Curriculum Structure 2023
Bachelor in Computer Application

SN	Course Code	Subject	C	L	P
	Year 1	**Semester 1**			
1	ENG 121	English for IT Professional	3	3	1
2	MTH 131	Mathematics I	3	3	0
3	ELX 111	Digital Logic System	3	3	1
4	CMP 116	Computer Fundamental and Application	3	3	2
5	CMP 117	Programming Logic and Technique	3	3	0
6	CMP 111	Computer Application Workshop	1	0	3
		Sub Total	**16**	**15**	**7**
	Year 1	**Semester 2**			
7	ENG 122	Business and Technical Communication	3	3	1
8	MTH 132	Mathematics II	3	3	0
9	ACC 131	Financial Accounting	3	3	1
10	CMP 118	Programming in C	3	3	3
11	ELX 112	Microprocessor and Computer Architecture	3	3	1
12	PRJ 151	Project I	1	1	2
		Sub Total	**16**	**16**	**8**
SN	**Course Code**	**Subject**	**C**	**L**	**P**
	Year 2	**Semester 3**			
13	CMP 215	Object Oriented Programming using Java	3	3	3
14	CMP 227	Data Structure and Algorithms	3	3	2
15	CMP 221	System Analysis and Project Management	3	3	1
16	CMP 380	Web Technologies I	3	3	3
17	CMP 230	Operating System	3	3	1
		Sub Total	**15**	**15**	**10**
	Year 2	**Semester 4**			
18	CMP 323	Software Engineering	3	3	1
19	CMP 226	Database Management System	3	3	3
20	CMP 242	Computer Graphics and Multimedia Technology	3	3	2
21	MTH 320	Probability and Statistics	3	3	0
22	CMP 402	Web Technologies II	3	3	3
23	PRJ 251	Project II	2	1	3
		Sub Total	**17**	**16**	**12**
SN	**Course Code**	**Subject**	**C**	**L**	**P**
	Year 3	**Semester 5**			
24	MTH 230	Numerical Methods	3	3	1
25	CMP 317	DotNet Technology	3	3	3
26	CMP 336	Data Communication and Computer Networks	3	3	2

27	ELE 322	Research Methodology	3	3	1
28	MTH 330	Mathematical Foundation in Computer Application	3	3	0
		Sub Total	**15**	**15**	**7**
	Year 3	**Semester 6**			
29	CMP 316	Data Science and Analytics	3	3	2
30	CMP 314	Management Information System	3	3	0
31	CMP 350	Simulation and Modeling	3	3	1
32	MGT 322	Organization Management	3	3	0
33		Elective I	2	3	0
34	PRJ 351	Project III	3	1	3
		Sub Total	**17**	**16**	**6**
SN	**Course Code**	**Subject**	**C**	**L**	**P**
	Year 4	**Semester 7**			
35	CMP 401	Cyber Law and Professional Ethics	3	3	1
36	CMP 404	Mobile Application Development Technology	3	3	3
37	ECO 311	Applied Economics	3	3	0
38	INT 461	Internship	3	1	0
39		Elective II	3	3	0
		Sub Total	**15**	**13**	**4**
	Year 4	**Semester 8**			
40	CMP 415	Cloud Computing	3	3	2
41	CMP 416	Digital Economy	3	3	2
42		Elective III	3	3	0
43	PRJ 451	Project IV	5	1	3
		Sub Total	**14**	**10**	**7**
		TOTAL	**125**		

Pokhara University
Faculty of Science and Technology

Course Code: ENG 121 (3 Credits)		Full Marks: 100
Course Title: **English for IT Professional(3-3-1)**		Pass Mark: 45
Nature of the Course: Theory + Practical		Total Lectures: 48 hours
Level: Bachelor	Year : I / Semester : I	Program: Bachelor of Computer Application

1. Course Description:

This course is designed to cover grammar, vocabulary, reading and writing components. Grammar incorporates the essential aspects of technical English usage in context. Vocabulary covers words from different academic fields. The reading component deals with a wide variety of carefully selected materials. They include, among other things, informative passages on contemporary and critical issues. The writing part includes materials geared to developing various writing skills required for effective communication on matters of general and academic interests.

