier Transform (FFT) algorithm (decimation in time algorithm, decimation in frequency algorithm) 7.3. Computational complexity of FFT algorithm.

5.List of Tutorials

The following tutorial activities of 15 hours per group of maximum 24 students should be conducted to cover all the required contents of this course.

S.N.	Tutorials
1	Solve all the problems related to Section 1.
2	Solve the problems related to Z-Transform (Section 2)
3	Solve the problems related to Frequency Response of LTI system (Section 3)
4	Solve the problems related to Discrete Filter Structures (Section 4)
5	Solve the problems related to the design of digital IIR filters using Impulse Invariance and Bilinear Transformation methods
6	Solve the problems related to the design of digital FIR filters.
7	Solve the problems related to circular convolution, DIT and DIF

6.Practical Works

Laboratory work of 30 hours per group of maximum 24 students should cover implementation of the following topics using simulation software.

SN	List of Practicals
1	Generate and Investigate Basic Discrete Time Signals i. Unit impulse signal, Unit step signal, Ramp signal, Sinusoidal Signal, Exponential signal and square signal ii. Compute even and odd parts of signal iii. Convolution Sum of two sequences
2	Frequency response and pole zero plot of differential equation
3	Compute 4-point and 8-point DFT using FFT and investigate their frequency responses
4	Design an IIR lowpass filter using Impulse Invariance and Bilinear Transformation Method

5	Design a FIR filter using different windows and compare the result.
6	Real-World Digital Signal Analysis and Processing examples and demonstrations

7. Evaluation System and Students' Responsibilities

Evaluation System

The internal evaluation of a student may consist of assignments, attendances, internal assessment, lab reports, project works etc. The internal evaluation scheme for this course is as follows:

Internal Evaluation	**Weight**	**Marks**	**External Evaluation**	**Marks**
Theory		30	Semester-End examination	50
Attendances & Class Participations	10%			
Assignments	20%			
Presentations/Quizzes	10%			
Internal Assessment	60%			
Practical		20		
Attendances & Class Participations	10%			
Lab Report/Project Report	20%			
Practical Exam/Project Work	40%			
Viva	30%			
Total Internal		50		
Full Marks: 50 + 50 = 100				

Student Responsibilities

Each student must secure at least 45% marks in the internal evaluation with 80% attendance in the class to appear in the Semester End Examination. Failing to obtain such score will be given NOT QUALIFIED (NQ) and the student will not be eligible to appear in the End-Term examinations. Students are advised to attend all the classes and complete all the assignments within the specified time period. If a student does not attend the class(es), it is his/her sole responsibility to cover the topic(s) taught during the period. If a student fails to attend a formal exam, quiz, test, etc. there won't be any provision for a re-exam.

8. Prescribed Books and References

Text Books

1. J.G. Proakis and D.G. Manolakis, “Digital signal Processing”, Prentice Hall.

References

1. Alan V. Oppenheim, Ronald W. Schafer, John R. Buck, “Discrete-Time Signal Processing”, Pearson Education.
2. S. K. Mitra, “Digital signal Processing, A Computer-based Approach”, McGraw Hill

Pokhara University
Faculty of Science and Technology

Course Code.: ELX 320 (2 Credits)
Course title: **Embedded System (2-1-3)**
Nature of the course: Theory & Practice
Year, Semester:III, V
Level: Bachelor

Full marks: 100
Pass marks: 45
Time per period: 1 hour
Total periods: 45
Program: BE

1. Course Description

The course typically focuses on the design, implementation, and analysis of computer systems that are integrated into larger devices or systems to perform dedicated functions. These systems often operate in real-time environments and are optimized for specific tasks. The key topics of the course are: Microcontrollers, Programming Embedded Systems, Real-time Operating Systems (RTOS), Input/Output (I/O) Devices, Interrupts and Timers, Power Management, Optimization and Embedded Software Development Tools and IoT based embedded systems.

