Pokhara University
Faculty of Science and Technology

Course Code: MEC 116
Course Title: Basic Engineering Drawing (0-0-3)
Nature of the Course: Practical
Level: Bachelor

Full Marks: 100
Pass Marks: 45
Total Duration: 45 hours
Program: BE

Course Description
This course is designed to provide students the knowledge and skills to draw, visualize and represent objects manually as well as with the application of computer aided techniques. The course will be delivered using tutorials and self-learnings by the students.

1. General Objectives
The general objectives of this course are:
- To enhance knowledge and skills to draw and visualize geometrical shapes of objects
- To enable students to draw, visualize and representation objects using Computer aided techniques.

2. Methods of Instruction
Lecture, discussions, demonstration, tutorials and assignments

3. Contents in Detail

Specific Objectives	Contents
Recognize the drawing instruments, drawing sheets, lettering and dimensioning.	**Unit I: Introduction to engineering drawing (3 hrs)** 1.1 Manual drawing instruments, drafting machines, drawing paper and materials, preparation for drawing, cautions in use of instruments, drawing sheets-their layout and planning 1.2 Technical lettering and dimensioning: Single-stroke letters, capital and lowercase letters, vertical and slant lettering, vertical and inclined numerals 1.3 Procedure for lettering, dimensioning terms and notations, theory of dimensioning, system of dimensioning, use of scales, units and general rules of dimensioning
Draw basic geometrical shapes	**Unit II: Geometrical Constructions (7 hrs)** 2.1 Construction involving lines and angles, bisecting and trisecting lines and angles, division of lines, proportional division of lines 2.2 Construction of polygons, constructions using tangents circles and arcs, open and cross belt tangents 2.3 Construction of conic and engineering curves: ellipse (four center method), parabola (Tangent method), hyperbola (eccentricity method), cycloid, involute, Archimedean spiral, helix.
Visualize objects through	**Unit III: Multi-view Drawings & Sectional Views (15 hrs)**

orthographic projections	3.1 Introduction to projection (point, line plane), orthographic projection 3.2 Selection of views, ways for making a multi-view drawing 3.3 Introduction of sectional views, half and full sectional views, offset sectional view, hatching lines
Illustrate the surface development	**Unit IV: Developments of surfaces (4 hrs)** 4.1 Introduction of surface development 4.2 Complete developments of truncated right solids (Cylinder, Cone, Pyramid and Prism), frustums of right solids (Cone & Pyramid)
Visualize and draw objects through isometric, oblique and perspective drawings	**Unit V: Isometric, Oblique and Perspective Projections (6 hrs)** 5.1 Introduction of Axonometric projection, isometric projection, methods and procedure for making an simple isometric drawing 5.2 Introduction of oblique projection and oblique drawing 5.3 Perspective projection, position of object, construction of one-point
Recognize symbols and use computer aided techniques to visualize and draw objects	**Unit VI: Symbols and Computer Aided Drawing (10 hrs)** 6.1 Electrical and Electronics symbols 6.2 Introduction to AutoCAD, Basic commands for 2D drawing like: Line, Circle, Polyline, Rectangle, Hatch, Fillet, Chamfer, Trim, Extend, Offset, Dim style, etc. 6.3 Basics of 3D drawings (In computer laboratory with only demonstration and practices)

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

4. Laboratory Work (45 hrs for a group of maximum 24 students)
Lettering and use of drawing instruments; Dimensioning; Geometrical and Projection drawing; Multiview drawings; Sectional views, Development of surfaces; Axonometric projection; Symbols and AutoCAD Drawing

5. Evaluation System and Students' Responsibilities

Evaluation System
The evaluation of a student may consist of attendance, assignments, term-exams, projects etc. The final examination will be held by the PU Examination Controller's Office. The internal and external evaluation detail is given in the table below:

Internal Evaluation	**Weight**	**Marks**	**External Evaluation**	**Marks**
Practical				
Attendance and Class Participation	10%		Semester-End examination	50
Drawing Sheets Evaluation	30%			
Assignment	10%			
Internal Assessment	50%			
Total Internal		50		
Full Marks: 50 + 50 = 100				

Students' 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 score will be given NOT QUALIFIED (NQ) to appear 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.

6. Prescribed Books and References

Text Book

1. Luzadder, W. J. & Duff, J. F (2015). *Fundamentals of Engineering Drawing with an introduction to Interactive Computer Graphics for Design and Production*, Pearson India Education Services.

References

1. Gill, P. S. (2009). *Engineering Drawing*, Seagull Books Pvt Ltd. India.
2. Dhawan, R. K. (2019). *A Textbook of Engineering Drawing*, S. Chand Publishing.
3. Omura, G. (2012). *Mastering AutoCAD 2013 and AutoCAD LT 2013*, John Wiley & Sons.