2. General Objectives:

The general objectives of this course are as follows:

a) To enhance the students' understanding of the grammatical system and its uses,
b) To expand the students' repertoire of general and academic vocabulary,
c) To familiarize students with the techniques of using vocabulary,
d) To develop in students an ability to comprehend and interpret different kinds of written texts by exposing them to a wide variety of authentic contemporary text materials,
e) To develop in students, the different kinds of writing skills needed for effective communication on matters of general and academic interests.

3. Methods of Instructions:

- Lecture and discussion
- Demonstration
- Presentation
- Explanation and illustration
- Group and individual work
- Project work
- Self-study etc.

It is expected that students are fully engaged with the teacher in subject matters and lessons to excel their interactive and presentation skills.

4. Course Contents

Specific Objectives	**Course Content**
Unit 1: Foundation of English Language	**8 hrs**
• Explain the basic concept of structures of English and academic language, focus on grammatical structures and developing vocabularies	1.1 Introduction to English Language 1.2 Academic English Basics 1.3 Parts of Speech 1.4 Word formation: prefixes and suffixes 1.5 Synonyms and antonyms

	1.6 Phrasal verbs 1.7 Sentence structures: Simple, compound and complex 1.8 Voice- Active and passive voice 1.9 Verities of English
Unit 2: English Language Terminology	**6 hrs**
• Describe the role of appropriate tenses, prepositions and articles • Explain the use of supply the appropriate words	2.1 Tense 2.2 Prepositions 2.3 Article 2.4 Word order 2.5 Countable and uncountable nouns 2.6 Conjunctions, relative clauses
Unit 3: Review and Critical Thinking	**5 hrs**
• Explain the role of critical analysis, reviews orally	Reviews and critical thinking
Unit 4: General Communication Skills	**5 hrs**
• Describe the objectives and characteristics of Communication • Explain the ways of develop effective presentation strategies	4.1. Listening and speaking 4.2. Active listening 4.3. Effective speaking 4.4. Effective presentation strategy 4.5. Group communication
Unit 5: Reading and Writing	**3 hrs**
• Compare the ideas for reading texts and extract general ideas	5.1 Reading and its techniques 5.2 Technical writing 5.3 Paragraphs and essays
Unit 6: Leading and Misleading of Technical Communication	**5 hrs**
Explain for the workplace; Style in technical prose	6.1 Ethics at work 6.1.1 Ethics at framework 6.1.2 Ethics for students 6.1.3 How is ethics related to technical communication? 6.2 Writing ethically
Unit 7: The Essentials of Writing	**10 hrs**
• Describe the various technical essays • Explain the characteristics of writing	7.1 The Structure of an Essay 7.1.1 The Introduction 7.1.2 Body Paragraphs 7.1.3 The conclusion 7.2 The writing processes
Unit 8: Reading Comprehension and Summary Writing	**6 hrs**
• Explore on language construction and	8.1 Note taking, summary and precis writing. 8.2 Computer makes the world smarter and

find the theme	smaller (Nancy Stern & Robert A. Stern) 8.3 How to become a programming expert? (Erich H. & John Ewan) 8.4 The Future of Information Technology (Erich H. & John Ewan).

*Note: To make class more practical oriented, teachers are requested to devise appropriate tutorials with practical work***.**

5. Evaluation System and Students' Responsibilities

Evaluation System

The internal evaluation of a student may consist of assignments, attendance, term-exams, lab reports and projects etc. The tabular presentation of the internal evaluation is as follows:

Internal Evaluation	**Weight**	**Marks**	**External Evaluation**	**Marks**
Theory		**30**	**Semester End**	**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				

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. Oxford English for Information Technology (Erich H. & John Ewan)

Reference Books:

1. Swan, M. (2005). *Practical English Usage* (Vol. 688). Oxford: Oxford university press.
2. Gardner, P., 2005. *New directions: Reading, writing, and critical thinking*. Cambridge

University Press.

3. Graves, H. and Graves, R., 2012. *A Strategic Guide to Technical Communication-(US)*. Broadview Press.

Pokhara University
Faculty of Science and Technology

Course Code: MTH 131 (3 Credits)		Full Marks: 100
Course Title: **Mathematics I (3-3-0)**		Pass Mark: 45
Nature of the Course: Theory		Total Lectures: 48 hours
Level: Bachelor	Year : I / Semester : I	Program: Bachelor of Computer Application

1. Course Description:

This course includes set theory and real number system, functions and graphs, differential calculus and its application, partial derivative and its application, symbolic logics, matrices and determinants which are essential as mathematical foundation for computing.