2. General Objectives

The general objectives of this course are:

- To equip students with the knowledge and skills to design, develop, and implement efficient hardware-software systems for specific, real-time applications
- To acquint students with microcontrollers, interface devices, manage real-time operations, and optimize system performance for various embedded applications like IoT, automotive, and robotics

3. Contents in Detail

Specific Objectives	Contents
• Define components, and list the characteristics of embedded systems • Differentiate between embedded systems and general-purpose computing systems • Learn the various applications and use cases of embedded systems (e.g., automotive, IoT, consumer electronics) • Gain knowledge about key components of Embedded System.	**Unit I: Introduction to Embedded Systems (3 hrs.)** 1.1 Definition and Overview of Embedded Systems 1.2 Embedded Systems vs. General-Purpose Systems 1.3 Applications and Domains (Automotive, Healthcare, Industrial Control, IoT, etc.) 1.4 Key Components: Microcontrollers, Sensors, Actuators, Peripherals

• Provide in-depth knowledge of programming languages commonly used in embedded systems, such as Embedded C. • Design, write, test, and debug software for embedded systems that meet specific functional requirements	**Unit II: Programming for Embedded Systems (5 hrs.)** 2.1 Overview of AVR Architecture 2.2 Embedded C Programming for Microcontroller: • Introduction to C for Embedded Systems. • Data Types, Control Structures, and Pointers • Memory Management • AVR Interrupt Handling • Input and output ports interfacing on AVR • Timers and Counters in AVR • Serial Communication in AVR
• Emphasize the development of both practical and theoretical understanding of real-time systems • Gain proficiency in utilizing a leading Real-Time Operating System (RTOS) such as VxWorks • Evaluate the advantages and disadvantages of various concepts, including task scheduling, context switching, task synchronization, resource sharing, deadlock, priority inversion, multithreading, and multitasking	Unit III: Real-Time Operating Systems (RTOS)(5 hrs.) 3.1 Concepts of Real-Time Systems 3.2 Introduction to RTOS (e.g., FreeRTOS, VxWorks) 3.3 Task Scheduling, Context Switching, Task Synchronization 3.4 Resource Sharing, Deadlock, and Priority Inversion 3.5 Multithreading and Multi-tasking in RTOS
• Gain insight about VHDL Programming, different modeling styles, data types, sub program and packages, test benches. • Learn to program in VHDL for combinational and sequential cicuits	Unit IV: Embedded System Design using VHDL (5 hrs.) 4.1 Introduction to VHDL 4.2 Different Modelling styles in VHDL for combinational and sequential circuits 4.3 Data types in VHDL 4.4 Sub program and Packages 4.5 VHDL realization for combinational and sequential circuits
• Understand the distinction between serial and Wireless communication and when to use each in embedded applications. • Explore the different types of communication (e.g., device-to-device, device-to-network, and device-to-cloud) that are	**Unit V: Communication Protocols (3 hrs.)** 5.1 Serial Communication: UART, SPI, I2C 5.2 Wireless Communication: Bluetooth, ZigBee, Wi-Fi, LoRa, GSM/GPRS 5.3 Networking: TCP/IP Basics in Embedded Systems

essential for embedded systems. • Understand Wireless Communication Protocols such as Bluetooth and BLE (Bluetooth Low Energy), Wi-Fi, ZigBee and LoRa	
• Learn the types of peripherals commonly used in embedded systems, including input devices (e.g., sensors, switches), output devices (e.g., displays, actuators), and communication devices (e.g., serial interfaces).	**Unit VI: Peripherals and Interfacing (4 hrs.)** 6.1 Sensor Interfacing:Analog and Digital Sensors (e.g., temperature, humidity, motion) 6.2 Actuator Interfacing: • Motor Control (DC, Stepper, Servo) • PWM for controlling brightness, speed, etc. 6.3 Display Interfacing: • LCD, and Seven Segment Displays
• Understand the various communication protocol used in IoT, such as MQTT • Understand common used hardware platforms in IoT. •	**Unit VII: Internet of Things (IoT) and Embedded Systems (3 hrs.)** 7.1 Introduction to IoT Concepts and Embedded System's Role in IoT. 7.2 IoT communication protocol: MQTT(Basic Concept) 7.3 Overview of common IoT hardware platforms: Arduino, ESP32, and Raspberry Pi. (Basic Introduction)

Note: The figures in the parentheses indicate the approximate periods for the respective units.