Pokhara University
Faculty of Science and Technology

Course No.: MTH 110
Course title: Calculus I (3-2-0)
Nature of the course: Theory
Level: Bachelor

Full marks: 100
Pass marks: 45
Total lectures: 45 Hrs
Program: BE

1. Course Description

This course is designed for developing competency of the students on the fundamental concepts, principals and applications of differential and integral calculus for solving engineering problems. It is equipped with differential calculus, integral calculus and ordinary differential equations. The review part of the content is based on previous learning in the school level. The course will be delivered through lecture method, assignment on practically base engineering problems and class tests.

2. General Objectives

The course is designed with the following general objectives:

- To acquaint the students with applications of differential and integral calculus in engineering.
- To expose the students with the application of differential equations for modeling engineering problems.

3. Methods of Instruction

Lecture, tutorials, discussions and assignments

4. Contents in Detail

Specific objectives	Contents
• Explain the importance of limit and continuity in differential problems and use Leibnitz theorem to evaluate higher derivatives.	**Unit I: Limit Continuity and Derivatives (5 hrs)** 1.1 Introduction 1.2 Limit, continuity and differentiability 1.3 Higher order derivatives by Leibnitz method.
• Apply derivatives in mean value theorem, series expansion, asymptotes and trace curve for the given function.	**Unit II: Applications of Derivatives (8 hrs)** 2.1 Mean value theorems: Rolle's theorem, Lagrange's Theorem (Geometrical interpretation and verification) and applications 2.2 Higher order mean value theorem: Taylor's Series, Maclaurin's Series expansion of function. 2.3 Asymptotes to Cartesian curves up to four degrees. 2.4 Curve tracing in Cartesian form and parametric form 2.5 Curvature
• Evaluate Proper and improper integrals.	**Unit III: Integral Calculus (6 hrs)** 3.1 Introduction 3.2 Review on Indefinite Integral and fundamental theorem of integral calculus.

	3.3 Definite integral and its properties 3.4 Improper Integrals; comparison test. 3.5 Reduction formula, Beta Gamma functions
• Evaluate arc length, area, and volume through integration.	**Unit IV: Application of Integral (6 hrs)** 4.1 Application of integrals for finding area beneath a curve and between two curves and arc length 4.2 Surface and volume of solid of revolution in the plane for Cartesian and parametric curves.
• Compute partial derivatives with the concept of total differentials.	**Unit V: Partial Differentiation (3 hrs)** 5.1 Introduction 5.2 Partial Derivatives 5.3 Homogeneous function and Euler's theorem for the function of two and three variables 5.4 Total Derivatives and Differentiation of Implicit functions.
• Define extreme value and compute its value for two and three variables through partial derivatives.	**Unit VI: Application of Partial Differentiation (4 hrs)** 6.1 Extrema of functions of two and three variables. 6.2 Lagrange's method of undetermined Multipliers (up to 2 multipliers)
• Solve first order differential equations.	**Unit VII: First Order Ordinary Differential Equations (6 hrs)** 7.1 Review of separable, homogeneous and exact differential equation with engineering applications 7.2 Linear, Bernoulli equation and Riccati's equation with engineering application. 7.3 Mathematical modeling of engineering problems using first order equation.
• Solve second order differential equations in relation to engineering problems.	**Unit VIII: Second Order Ordinary Differential Equations (7 hrs)** 8.1 Second order Homogeneous ODE with constant and variable coefficients, Euler-Cauchy equation. 8.2 Existence and uniqueness of solutions, Wronskian and general solutions for solving ODE. 8.3 Non-homogeneous second order ODE and Solution by undetermined coefficients and variation of parameters and engineering application

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

5. List of Tutorials

Tutorial work covers the work to be done in tutorial. This will enable the students to compute the mathematics problem under the supervision of the course leader. The major tutorial works are as follows:

Total: 30 Hours

Unit no.	Unit name	List of Tutorials	Tutorial hours
1	**Limit Continuity and Derivatives**	1.1 Problems on Limit and continuity. 1.2 Show that differentiability implies continuity but the converse may not be true 1.3 Evaluation of higher order derivatives by Leibnitz method	1 hr 1 hr 1 hr
2	**Applications of Derivatives**	2.1 Problems in Mean value theorems: Rolle's theorem, Lagrange's Theorem 2.2 Expand the functions through Taylor's Series, and Maclaurin's Series 2.3 Evaluation of Asymptotes to Cartesian curves. 2.4 Trace Curve for the equations in Cartesian form and parametric form 2.6 Problems in Curvature	1 hr 2 hrs 2 hrs 2 hrs 1 hr
3	**Integral Calculus**	3.1 Evaluation of Indefinite Integrals, Definite integrals, Improper Integrals; 3.2 Deduce Reduction formula, and solve problems related to Beta Gamma functions.	2 hrs 2 hrs
4	**Application of Integral**	4.1 Evaluation of area, arc length. 4.2 Evaluation of Surface volume of solid of revolution in the plane for Cartesian and parametric curves.	1 hr 2 hrs
5	**Partial Differentiation**	5.1 Prove Euler's theorem for the function of two and three variables 5.2 Calculate total derivatives and differentiation of Implicit functions.	1 hr 1 hr
6	**Application of Partial Differentiation**	6.1 Evaluation of Extrema of functions of two and three variables and Lagrange's method of undetermined Multipliers (up to 2 multipliers)	2 hrs
7	**First Order Ordinary Differential Equations**	7.1 Solution of separable, homogeneous and exact differential equation Linear, Bernoulli equation and Riccati's equation with engineering applications 7.2 Mathematical modeling of engineering problems using first order equation.	2 hrs 1 hr
8	**Second Order Ordinary Differential Equations**	8.1 Solve second order homogeneous ODE with constant and variable coefficients, Euler-Cauchy equation. 8.2 Solve non-homogeneous second order ODE by undetermined coefficients and variation of parameters in engineering application	3 hrs 2 hrs