2. General Objectives:

The general objective of this course is to provide the students with basic mathematical skills required to understand Computer Application Courses.

3. Methods of Instructions:

Lecture, Tutorial, Discussion, Assignments and Practical works.

4. Contents in Detail

Specific Objectives	**Contents**
• Solve problems related to sets, Absolute value and intervals.	**Unit T: Set Theory and Real Number (6 Hrs.)** 1.1 Introduction 1.2 Types of sets 1.3 Operations on Sets (Union, Intersection, Difference, Complement) 1.4 Venn diagram 1.5 Cardinal Number of Set and Problems Related to Sets 1.6 Real Number 1.6.1 Types of real numbers 1.7 Absolute value of real numbers 1..8 Open and closed intervals
• Solve the problem related to functions and graph • Identify and plot the curves using any software tools (MATLAB/ Octave/ Mathematica, etc.) • Solve inequality and draw in graph.	**Unit 2: Functions and Graphs (5 Hrs)** 2.1 Definition of function 2.2 Domain and Range of a function 2.3 Inverse function 2.4 Type of functions : Constant, Algebraic (linear, Quadratic, Cubic), Trigonometric, exponential and Logarithmic and their graphs. Composite function. 2.5 Linear inequality and their graph (up to two variables)
• Solve the problems related to limit,	Unit 3: **Limits & Continuity of Functions (5 Hrs.)**

continuity, discontinuity, demand and profit function.	3.1 Introduction 3.2 Limit of functions 3.3 Continuity & discontinuity of functions 3.4 Demand & Profit function
• Solve the problems related to derivative and its aplications. • For visualization of curve use any software tools (MATLAB/ Octave/ Mathematica, etc)	Unit 4: **Derivative and it's application (10 Hrs.)** 4.1 Introduction 4.2 Techniques of differentiation 4.3 Derivative of algebraic, exponential, logarithmic & simple trigonometric functions. 4.4 Higher order derivative (up to 3rd order) 4.5 Application of derivative 4.5.1 Technique of finding limits (using L-hospital rule) 4.5.2 Increasing & decreasing function 4.5.3 Maxima & minima of function of one variable 4.5.4 Concavity of the function 4.5.5 Inflection point 4.6 Average cost & Marginal cost 4.6.1 Average revenue & marginal revenue 4.6.2 Profit maximization under perfect competition 4.6.3 Profit maximization under monopoly
Find partial derivatives and its applications.	**Unit 5: Multivariable functions and Partial Derivatives (8 Hrs.)** 5.1 Introduction 5.2 Partial derivatives 5.3 Homogeneous function 5.4 Euler's theorem 5.5 Extreme values and saddle points for multivariable functions. 5.6 Lagrange multipliers
Solve the problems related to Logics.	**Unit 6: Symbolic Logics (6 Hrs.)** 6.1 Introduction 6.2 Statements 6.3 Logical connectives 6.4 Conjunction, Disjunction, Negation, conditional or Implication, Bi-conditional 6.5 Logical equivalence 6.6 Negation of compound events 6.7 Tautology & contradiction 6.8 Mathematical induction
• Solve the problems related to matrices and determinants. • Operation of matrices and determinants by using any software tools (MATLAB/ Octave/	**Unit 7: Matrices and Determinant (8 Hrs.)** 7.1 Introductions of Matrices 7.2 Types of Matrices 7.3 Equality of Matrices 7.4 Algebra of Matrices 7.5 Transpose, Minors and Cofactors of Matrix 7.6 Determinant

Mathematical, etc.)	7.7 Properties of determinants 7.8 Inverse of matrices 7.9 Singular and non-singular matrix 7.10 Solution of Linear equation by matrix inversion method, Cramer's rule, Gauss elimination method

5. List of Tutorials	
SN	**Tutorials**
1.	Problem of sets, absolute value and intervals.
2.	Function and graph, domain and range, linear inequality and their graph.
3.	Limits, continuity, demand and profit function.
4	Derivative of algebraic, exponential, logarithmic & simple trigonometric functions. Higher order derivative, Increasing & decreasing function, Maxima & minima of function of one variable, Average cost & Marginal cost, Profit maximization under perfect competition, Profit maximization under monopoly.
5	Partial derivative, Verification of Euler's theorem, Extreme values and saddle points for multivariable functions, Lagrange multipliers.
6	Exercise for each topic, Mathematical induction.
7	Algebra of Matrices, Transpose, Minors and Cofactors of Matrices, Evaluation of determinant by using properties, Inverse of matrices, Solution of Linear equations.