10 Methods of Instruction

Lecture, tutorials, lab works, projects.

5. List of Tutorials

The following tutorial activities of 15 hours per group of maximum 24 students should be conducted to cover all the required contents of this course.

S.N.	Tutorials
1	Beginner-Level Projects: Home Automation System, Smart Door Lock System, Temperature and Humidity Monitoring, Obstacle Avoidance Robot, Digital Thermometer
2	Intermediate-Level Projects: Weather Station, Smart Irrigation System, Bluetooth-Controlled Car, Home Security System, Pulse Oximeter
3	Advanced-Level Projects: IoT-Based Energy Meter, Drone Control System, Smart Traffic Light System, Heart Rate Monitor using ECG, Wearable Health Monitor

6. Practical Works

S.N.	Practical works
1	Introduction of VHDL for simulation of digital logic circuits and its components

2	Design and simulate basic logic gates (AND, OR, NOT, NAND, NOR, XOR, XNOR) using VHDL
3	Implement and test combinational circuits such as multiplexers, demultiplexers, encoders, and decoders
4	Design and simulate basic arithmetic circuits, including half adders, full adders, and subtractors
5	Create and analyze sequential circuits like flip-flops (D, JK, T), counters (binary, decade), and shift registers
6	Develop and simulate finite state machines, including both Mealy and Moore models, for various applications

7. **Evaluation system and Students' Responsibilities**

Evaluation System

In addition to the formal exam(s) conducted by the Office of the Controller of Examination of Pokhara University, the internal evaluation of a student may consist of class attendance, class participation, quizzes, assignments, presentations, written exams, etc. The tabular presentation of the evaluation system is as follows.

External Evaluation	**Marks**	**Internal Evaluation**	**Marks**
Semester-End Examination	50	Class attendance and participation	5
		Tutorials and projects works	5+5
		Quizzes/assignments and presentations	10
		Internal Term Exam	25
Total External	50	Total Internal	50
Full Marks 50+50 = 100			

Students' Responsibilities:

Each student must secure at least 45% marks in the internal evaluation with 80% attendance in the class to appear in the Semester End Examination. Failing to obtain such score will be given NOT QUALIFIED (NQ) and the student will not be eligible to appear in the End-Term examinations. Students are advised to attend all the classes and complete all the assignments within the specified time period. If a student does not attend the class(es), it is his/her sole responsibility to cover the topic(s) taught during the period. If a student fails to attend a formal exam, quiz, test, etc. there won't be any provision for a re-exam.

8. Prescribed Books and References

Text Book

1. Simon, D. E. (1999). *An embedded software primer* (Vol. 1). Addison-Wesley Professional.
2. Mazidi, M. A., Naimi, S., & Naimi, S. (2010). *AVR Microcontroller and Embedded Systems The* (p. 364). Pearson India.
3. Perry, D. L. (2002). *VHDL: programming by example* (Vol. 4). New York: McGraw-Hill.
4. Bahga, A., & Madisetti, V. (2014). *Internet of Things: A hands-on approach*. Vpt.

Reference Books

1. Peckol, J. (2019). *Embedded systems: A contemporary design tool* (2nd ed.). Wiley.
2. Vahid, F., & Givargis, T. (2010). *Embedded system design: A unified hardware/software introduction* (3rd ed.). Wiley.