6. Evaluation System and Students' Responsibilities

Evaluation System
Internal evaluation is done as follows:

Internal Evaluation	Marks	External Evaluation	Weight	Marks
Attendance & Class Participation	10%	**Semester End Board Examination**	**50%**	50
Assignments	20%			
Presentations/Quizzes	10%			
Term exam	60%			
Total Internal	50			
Full Marks: 50 + 50 = 100				

Students' Responsibilities
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 score will be given NOT QUALIFIED (NQ) and the student will not be eligible to appear 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.

7. Prescribed Books and References

Text Books

1. Kreyszig, E. *Advance Engineering Mathematics,* New Delhi: John Wiley and Sons Inc.
2. Stewart, J. *Calculus, Early Transcendental.* India; Cengage Learning.

References

1. Thomas, G. & Finney, R. *Calculus and Analytical Geometry.* New Delhi: Narosa Publishing House.
2. Mishra, P., Mishra, R., Mishra, V. P., & Mishra, M. *Advance Engineering Mathematics.* New Delhi: V. P. Mishra Publication.
3. Dass, H. K. & Verma R. *Higher Engineering Mathematics.* New Delhi: S Chand Publishing.

Pokhara University
Faculty of Science and Technology

Course Code: CMP 122
Course Title: Computer Workshop (0-0-3)
Nature of the Course: Practical
Level: Bachelor

Full Marks: 100
Pass Marks: 45
Total Duration: 45 hours
Program: BE Computer/IT/Software

1. Course Description

This course provides the knowledge and hands-on skills of computer hardware, software, computer networking and enables students to identify and rectify the onboard computer hardware, software and network related problems. Students will be able to understand the hardware specifications for the operating system and various application programs.

General Objectives

The general objectives of this course are:

- To familiarize the students with the computer systems hardware, basic practical works and computer networking concepts
- To make the students competent in install/update operating system and various application software, manage data backup and restore operations on a computer system

2. Methods of Instruction

Practical Works and Project works

3. Contents in Detail

Specific Objectives	Contents
• Recognize the computer hardware and accessories.	**Unit I: Introduction to Desktop/Workstation Assembly** 1.1 Demonstration of computer hardware and peripheral devices: Processors, Motherboard, Memory, Storage devices, I/O Devices. 1.2 Demonstration of Power Supply-SMPS, Internal cablings and Ports.
• Assemble and Dissemble PC with power supply, internal cabling, Motherboard and PCI Device	**Unit II: Assembling Desktop Computer** 2.1 Assembling steps and precautions. 2.2 Setting of motherboards, memory, hard disk, processors, CDROM, SMPS and other devices.
• Assemble Laptop power supply, internal cabling, Motherboard and PCI Devices.	**Unit III: Assembling Laptop** 3.1 Assembling steps and precautions. 3.2 Demonstration of motherboards, memory, hard disk, processors, CDROM, SMPS and other devices.

• Install OS (Linux and Windows) in Desktop/Laptop	**Unit IV: Laptop/Desktop OS Installation** 4.1 BIOS setting, firmware types, BOOT configuration. 4.2 Hard disc partitioning. 4.3 OS installation (Linux, windows). 4.4 User account management 4.5 Printer, Scanner installation.
• Recognize computer networks-wired LAN and Wireless LAN configuration.	**Unit V: Networking and Internet Setup** 5.1 Introduction to Computer Network, Network topologies, wired and wireless networking media. 5.2 Network cabling, cable types and connectors. 5.3 Basic wireless AP configuration.
• Install and update application software and utility software. • Scan and remove viruses from computer system.	**Unit VI: Software Installation** 6.1 Installation of application and utility software. 6.2 Update of firmware patches, 6.3 Virus scanning.
• Know the functioning of server of Mail/Data/Domain/FTP. • Apply FTP/SCP Client tool to upload/download files to FTP server.	**Unit VII: Server Installation and Backup** 7.1 Introduction to server. 7.2 IP Address setting, Sharing of Printer/Device 7.3 Study of different server (Email, Data, Domain, FTP).
• Use the basic troubleshooting tools and utilities.	**Unit VIII: Hardware and Software Troubleshooting** 8.1 Basic Repair and Maintenance of Desktop/Laptop. 8.2 Replacement of Passive Components. 8.3 Use of basic software troubleshooting commands. 8.4 Use of trouble shooting tools and utilities.