6. List of Practical by using MATLAB/Mathematica/Octave/other software tools	
1.	Visualization of Functions and Graphs
2.	Visualization of increasing and decreasing functions, maxima and minima
3.	Visualization of Multivariable functions
4.	Operation on Matrices and Determinant
5.	Solution of Linear equation by matrix inversion method.

7. Evaluation System and Students' Responsibilities

Evaluation System

The internal evaluation of a student may consist of assignments, attendance, term-exams, lab reports and projects etc. The tabular presentation of the internal evaluation is as follows:

Internal Evaluation	**Weight**	**Marks**	**External Evaluation**	**Marks**
Theory		**50**	**Semester End**	**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 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. Budnick F.S. ***Applied Mathematics for Business Economics and the Social Sciences*** McGraw-Hill Ryerson Limited
2. Thomas and Finney ***Calculus*** Pearson

References:

1. D.R. Bajracharya, R.M. Shrestha &et al**, *Basic Mathematics I, II*,** Sukunda Pustak Bhawan, Nepal
2. Rudra Pratap ***Getting Started with MATLAB***, Oxford University Press 2010
3. Yamane,Taro: ***Mathematics for Economist***, Prentice Hall of India
4. K.K. Shrestha & R. K. Thagurathi, ***Applied Mathematics***

Pokhara University **Faculty of Science and Technology**		
Course Code: ELX 111 (3 Credits)		Full Marks: 100
Course Title: **Digital Logic System (3-3-1)**		Pass Mark: 45
Nature of the Course: Theory + Practical		Total Lectures: 48 hours
Level: Bachelor	Year : I / Semester : I	Program: Bachelor of Computer Application

1. Course Description

This course is designed to covers the concepts of digital logic systems. This course includes the role of information representation, fundamental concept of computing, principles and properties of Boolean algebra and its application in simplification, circuit analysis and gate implementation. It also covers the use of flip flops in the design of synchronous and asynchronous sequential logic circuits. Finally, students will be capable to study functional units of simple computer ALU. Group discussion, lab works, simulation through software tools and implementation of grouped project are major instructional approach.

2. General Objectives

The course is designed with the following general objective:

- To provide basic knowledge of logic systems.
- To familiarize student with the basic tools to study various digital circuits.
- To develop the skill among the student about to analyze the problem and develop concept for software development.

3. Methods of Instruction

- Lecture
- Tutorial
- Laboratory work
- Group discussion
- Grouped project work

4. Contents in Detail

Specific Objectives	**Contents**
• Explain the number representation in different number systems and their conversions • Identify the method of computing and role of complements in subtraction • Compare among binary codes, alphabets and other characters in secure computing.	**Unit 1 : Number system and Coding (8 hrs)** 1.1 Analog and digital system- introduction and advantages 1.2 Digital computer and numerical representation 1.3 Number system 1.3.1 Decimal 1.3.2 Binary 1.3.3 Octal 1.3.4 Hexadecimal 1.4 Conversion among binary, octal, decimal and Hexadecimal number systems