Pokhara University
Faculty of Science and Technology

Course Code.: MGT 320 | Full marks: 100
Course title: Engineering Management (2-1-0) | Pass marks: 45
Nature of the course: Theory | Time per period: 1 hour
Total periods: 30
Level: Bachelor | Program: BE

1. Course Description

This course is designed to provide essential understanding of basic management knowledge and enhance their managerial capabilities integrating technology and contemporary issues. It includes introduction of management, planning, organizing, staffing, leading and motivation, controlling and recent trends in management specially tailored for software, computer and IT engineering. The course delivery will consist of lecture sessions, case studies and presentations on professional management practices.

2. General Objective

The course is designed to impart knowledge on management and emerging engineering management trends and develop skills of the students to enhance their managerial capabilities and enable them to apply in a technology-based organization.

3. Contents in Detail

Specific Objectives	Contents
To develop a foundational understanding of management, organization and engineering management	**Unit I: Introduction (6 Hrs.)** 1.1 Management 1.1.1 Functions of management 1.1.2 Level and scope of management 1.1.3 Principles of management 1.2 Organization 1.2.1 Characteristics of organization 1.2.2 Types of organization: formal and informal organizations, virtual organization 1.3 Engineering Management 1.3.1 Importance of management in technology-driven environments 1.3.2 Engineering functions in organizations: product development, operations, IT systems, quality assurance and others 1.3.3 Roles and responsibilities of an engineering manager
To familiarize students with the planning and organizing and identify their emerging issues in ICT enterprises	**Unit II: Planning and Organizing (6 Hrs.)** 2.1 Planning 2.1.1 Levels of planning: strategic, tactical and operational 2.1.2 Steps in planning 2.1.3 Tools for planning 2.1.4 Importance of planning

	2.2 Organizing 2.2.1 Process of organizing 2.2.2 Organization structure 2.2.3 Types of organization structure 2.2.2.1 Traditional structure: line and functional 2.2.2.2 Modern structure: matrix, network, hybrid 2.4 Emerging planning and organizing issues for ICT enterprises
To enable students to analyze and address key issues in motivating and leading a technical workforce	**Unit III: Motivation and Leadership (6 Hrs.)** 3.1 Motivation 3.1.1 Theories of motivation: Maslow's hierarchy, Herzberg's two factor, Expectancy, Equity 3.1.2 Techniques for motivation 3.2 Leadership 3.2.1 Leadership styles: autocratic, democratic, servant and transformational 3.2.2 Characteristics of learning organization in the ICT industry 3.3 Challenges and strategies for motivating and leading technical workforce
To enhance students' knowledge of human resource management and control functions, emphasizing their practical application for managing ICT organization	**Unit IV: Human Resource Management and Control (8 Hrs.)** 4.1 Human Resource Management 4.1.1 Functions of human resource management 4.1.2 Job analysis, job specification, job description 4.1.3 Recruitment and selection 4.1.4 Human resource training (on the job and off the job) 4.1.5 Performance appraisal and methods 4.1.6 Challenges in managing people in ICT workforce 4.2 Control 4.2.1 Importance 4.2.2 Process and types 4.2.3 Techniques 4.2.4 ICT tools for effective control of engineering projects and organizations.
To expose students to emerging trends in engineering management and their application in ICT driven organizations	**Unit V: Emerging trends in engineering management (4 Hrs.)** 5.1 Participative management, conflict resolution, change management, quality management, innovation management and disruption 5.2 Recent engineering management concepts for managing ICT based projects and organizations

4. Methods of Instruction

Lecture, Tutorials, Discussions, Assignments and Presentation

5. List of Tutorials

S.N.	Tutorials
1	IT companies case studies related to management functions
2.	Students' presentation on course contents, and relevant current management issues
3.	Identification and use of recent ICT based management tools

6. Evaluation system and Students' Responsibilities

Evaluation System

In addition to the formal exam(s) conducted by the Office of the Controller of Examination of Pokhara University, the internal evaluation of a student may consist of class attendance, class participation, quizzes, assignments, presentations, written exams, etc. The tabular presentation of the evaluation system is as follows.