5. **Practical Works**

Laboratory works of 45 hours per group of maximum 24 students should cover all the topics stated in the content details of this course.

6. Evaluation system and Students' Responsibilities

Evaluation System

The evaluation of a student may consist of attendance, lab reports, projects works and viva etc. The tabular presentation of the evaluation is as follows:

Internal Evaluation	Weight	Marks	External Evaluation	Marks
Practical			Semester-End examination	xx
Attendance and Class Participation	20%			
Project Work	40%			
Report	20%			
Quizzes /Viva	20%			
Total		100		

Student's Responsibilities

Each student must secure at least 45% marks in internal evaluation with 80% attendance in the class in order to pass this course. 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.

7. Prescribed Books and References

References

1. Bose, S. K. (1996). *Hardware and Software of Personal Computers.* New Age International.
2. Rajaraman, V. and Adabala, N. (2014). *Fundamentals of computers.* PHI Learning.
3. Rosch, W. L. (2003). *Winn L. Rosch hardware bible.* Que Publishing.

Pokhara University
Faculty of Science and Technology

Course Code: ELX 110
Course title: Digital Logic (3-1-2)
Nature of the course: Theory & Practical
Level: Bachelor

Full Marks: 100
Pass Marks: 45
Total Lectures: 45 hours
Program: BE

1. Course Description

This course covers the various concepts of digital logic systems. This course emphasizes on fundamental concept, principles and properties of Boolean algebra and its application in simplification, circuit analysis and gate implementation. It covers the use of flip flops in the design of synchronous and asynchronous sequential logic circuits. It also covers the ALU design.

2. General Objectives

This course provides basic knowledge of logic systems, introduces basic tools to design various digital logic circuits and enables the students to design a basic digital computer.

3. Methods of Instruction

Lecture, Tutorial, Discussion, Readings and Practical works

4. Contents in Detail

Specific Objectives	Contents
• Compare analog and digital signal and system. • Use the information representation in any number system.	**Unit I: Introduction (2 hrs)** 1.1 Analog and digital signal 1.2 Analog and digital system 1.3 Numerical representation 1.4 Digital number system
• Deduce conversions among binary number systems. • Apply the different coding system to represent information.	**Unit II: Number Systems and Codes (6 hrs)** 2.1 Number systems 2.1.1 Decimal 2.1.2 Binary 2.1.3 Octal 2.1.4 Hexadecimal 2.2 Number system conversion 2.3 Complements (radix and diminished-radix) 2.4 Subtraction using complements 2.5 Binary coding systems 2.5.1 Weighted codes (BCD, 8 4 -2 -1, and 2 4 2 1) 2.5.2 Non-weighted codes (Excess-3 and Gray) 2.6 Alphanumeric and instruction codes
• Implement the simplified functions using simple and	**Unit III: Boolean Algebra and Logic Gates (4 hrs)** 3.1 Boolean algebra (definition, properties, postulates and theorems)

universal logic gates.	3.2 Logic gates, truth tables and Boolean function 3.3 Duality principle and complements 3.4 Gate implementation 3.5 Universality of NAND and NOR gates
• Simplify the Boolean function using map method.	**Unit IV: Simplification of Boolean Function (5 hrs)** 4.1 Venn diagram 4.2 Canonical forms and standard forms 4.3 Karnaugh map up to 5 variables 4.4 Minimum realization 4.5 Don't care conditions 4.6 Simplification in SOP and POS using K-map
• Design various combinational logic circuits and analyze them.	**Unit V: Combinational Circuit (4 hrs)** 5.1 Design procedure 5.2 Adder and subtractor 5.3 Code conversion 5.4 Analysis procedure 5.5 NAND and NOR implementation 5.6 Multilevel NAND and NOR gates 5.7 Parity generator and checker
• Design and implement the parallel adder/subtractor, comparator, multiplexer/demultiplexer and encoder/decoder.	**Unit VI: MSI and LSI Design (6 hrs)** 6.1 Introduction to Integration technology 6.2 Parallel adder and subtractor 6.3 Decimal / BCD adder 6.4 Magnitude comparator 6.5 Multiplexer and demultiplexer 6.6 Encoder and Decoder, 6.7 ROM and PLA
• Design and analyze sequential logic circuits.	**Unit VII: Sequential Circuits (6 hrs)** 7.1 Synchronous and asynchronous logic 7.2 Differences between Latch and fli-flop, Flip flops (RS, JK, D, T) and their truth table, excitation table and characteristic equation 7.3 Triggering of flip flops 7.4 State diagram and state table 7.5 State reduction and binary assignment 7.6 Design and analysis of clocked sequential circuit 7.7 Master-slave flip flops
• Design synchronous and asynchronous counters.	**Unit 8: Registers and Counters (6 hrs)** 8.1 Register, shift register and types of Shift register 8.2 Synchronous counters 8.2.1 up to 4-bit counters 8. 3 Asynchronous counters 8.3.1 BCD ripple counter, 8.3.2 Mod counter 8.4 Ring counter 8.5 Output hazard race