	1.5 Complements (Radix and diminished radix for binary and decimal only) 1.6 Subtraction using complements 1.7 Codes 1.7.1 Weighted codes (BCD, 84-2-1, 2421) 1.7.2 Non weighted codes (Excess 3 , Gray) 1.8 Alphanumeric and instruction codes
• Interpret facts, theorems and principles based on Boolean algebra. • Emphasize on logic gates, Boolean function and truth tables and gate implementation.	**Unit 2 : Boolean algebra and Logic gates (5 hrs)** 2.1 Basic definition and properties of Boolean algebra 2.2 Postulates and theorems of Boolean algebra 2.3 Logic gates, truth tables and Boolean function 2.4 Duality principle and complements 2.5 Implementation of NAND and NOR gates
• Explain the Boolean function simplification methods • Identify role of don't care conditions while simplifying Boolean functions.	**Unit 3 : Simplification of Boolean functions (6 hrs)** 3.1 Venn diagram 3.2 Canonical forms and standard forms 3.3 Karnaugh map up to four variables 3.4 Don't care conditions 3.5 Simplification in SOP and POS using K-map
• Describe how to design various combinational logic circuits • Describe internal logic diagram, operation and circuit implementation	**Unit 5 : Combinational logic (10 hrs)** 4.1 Adders and Subtractors 4.3 Code conversion 4.4 Analysis procedure 4.5 NAND and NOR implementation 4.6 Parity generation and checking 4.7 Binary Parallel adder and subtractor 4.8 BCD adder 4.9 Decoder and encoder 4.10 Multiplexer and demultiplexer 4.11 ROM and PLA
• Explain the type of flip flops with their role in sequential circuits • Identify the key characters of each flip-flop for Implementation	**Unit 5 : Sequential logic (6 hrs)** 5.1 Block diagram of sequential circuit 5.2 Flip flop (RS, D, JK, T, Master-slave), truth table, excitation table and characteristic equation 5.3 Triggering of flip flops 5.4 State diagram and state table

• Describe about registers, shift registers and types with timing sequences • Design synchronous, asynchronous and Mod counters	**Unit 6 : Registers and Counters (5 hrs)** 6.1 Register, Register with parallel load 6.2 Shift register and types of shift register 6.3 Ripple counters (Binary and BCD counter) 6.4 Up-down counter, Mod-6 counter, Johnson
• Describe about read and write operation in RAM • Describe about the processor unit and its diagram	**Unit 7 : Functional Units of Memory and ALU (8 hrs)** 7.1 Random access memory 7.2 Arithmetic logic unit 7.3 Status register 7.4 Processor unit

5. Laboratory work:

- Familiarization with logic gates.
- Familiarization with Boolean functions.
- Design of simple combinational circuits.
- Adder and subtractor
- Encoder and decoder
- Multiplexer and demultiplexer
- Demonstration of working principles of flip flops.
  Experiments are demonstrative either in trainer Kit or Simulation using appropriate software.

6. Evaluation system and Students' Responsibilities

Internal Evaluation

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

Internal Evaluation	**Weight**	**Marks**	**External Evaluation**	**Marks**
Theory		**30**	**Semester End**	**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 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 QUILIFIED (NQ) and the student will not be eligible to appear 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 re-exam.

7. Prescribed Books and References

Text Book

1. M. Morris Mano, *Digital Logic and Computer Design* Pearson India, 2017.

Reference Book

1. M. Morris Mano, *Digital Design* California State University, Los Angeles.
2. A. Anand Kumar, *Fundamental of Digital Circuits,* Fourth Edition, PHI Learning Private Limited, 2016.

Pokhara University Faculty of Science and Technology		
Course Code: CMP 116 (3 Credits)		Full Marks: 100
Course Title: **Computer Fundamental and Application (3-3-2)**		Pass Mark: 45
Nature of the Course: Theory + Practical		Total Lectures: 48 hours
Level: Bachelor	Year : I / Semester : I	Program: Bachelor of Computer Application

1. Course Description

This course is designed to conceptualize the computer fundamental knowledge and its application to solve the real-life problems. This course covers the concept of computer, computer organization, software, hardware, functional behavior of peripherals and important accessories, data communication, computer networks and burning technologies. In this syllabus unit I and II covers the fundamental concept of computer; unit III and VII covers the software, unit IV and V covers the hardware while unit VI and VIII covers the data sharing tools and trends. The course includes theoretical and practical implementation of software tools and applications.

2. Course Objectives

The general course objectives of this course are outlined as:

- To familiarize students with fundamental knowledge about computer system.
- To make the students understand software, hardware and their working procedure.
- To enhance students' knowledge about various components of computer.
- To train students about assembling and disassembling, general maintenance and networking of computers.

3. Methods of Instruction

The medium of instruction is English and the faculty members can apply the various tools and techniques for teaching methodologies. As per the student’s ability subject teachers/faculties can use various teaching methodologies like Class Room/Lecture-based, Discussion-based, Project-based Learning(PBL), Problem-Based Learning, Flipped Classroom-based, Active Learning, Socratic Method, Cooperative Learning, Experimental Learning, Gamification, Inquiry-based Learning, Constructivist Approach, Collaborative Learning, Direct Instruction, Differentiated Instruction, Montessori Method, Reggio Emilia Approach, Waldorf Education, Peer Teaching etc. are called alternative method of teaching to motivate the students for learning. After completion of each unit, faculty members can evaluate the students theoretically and practically. They can conduct VIVA, Supervised test, Questionnaire test, Assignment test, Project work and Practical work, Terminal examination as per the requirement.