External Evaluation	Marks	Internal Evaluation	Marks
Semester-End Examination	50	Class attendance and participation	10
		Case Study Discussion	5
		Quizzes/assignments and presentations	5
		Internal Term Exam	30
Total External	50	Total Internal	50
Full Marks 50+50=100			

Students' Responsibilities:

Each student must secure at least 45% marks in the internal evaluation with 80% attendance in the class to appear in the Semester End Examination. Failing to obtain such a score will be given NOT QUALIFIED (NQ) and the student will not be eligible to appear in the End-Term examinations. Students are advised to attend all the classes and complete all the assignments within the specified time period. Students are required to complete all the requirements defined for the completion of the course.

7. Prescribed Books and References

Test and Reference Books

1. Harold Koontz and Heinz Weihrich, *Essentials of Management*
2. Prem Raj Pant, *Principles of Management*
3. Govinda Ram Agrawal, *Organization and Management in Nepal*.

Recommended Resources

1. Online resources on content and cases
2. Articles collected from various Journals and Periodicals

Pokhara University

Faculty of Science and Technology

Course No.: CMP 348	Full marks: 100
Course title: **Software Engineering**	Pass marks: 45
Nature of the course: Theory/Tutorial/Practical	Time per period: 1 hour
	Total periods: 45
Level: Undergraduate	Program: BE

1. Course Description

This course intends to provide students with an in-depth knowledge and practical skills pertaining to the field of Software Engineering. It intends to make students familiar with the principles and practices followed in the field of software development and gives a comprehensive understanding of various phases of Software Development activities including Requirements Engineering, Design, Testing and Maintenance. Apart from that the course has also been designed to provide students with a managerial perspective needed to oversee and plan large software development activities using contemporary technologies ensuring product and process quality.

2. General Objectives

The course is designed with the following objectives:

1. To make the students familiar with the principles and practices of Software Engineering required for successfully carrying out development activities
2. To provide students with a detailed knowledge of different phases of software development lifecycle like requirements engineering, architectural design, implementation and testing
3. To equip students with necessary skills and knowledge required for overseeing software development activities including project management, quality assurance, configuration management
4. To familiarize students with the recent trends in the field of software engineering

3. Methods of Instruction

The method of instruction includes lectures, tutorials and practical classes to cover the theoretical, tutorial and practical aspects. Students can be involved in group discussions and presentations in order to assimilate new ideas and current trends in the field of software engineering. Short quizzes can be held in the class to check the students' level of comprehension. Project work needs to be assigned to students to come up with a comprehensive documentation of a software product demonstrating their level of understanding of the contents studied in the course.

4. Contents in detail with specific objectives

Specific Objectives	Contents
The chapter intends to provide a brief introduction of the field of software engineering and software project management and ethics Students make use of the management aspect of software engineering required to undertake software related projects like estimation, measurement, risk management and ethics	**Chapter 1: Software Engineering and Project Management (7 Hrs)** 1.1 Nature and Characteristics of Software 1.2 Software versus System Engineering 1.3 Software Crisis and Myths 1.4 Four Ps of Software Project Management 1.5 Process and Project Metrics 1.6 Measurement of Software: Metrics, Measure, Indicator 1.7 Project Estimation, Empirical Estimation Models 1.8 Software Risks, Assumption, Issues, Dependency 1.8.1 Identification 1.8.2 Mitigation

	1.8.3 Monitoring 1.8.4 Management 1.9 Software Engineering Ethics and Professional Practice
In this chapter students learn about different types of software process models, their strength and weaknesses and their applicability Students also learn to make use of Agile development strategies for software development process	**Chapter 2: Software Process Models and Agility (5 Hr)** 2.1 Software Development Lifecycle (SDLC) 2.2 Waterfall, Incremental, Prototyping, Iterative, Spiral, Rapid Application Development, Aspect Oriented Software Engineering 2.3 Agile Software Development 2.3.1 Extreme Programming 2.3.2 Scrum 2.3.3 Agile Project Management and Scaling Agile Methods 2.4 Pros and Cons of Process Models and their Applicability
This chapter helps students learn how to gather, analyze, and specify software requirements Students make use of various techniques like structured and object oriented approaches for requirements modeling and management	**Chapter 3: Requirements Engineering and Principles (7 Hr)** 3.1 Types of Requirements 3.2 Requirements Modeling Principles 3.3 Domain Analysis and System Models 3.3.1 Context Models 3.3.2 Behavioural Models 3.3.3 Data Models