• Design and implement arithmetic logic unit.	**Unit 9: Memory Unit and ALU (6hrs)** 9.1 Random access memory 9.2 Design of arithmetic logic unit 9.3 Accumulator 9.4 Shifter and status register 9.5 Processor unit

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	Discussion-based Tutorials • Analog and digital signal and system. • IC technology and parameters considered during fabrication • Sequential circuits and types • Output hazard races
2	Problem solving-based Tutorials • Conversion of numbers among number systems • Simplification of Boolean functions in SOP and POS using theorems and postulates. • Simplification of Boolean function in SOP and POS using K Map. • Logic gate implementation, used of NAND and NOR gates. • Implementation of Boolean function using various MSI and LSI components. • Design of various combinational circuits, code conversion circuits and parity generation and checking circuits. • Design of synchronous sequential circuit from state diagram. • Design of synchronous and asynchronous counters. • Design of arithmetic and logic unit.

6. Practical works (30 hours for a group of maximum 24 students)

S.N.	Practical Works
1	Familiarization with logic gates
2	Familiarization with Boolean functions
3	Design of simple combinational circuits.
4	Adder and subtractor
5	Encoder and decoder
6	Multiplexer and demultiplexer
7	Design of flip flops
8	Registers and counters

7. Evaluation system and Students' Responsibilities

Evaluation system

In addition to the formal exam(s), the internal evaluation of a student may consist of quizzes, assignments, lab reports, projects and class participation. The tabular presentation of the internal evaluation is as follows.

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

Student's Responsibility

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 score will be given NOT QUALIFIED (NQ) to appear 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 Book

1. Mano, M. M. (2017). *Digital logic and computer design*. Pearson Education India.

References

1. Rafiquzzaman, M., & McNinch, S. A. (2019). *Digital Logic: With an Introduction to Verilog and FPGA-Based Design*. John Wiley & Sons.
2. M. Morris Mano(2002). *Digital Design*. India: Prentice Hall.

Pokhara University
Faculty of Science and Technology

Course No.: CMP 116
Course title: Discrete Structure (3-1-0)
Nature of the course: Theory
Level: Bachelor

Full Marks: 100
Pass Marks: 45
Total Lectures: 45 hours
Program: BE

1. Course Description

This course covers the basic principles of discrete structure that form the essential foundation for every area of computer science. This course first introduces the basics of discrete structure and then discusses the different sorts of logics along with the mathematical reasoning methods. This course also elaborates the graph theory used in today's computer science and finally introduces the language and grammars to build the automata.

2. General Objectives

The general objectives of this course are:

- To provide mathematical foundation for the computational science.
- To acquaint the student with the knowledge of propositional logic, predicate logic and mathematical reasoning in these logics.
- To make students cognizant to build the finite state automata.

3. Methods of Instruction

Lecture, Discussion, Readings and Case Study

4. Contents in Detail

Specific Objectives	Contents
• Apply the set theory, functions and basic of counting.	**Unit I: Basic Discrete Structures (8 hrs)** 1.1 Sets, Set Operations, Venn Diagram, Inclusion-Exclusion Principle and Computer Representation of Sets 1.2 Basic Concept of functions, Injective and Bijective Functions, Inverse and Composite Functions, Sketch Functions, Functions for Computer Science (Ceiling Function, Floor Function, Boolean Function, Exponential Function) 1.3 Integers and Division, Primes and Greatest Common Divisor, Extended Euclidean Algorithm 1.4 Matrices: Zero-One Matrices, Boolean Matrix Operations 1.5 Basics of Counting, Pigeonhole principle and Permutations and Combinations
• Identify the relations and their properties to deduce the closures of relations.	**Unit II: Relations (6 hrs)** 2.1 Relations and their types 2.2 Properties of Relations 2.3 Representation of relations 2.4 Closures of Relations 2.5 Equivalence Relations