4. Course Contents

Specific Unit wise Objective	**Course Contents**
Unit 1: Introduction to Computer	**5 hours**
1) Explain the conceptual knowledge of computer with historical and generation background 2) Describe the types of computer with characteristics and applications	1.1 Definition 1.2 History of computer 1.3 Generation of computer 1.4 Types of Computer 1.5 Characteristics of computer 1.6 Applications of computer
Unit 2: Basic Organization of Computer	**4 hours**

1) Explain the organization of computer with their functional components 2) Describe the working mechanism of computer system	2.1 Basic function of computer 2.2 Basic functional organization of computer 2.2.1 Input unit 2.2.2 Output unit 2.2.3 Storage unit 2.2.4 Arithmetic and logic unit 2.2.5 Control unit 2.2.6 Central Processing Unit (CPU) 2.3 The system concept
Unit 3: Computer Software	**6 hours**
1) Describe the concept of computer software with their roles 2) Explain the software architectures their application	3.1 Introduction to software 3.2 Logical system architecture 3.3 Types of software 3.3.1 System Software 3.3.2 Application Software 3.3.3 Utility Software 3.4 Firmware 3.6 Middleware
Unit 4: Computer Accessories	**9 hours**
1) Familiarized with input and output accessories with their application 2) Identify the role of agronomical design in related devices	4.1 The Input Accessories 4.1.1 Keyboard Devices 4.1.2 Point and Draw Devices 4.1.3 Yoke 4.1.4 Data Scanning Devices 4.1.5 Digitizer 4.1.6 Microphone 4.1.7 Electronic Cards Based Devices 4.1.8 Speech Recognition Devices 4.1.9 Vision Based Devices 4.2 The Output Accessories 4.2.1 Monitor 4.2.2 Printer 4.2.3 Plotter 4.2.4 GPS 4.2.5 Projectors 4.2.6 Headphones 4.2.7 Soundcard/video Card 4.2.8 Speaker 4.2.9 Voice Response System 4.2.10 Computer Output Microfilm (COM) 4.2.11 SGD Speech Generation Device 4.3 Ergonomically Designed Devices
Unit 5: Storage Devices	**7 hours**
1) Describe the characters of primary and secondary storage devices 2) Explain the function and behavior of different types of memories .	5.1 Introduction 5.2 Types of Storage Devices 5.3 Cache memory 5.4 Registers 5.5 Primary Memory 5.5.1 Introduction 5.5.2 Characteristics of main memory 5.5.3 Types of primary memory 5.5.4 Difference between RAM and ROM, DRAM and SRAM, primary and secondary memory

	5.6 Secondary Memory 5.6.1 Introduction 5.6.2 Characteristics of secondary memory 5.6.3 Types of Secondary memory 5.6.4 Difference between Hard disk and SSD, magnetic and optical disk, direct and sequential access magnetic memory
Unit 6: Data Communication and Computer Network	**5 hours**
1) Interpret the method of data communication including transmission modes and media 2) Explain the relationship of computer network and topologies with their advantages and disadvantages 3) Compare the role of email and internet in modern life	6.1 Data Communication 6.1.1 Introduction 6.1.2 Basic Elements of a communication system 6.1.3 Data Transmission Modes (Simplex, Half Duplex, Full Duplex) 6.1.4 Data transmission Speed (Narrowband, Voice band, Broadband) 6.1.5 Data Transmission Media (Twisted-pair wire, coaxial cable, Optical fibers, Microwave system Communication satellite) 6.2 Computer Network 6.2.1 Definition 6.2.2 Types of Computer Network (PAN, LAN, CAN, MAN and WAN), Differences, advantages disadvantages 6.2.3 Network Topologies, advantages, disadvantages 6.2.4 Intranet, Extranet, Internet 6.2.5 E-mail
Unit 7: Operating System	**6 hours**
1) Explain the features of operating system and its historical background 2) Describe the functional and architecture behaviors of various type of OS. 3) Explain the different installation methods for operating system	7.1 Introduction 7.2 History and evolution of OS 7.3 Objective of OS 7.4 Generation of OS 7.5 Functions of OS 7.6 Types of OS 7.7 System architecture of OS 7.8 Different between GUI and TUI/CUI 7.9 The booting system 7.10 Windows/ Linux operating system, settings, properties and installation guides
Unit 8: AI and Emerging Technologies	**6 hours**
1. Familiarize with Artificial Intelligence and its applications. 2. Explain about emerging trends of technologies.	8.1 Introduction to AI 8.2 AI and its Applications 8.2.1 Natural Language Processing 8.2.2 Machine vision 8.2.3 Expert system 8.3 Machine Learning 8.4 Neural Networks 8.5 Blockchain technology 8.6 IoT 8.7 Cloud Computing 8.8 Cyber Security