	3.4 Requirements Engineering Process 3.4.1 Feasibility Study 3.4.2 Requirements Elicitation, Analysis, Modeling, Specification 3.4.3 Requirements Validation 3.5 Object Oriented Analysis
This chapter intends to get students well acquainted with software design principles and patterns. Students use various architectural styles and design patterns including object oriented approach for designing software systems	**Chapter 4: Software Design, Architecture and Principles (7 Hr)** 4.1 Design Modeling Principles 4.2 Design Process 4.3 Architectural design 4.3.1 Layered 4.3.2 Repository 4.3.3 Client Server 4.3.4 Pipe and Filter 4.4 Interface, Component, Database Design 4.5 Design Patterns 4.6 Security by Design 4.7 Object Oriented Design 4.8 Embedded System Design

This chapter imparts knowledge regarding various testing techniques used for proper validation and verification of software. It also lets students become aware of the inevitable changes that occur and ways of maintaining software systems Students design test cases for proper validation and verification of the software product	**Chapter 5: Testing Techniques and Maintenance (7 Hr)** 5.1 Validation and Verification 5.2 Testing Phases 5.2.1 Development Testing 5.2.1.1 Unit Testing 5.2.1.2 Component Testing 5.2.1.3 System Testing 5.2.2 Release Testing 5.2.2.1 Requirements-based Testing 5.2.2.2 Scenario Testing 5.2.2.3 Performance Testing 5.2.3 User Testing 5.2.3.1 Alpha Testing 5.2.3.2 Beta Testing 5.2.3.3 Acceptance Testing 5.3 Test Case Development Strategies 5.3.1 Boundary Value Analysis 5.3.2 Equivalence Partitioning 5.3.3 Basis Path Testing 5.3.4 Control structure Testing 5.4 Software Changes and Evolution 5.5 Maintenance Process and Reengineering

This chapter focuses on quality management of software and the existing standards for compliance and assessing capability In this chapter, students make use of ways of ensuring quality in process and product by adhering to the standards which is essential in producing good software	**Chapter 6: Software Quality Assurance Process (4 Hr)** 6.1 Software Quality Concepts, Software Reliability 6.2 Software Quality Management and Planning 6.3 Software Standards and their Compliance 6.3.1 ISO, CMMI 6.4 Capability Assessment and Process Improvement 6.5 Process and Product Standards 6.6 Reviews and Inspections
This chapter focuses on the need and process of proper management of software configuration items produced as deliverables during the software development process Students use the techniques and tools that exist for keeping track of the software configuration items	**Chapter 7: Software Configuration Management (4 Hr)** 7.1 Software Configuration Items 7.2 Configuration Management Activities 7.2.1 Change Management 7.2.2 Version Management 7.2.3 System Building 7.2.4 System Release

This chapter deals with the recent trends and advancements in the field of software engineering This chapter enables students to make use of contemporary techniques for reusing software and software components and use advanced technologies for software development	**Chapter 8: Advanced Software Engineering Concepts (4 Hr)** 8.1 Software Reuse 8.1.1 Application Frameworks 8.1.2 Software Product Lines 8.1.3 COTS Product Reuse 8.2 Cloud Based Software Engineering 8.3 Artificial Intelligence in Software Engineering

5. List of Tutorials

The following tutorial activities of 15 hours per group of maximum 24 students should be conducted to cover all the required contents of this course.