	2.6 Partial Orderings
• Deduce the recurrence relations.	**Unit III: Recurrence Relation (7 hrs)** 3.1 Recursive Definition of Sequences 3.2 Solution of Linear recurrence relations 3.3 Solution to Nonlinear Recurrence Relations 3.4 Application to Algorithm Analysis
• Apply the mathematical reasoning methods for reasoning in the propositional and the predicate logic.	**Unit IV: Logic, Induction and Reasoning (8 hrs)** 4.1 Propositions and Truth Functions 4.2 Propositional Logic 4.3 Propositional Equivalences 4.4 Predicate logic and Quantifiers 4.5 Deduction in Predicate logic 4.6 Rules of Inferences 4.7 Mathematical Reasoning- Direct Proof and Indirect Proof (Proof by Contradiction and Proof by Contraposition) 4.8 Mathematical Induction
• Illustrate the applications of graph theory to resolve the various real-world problem.	**Unit V: Graph Theory (9 hrs)** 5.1 Graph and its types. 5.2 Graph Representation (Adjacency matrix, Incidence Matrix and Path Matrix) 5.3 Walk, Path, Trails, and Circuits (Cycle) 5.4 Regular graph, complete graph, cycle graph, connected graph, simple and bipartite graph, Eulerian Graph, Hamilton Graph 5.5 Transport Network, Max-Flow and Min-Cut Theorem 5.6 Applications of graph theory
• Deduce deterministic and non-deterministic finite automata and compare them.	**Unit VI: Language, Grammar and Automata (7 hrs)** 6.1 Language and Grammars 6.2 Language and Automata 6.3 Finite State Automata 6.4 Deterministic Finite Automata 6.5 Non-Deterministic Finite Automata 6.6 Regular Expressions

5. List of Tutorials

The various tutorial activities that suits 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 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:

1. Problem solving-based Tutorials: (10 hrs)
    1. Solving whether a function is injective, surjective or bijective.
    2. Representing Boolean operation in Boolean matrix.
    3. Solving homogenous recurrence relation with single, double (double but same), triple different roots
    4. Solving nonlinear recurrence relation (quadratic, exponent, linear)

5. Solving goal related problem using rules of inference in propositional and predicate logic
6. Verifying formula using mathematical induction
7. Problems related to representing graph using incidence, path and adjacency matrix.
8. Applying max-flow min cut theorem in network graph.
9. Design of finite state automata and non-deterministic finite state automata to deterministic finite automata conversion problems
10. Design of finite state automata to check regular expression is accepted or not.

2. Review and Question/Answer-based Tutorials: (5 hrs)
11. Case study on real use of graph in real time scenario.
12. Students ask questions within from the course content and assignments and review key course content in preparation for tests or exams

6. Evaluation System and Students' Responsibilities

Evaluation System

The internal evaluation of a student may consist of assignments, attendance, term-exams, and project works etc. The internal evaluation scheme for this course is as follows:

Internal Evaluation	**Weight**	**Marks**	**External Evaluation**	**Marks**
Theory		50	Semester-End examination	50
Attendance & Class Participation	10%			
Assignments	20%			
Presentations/Quizzes	10%			
Internal Assessment	60%			
Total Internal		50		
Full Marks: 50 + 50 = 100				

Students' Responsibilities

Each student must secure at least 45% marks in internal assessment evaluation with 80% attendance in the class in order to appear in the Semester End Examination. Failing to get such score will be given NOT QUALIFIED (NQ) to appear 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.

7. Prescribed Books and References

Text Books

1. Kolman, B., Busby, R. C., & Ross, S. (1995). *Discrete mathematical structures.* Prentice-Hall, Inc.

References

1. Johnsonbaugh, R. (2009). *Discrete mathematics*. Pearson.
2. Mott, J. L., Kandel, A., & Baker, T. P. (Eds.). (1986). *Discrete mathematics for computer scientists & mathematicians*. Prentice-Hall, Inc.

3. Rosen, K. H. (2011). *Discrete mathematics and its applications*. William C Brown Pub.

Pokhara University
Faculty of Science and Technology

Course No.: MTH 120
Course title: Problem Solving Techniques (3-1-0)
Nature of the course: Theory
Level: Bachelor

Full marks: 100
Pass marks: 45
Time per period: 1 hour
Total Periods: 45
Program: BE (IT, Software)

1. Course Description

Many everyday activities involve problem solving using some kinds of techniques. In order to solve the problems, one must think analytically to find a solution of the problem. This course is designed to provide the students with the basic principles of problem solving techniques that include both mathematical problems and non-mathematical problems. It deals with methodology of analytic thinking and finding appropriate techniques to solve a given problem.

2. General Objectives

The general objectives of this course are:

- To acquaint the students with the conceptual clarity in thinking analytically and logically to solve real-life problems.
- To provide the students with the knowledge of fundamental approaches of how to translate verbal discussion onto analytical data and then how to solve it by computer.

3. Methods of Instruction

Lecture, Discussion, Readings and Case Study.