5. Laboratory Work

A. Office Automation

a) **Word-processing**

1. Basic options of word-processing for typing, editing, formatting, margin setting, viewing, designing, printing a document.
2. Creating, insetting, editing, formulating table for word-processing
3. Document preparation with table, figure, page number, margin setting etc. and printing document as a report submission.

b) **Spreadsheet**

1. Preparing sheet for data processing like, arithmetic, logical and other types of functional operation
2. Preparing data table for calculation, analysis and creating various charts for presentation.
3. Inserting picture, table, graphs into word-processing
4. Printing documents after proper setting into the required format.

c) **Presentation**

1. Creating various types of slides with master slide for presentation
2. Setting the slide into the required format.

B. **Email and Internet**

1. Setting various kinds of email account and using them for personal and group purpose
2. Uploading and downloading the information form internet.

6. Evaluation System and Students' Responsibilities

Evaluation System

The internal evaluation of a student may consist of assignments, attendance, term-exams, lab reports and projects etc. The tabular presentation of the internal evaluation is as follows:

Internal Evaluation	**Weight**	**Marks**	**External Evaluation**	**Marks**
Theory		**30**	**Semester End**	**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				

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.

7. Prescribed Books and References

Text Books

1. P.K. and Priti Sinha*, Foundations of Computing*, BPB Publications, Third Edition

References Books

1. A Text Book of Computer Fundamental and Application, OCEM Publication, (Second Edition), by Hari Bhandari
2. Computer Fundamentals, BPB Publications by V.K Jain
3. Fundamental of Computers, by Balagurusamy E., New Delhi: Tata McGraw Hill.

Pokhara University **Faculty of Science and Technology**		
Course Code: CMP 117 (3 Credits)		Full Marks: 100
Course Title: **Programming Logic and Technique (3-3-0)**		Pass Mark: 45
Nature of the Course: Theory + Practical		Total Lectures: 48 hours
Level: Bachelor	Year : I / Semester : I	Program: Bachelor of Computer Application

1. **Course Description**
   This course is designed to provide students with a solid foundation in programming logic and techniques, emphasizing the fundamental principles necessary for effective problem-solving and algorithmic thinking. This course focuses on basics of programming fundamentals and languages, program development schemas and models, and maintenance strategies. By the end of this course, students will gain understanding of programming logic and techniques, enabling them to approach complex problems systematically, design effective algorithms, and implement solutions to real-time problems.

2. **Course Objectives**
   This course is designed with the following general objectives:
   - To familiarize the students with the concept of fundamental problem analysis, modeling and coding techniques.
   - To develop the skill about programs, programing and various programming languages.
   - To promote the knowledge about the phases of program development and its methodologies.
   - To acquaint the student with create and innovate the ideas, skills using specific tools and techniques.
   - To make the student able to develop a plan and procedure for a standardized program, minimizing errors and budget.
   - To provide the concept about programming in client server, web server and mobile, cloud.

3. **Methods of Instruction**
   The medium of instruction is English and the faculty members can apply the various tools and techniques for teaching methodologies. As per the student's ability faculties have their rights to use the teaching methodologies like Class Room/Lecture-based, Discussion-based, Project-based learning(PBL), Problem-based Learning (PBL), Flipped Classroom-based, Active Learning, Socratic Method, Cooperative Learning, Experimental Learning, Gamification, Inquiry-based Learning, Constructivist Approach, Collaborative Learning, Direct Instruction, Differentiated Instruction, Montessori Method, Reggio Emilia Approach, Waldorf Education, Peer Teaching etc. are called alternative method of teaching to motivate the students for learning. After completion of each unit, faculty members can evaluate the students theoretically and practically if exists. They can conduct VIVA, Supersized test, Questionnaire test, Assignment test, Project work and Practical work, even Terminal examination as per the requirement.