S.N.	Tutorials
1	Discussions and exercises on Process models and their suitability
2	Exercises on computing metrics for a software system (Function Point, Lines of Code (LOC), COCOMO model) and using them to derive measures and indicators
3	Exercises on computing risk exposure factor and prioritizing risks
4	Exercises on developing test cases using testing techniques like Basis Path Testing, Boundary Value Analysis, Equivalence Partitioning and Control Structure Testing
5	Exercises on performing Domain investigation/analysis and specifying requirements
6	Exercises on drawing Data Flow diagram (Context and lower level diagrams)
7	Exercises on drawing Entity Relationship diagram

8	Exercises on drawing Use case, Sequence, Activity diagram
9	Exercises on drawing Architectural, Component and Class diagram

6. Practical Work

The practical work should consist of exercises focusing on all of the aspects covered in the course pertaining to software development. Students are required to make use of a CASE tool to come up with the documentation of the deliverables of each and every phase of the Software Development Life Cycle including analysis, design, implementation, testing.

A project work should be given to students in which they are required to demonstrate the assimilated knowledge in the course by coming up with detailed artifacts.

S.N.	Practical works
1	Introduction of a CASE tool for Software Engineering Activities (Rational Rose, Star UML, draw.io, Dia, Visio etc.)
2	Identify a domain that needs automation through the use of software and perform Domain analysis/investigation of that domain
3	Perform Requirements analysis of a system and develop a requirements catalog, translate them into task or user story format and enter them in a project management tool (Trello, Jira etc.)
4	Use a CASE tool to develop Data Flow Diagram of the functional requirements captured
5	Use a CASE tool to develop Entity Relationship (ER) diagram
6	Use a CASE tool to develop visual models of the requirements captured using Use case, Sequence and Activity diagram
7	Design the overall architecture of a software system
8	Develop a testing strategy and test cases for testing a software system
9	Use automated tools for properly managing software configuration items and their versions

7. Evaluation system and students' responsibilities

Internal Evaluation

In addition to the formal end-semester exam(s), the internal (formative) evaluation of a student may consist of quizzes, assignments, lab reports, projects, class participation and presentation etc. The tabular presentation of the internal evaluation is as follows. The components may differ according to the nature of the subjects.

Internal Evaluation	**Weight**	**Marks**	**External Evaluation**	**Marks**
Theory		30	Semester-End examination	50
Attendance & Class Participation	10%			
Assignments	20%			
Presentations/Quizzes	10%			
Internal Assessment	60%			
Practical		20		
Attendance & Class Participation	20%			
Lab Report/Project Report	30%			
Practical Exam/Project Work	30%			
Viva	20%			
Total Internal		50		
Full Marks: 50 + 50 = 100				

Student requirements:

Each student must secure at least 45% marks in internal evaluation with 80% attendance in the class in order to appear in the semester-end examination. Failing to get such a score will be equated with NOT QUALIFIED (NQ) and the student will not be eligible to appear in the End- Semester examinations. Students are advised to attend all the classes and complete all the assignments within the specified time period. Failure of a student to attend a formal exam, quiz, test, etc. won't qualify him/her for re-exam. ***Students are required to complete all the requirements defined for the completion of the course***

8. Prescribed Books and References

Text Books:

Ian Sommerville, Software Engineering, 10th Edition, Addison - Wesley, ISBN-13: 978-0-13-703515-1

Roger S. Pressman, Bruce R. Maxim, Software Engineering a Practitioner's Approach. 8th Edition, McGraw Hill, ISBN-13: 978-0-07-802212-8

References:

Grady Booch, James Rumbaugh, Ivar Jacobson, The Unified Modeling Language User Guide, Addison Wesley, ISBN: 0-201-57168-4

Frederick P. Brooks, The Mythical Man Month: Essays on Software Engineering, ISBN-10. 9780201835953

Derek Partridge, Artificial Intelligence and Software Engineering, ISBN 9780893916060

Muthu Ramacharan, Zaigham Mahmood, Software Engineering in the Era of Cloud Computing, 2021, ISBN: 978-3-030-33623-3

Pankaj Jalote, Software Engineering- A Precise Approach, Latest Edition

Rajib Mall, Fundamental of Software Engineering, Latest Edition