4. Contents in Detail

Specific Objectives	Contents
• State various types of mathematical and non-mathematical problems.	**Unit I: Basic Concepts of Problem Solving (6 hrs)** 1.1 A First Problem 1.2 How to count 1.3 The Use of Induction 1.4 Problem of logics 1.5 Issues of Parity

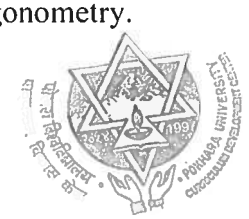
• Use the concept of geometry in solving real world problems.	**Unit II: Application of Geometry (6 hrs)** 2.1 Classical Planer Geometry 2.2 Analytic Geometry 2.3 Miscellaneous and Exotic Geometry Problems 2.4 Solid Geometry
• Deduce the solutions to the problems that need to think analytically and logically.	**Unit III: Miscellaneous Problem-Solving Techniques (10 hrs)** 3.1 Probabilistic Approach to Solving Counting Problems 3.2 Logic Problems • Simple Logic • Theory of Games • Tracing Routes • Learning from Parity • Mysterious Arithmetic Problems • Surprise 3.3 Problems from Recreational Math • Magic Square • Weighing Problems 3.4 Problems of Algebra and Analysis • Inequality • Trigonometry and related ideas
• Apply mathematical concept to solve real-life problems.	**Unit IV: Solving Miscellaneous Real-Life Problems (8 hrs)** 4.1 Miscellaneous Problems 4.2 Impossible Problems 4.3 Problems from Everyday Life 4.4 Statistics

5. List of Tutorials

The following tutorial activities of 15 hours per group of maximum 24 students should be conducted to cover the content of this course:

A. Problem solving-based Tutorials: (9 hrs)

1. Solve problems related to counting(Counting number of zeros in factorial or in given expression, handshaking problem,)
2. Verify mathematical formulas using mathematical induction.
3. Solving problems related with analytic and solid geometry.
4. Solving problems related with game theory, weighing and magic square.
5. Solving problems of counting using a probabilistic way.
6. Solving problem of tracing routes.
7. Solving problems related to inequality and trigonometry.
8. Solving problem related to everyday life.
9. Solving problems related with statistics.



B. Review and Question/Answer-based Tutorials: (6 hrs)

1. Case study of "Impossible problems" followed by Oral Presentation in class.
2. Students ask questions within from the course content and assignments and review key course content in preparation for tests or exams.

5. Evaluation System and Students' Responsibilities

Evaluation System

The internal evaluation scheme for this course is as follows:

Internal Evaluation	**Weight**	**Marks**	**External Evaluation**	**Marks**
Theory		50	Semester-End examination	50
Attendance & Class Participation	10%			
Assignments	20%			
Presentations/Quizzes	10%			
Internal Assessment	60%			
Total Internal		50		
Full Marks: 50 + 50 = 100				

Student Responsibilities

Each student must secure at least 45% marks in internal assessment evaluation with 80% attendance in the class in order to appear in the Semester End Examination. Failing to get such score will be given NOT QUALIFIED (NQ) to appear 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.

6. Prescribed Books and References

Text Book

1. Krantz, S. G. (1997). *Techniques of problem solving*. Universities Press.

References

1. Etter, D. M. (1994). *Engineering problem solving with ANSI C: fundamental concepts.* Prentice-Hall, Inc..
2. Lakatos, I. (1976). *Proofs and Refutation The Logic of Mathematical Discovery.* Cambridge University Press.
3. Polya, G. (2004). *How to solve it: A new aspect of mathematical method.* Princeton university press.

Pokhara University
Faculty of Science and Technology

Course No.: CMP 124

Course title: **Programming in C (3-1-3)**

Nature of the course: Theory & Practical

Level: Bachelor

Full marks: 100

Pass marks: 45

Total Lectures: 45 hrs

Program: BE (Computer, IT and Software)

1. Course Description

This course is designed to develop the skills in students to use the C language, which follows the structured programming paradigm, to develop the computer programs. It introduces the different generations of programming languages, the origin, strengths and basic constructs of the C language. After completion of this course, the students will be able to use the C language to resolve a given problem through the problem solving steps- problem analysis, design of algorithm and flowchart, coding using the C language, executing and compiling the developed program, testing and debugging the program and finally well documenting the program for the future understanding.

2. General Objectives

- To acquaint the students with basic knowledge of computer language and generations of programming languages.
- To develop the skills in students to solve a given problem using computer program.
- To acquaint the students with the knowledge of structured programming paradigm (using the C language) to develop the computer programs.

3. Methods of Instruction

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

4. Contents in Detail.

Specific Objectives	Contents

- Explain, compare and classify programming languages and its generations. - Explain and compare the types of software. - Design and develop algorithms and flowcharts. - Familiarize with computer program documentation.	**Unit 1: Programming languages and problem solving (6 hrs)** 1.1 Programming Languages (machine-level language, assembly language and high-level language) and its generations. 1.2 Software and its types 1.3 Structured programming 1.4 Problem solving using computer- Problems Analysis (understanding of the problem, feasibility and requirement analysis), Design (Algorithm and flowchart), Coding (compilation/interpretation and execution), Testing and debugging, Implementation, Evaluation and Maintenance of computer programs, Program documentation
- Describe the origin of the C language and know its strength. - Understand the basic constructs of the C language.	**Unit 2: Introduction to C (6 hrs)** 2.1 History of C language 2.2 Features of C 2.3 The C as a middle-level language 2.4 The C as a system programming language 2.5 The C character set 2.6 Keywords and Identifiers 2.7 Data types 2.8 Constants, variables and their declaration 2.9 Formatted input/output functions 2.10 The C Operators
- Understand and implement the various control structures of the C language.	**Unit 3: Control Structures (6 hrs)** 3.1 Introduction and types of control statements- sequential, branching and looping statements 3.2 Branching statements- simple if statement, if-else, nested if, if-else-if ladder and switch statements 3.3 Looping statements- for loop, while loop, do-while loop, nested loop 3.4 The break, continue and goto statements

- Develop the C program that uses the various types of single and multi-dimensional arrays.	**Unit 4: Arrays and Strings (6 hrs)** 4.1 Introduction to arrays 4.2 One dimensional and Multidimensional arrays 4.3 Initialization of arrays and accessing the elements of arrays 4.4 Strings- the character arrays 4.5 Functions related to the strings
- Design and develop the C programs using functions.	**Unit 5: Functions (6 hrs)** 5.1 Introduction 5.2 Importance of functions 5.3 Returning a value from a function and sending a value to a function 5.4 Function prototypes 5.5 Calling a function- Call by value and Call by reference 5.6 Recursive functions 5.7 Passing an array to a function 5.8 Local variables, formal parameters and global variables 5.9 Storage classes 5.10 Pre-processor directives- C libraries, macros and header files
- Use the pointers in arrays, functions and programs to dynamically allocate and deallocate memory.	**Unit 6: Pointers (6 hrs)** 6.1 Introduction 6.2 Pointer operators 6.3 Pointer arithmetic 6.4 Returning multiple values form functions using pointers 6.5 Pointers and Arrays 6.6 Double indirection 6.7 Dynamic memory allocation

- Use the structures and unions to store and access the heterogenous data required in a program.	**Unit 7: Structure and Union (5 hrs)** 7.1 Definition of Structure 7.2 Nested-Structure 7.3 Array of Structure 7.4 Structures and Pointers 7.5 Union 7.6 Self-referential structure
- Use the C file handling concepts to store the data permanently in a computer file and access them whenever required.	**Unit 8: Files and File Handling (4 hrs)** 8.1 FILE pointer, File opening modes (read, write, append) 8.2 File handling functions 8.3 Creating and operating a file in different modes

5. **Practical Works**

Laboratory works of 45 hours per group of maximum 24 students should cover all the concepts of C language studied in the lectures. Students should submit a final project that uses all the constructs and features of C studied in this course. The marks for the practical work will be based on the project work.

6. List of Tutorials:

The various tutorial activities that suits this course should cover all the content of this course to give student a space to engage more actively with the course content in the presence of 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: (2 hrs)

1. Evolution of Programming languages and its generations (Class discussion)
2. Software and its types.
3. Generations of programming languages.
4. Structured programming. (Oral Presentation).

B. Problem solving-based Tutorials: (10 hrs)

1. Develop algorithms and flowcharts to solve various problems such as to find largest number among three numbers, prime numbers, temperature conversion, product of matrices, finding sum of the terms in series, printing various patterns etc.
2. Develop the C programs for the problems for which you developed the algorithms.
3. Write a program to pass an array to a function.
4. Write a program to use pointers to pass multiple values from a function.
5. Write a program to use the basic string functions to manipulate string data.

6. Write a program to use the principle of recursion to solve the complex problems such as to find factorial of a number, fibonacci series.
7. Write a program to illustrate the macros and header files.
8. Write a program to to illustrate how memory is allocated and deallocated in C language.
9. Write a program to use the nested structure. Discuss the scenarios when the structures and unions are used in real practice.
10. Write a program to solve simple file handing problems.

C. Review and Question/Answer-based Tutorials: (3 hrs)

1. Case study of "Development of C with the UNIX operating system and origin of C++ languages" followed by Oral Presentation in class.
2. Students ask questions within from the course content and assignments and review key course content in preparation for tests or exams.

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

Internal Evaluation

The internal evaluation of a student may consist of assignments, attendance, internal assessment, lab reports and 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 score will be given NOT QUALIFIED (NQ) to appear 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. Balagurusamy, E. (2016). *Programming In Ansi C.* Tata McGraw-Hill.
2. Gottfried, Byron S. (2018). *Programming with C.* Tata McGraw-Hill.

References:

1. Kelley A. & Pohl I.(2001). *A Book on C, Programming in C.* Addison-Wesley.
2. Kernighan, B. W., & Ritchie, D. M. (2002). *The C programming language.* Prentice Hall.