4. **Course Contents**

Specific Unit wise Objective	**Course Contents**
Unit 1: Programming Languages	**14 hours**

1) Describe the computer languages, generation and their types with their features 2) Explain the program translation processes from a source program to object/executable file. 3) Interrupt the loading, linking and relocating the program into the memory while compilation.	1.1 Introduction 1.2 Analogy with Natural Language 1.3 Classification of Computer Language (Low Level Language and High Level Language) 1.3.1 Low Level Languages, (Machine and Assembly Language) 1.3.2 Advantages and Disadvantages of Machine Level Language 1.3.3 Advantages of Assembly Language over machine Level 1.3.4 Disadvantages of Assembly Language 1.4 High Level Language 1.4.1 Advantages and Limitations of High Level Language 1.4.2 Difference between Low Level and High Level Language 1.4.3 Program Language Translator 1.4.3.1 Types of Translator (Compiler, Assembler and 1.4.3.2 Different among Translator 1.5 Linker and Loader 1.5.1 Types of Linder and Loader 1.6 Generation of Computer Languages 1.6.1 Introduction to 3GL and 4GL 1.6.2 Advantages of 4GL over 3GL 1.6.3 Feature of 4GL (Object Oriented Language) 1.7 Machines Independent and Portability of programs 1.8 Important types of High-Level Language
Unit 2: Program/ Software Development	**10 hours**
1) Describe the program development step and role of life cycle. 2) Explain the role of budget in a program development 3) Identify the role of programming tools for the good programming 4) Explain the role of creative, logical and analytical thoughts for requirement study and efficient program development	2.1 Introduction to program development/software development 2.2 Program Development Life Cycle. 2.2.1 Problem identification 2.2.2 Problem Analysis/Requirement Analysis 2.2.3 Program Design Phases (Data design, Input, Output, Procedure, File design) 2.2.4 Program coding 2.2.5 Program Testing and Debugging 2.2.6 Program Delivery/Implementation and Evaluation 2.2.7 Program Maintenance 2.2.8 Documentation 2.3 Basic Programming Tools 2.3.1 Algorithm 2.3.2 Flowchart 2.3.3 Decision Table 2.3.4 Pseudocode 2.4 Comparison of Programming Tools 2.5 Advantages and Disadvantage of Programming Tools 2.6 Example of Programming Tools
Unit 3: Programming Techniques	**8 hours**
1) Describe programming techniques and models 2) Compare programming techniques and model for easily identify the advantages and disadvantages 3) Compare cohesion and coupling with their types 4) Explain the internal relationship of modules for communication in programming.	3.1 Introduction 3.2 Type of Programming Technique (Approaches) 3.2.1 Modular, Top down, bottom up, Structure and Object Oriented approach 3.2.2 Advantages and Disadvantages 3.3 Program Development Methods (Models) 3.3.1 Types of Program Development Models (Water fall or Traditional model, Prototype or Transformation model, Spiral model, Iterative model, V-model, RAD model, Bog Bang model, Evolutionary models, Agile) 3.3.2 Advantage and Disadvantages of Waterfall, Prototype and Spiral Model (Simple Introduction to other Models) 3.4 Cohesion and coupling 3.4.1 Types of Cohesion and Coupling 3.4.2 Use of Cohesion and Coupling in Program Development 3.5 Logics o Program Development 3.5.1 Types of Logics (Sequential, Selection Iteration and Recursion) 3.5.2 Differentiation

	3.6 Communication between modules
Unit 4: Program Maintenance	**8 hours**
1) Describe the software maintenance, its cost issues and time factors 2) Compare the ratio of the budget for development and maintenance of the program. 3) Describe the role of program documentation and its standard 4) Explain the role of system flow and DFD in program design 5) Describe the requirement specification of the program and development procedures	4.1 Introduction 4.2 Types of Program maintenance (Corrective maintenance, Adoptive maintenance, Perfective maintenance, Emergency maintenance) 4.3 Problem Areas in Program Maintenance 4.4 Cost issues in software maintenance 4.5 Impact of software Errors 4.6 Program documentation and its standards 4.6.1 Requirements of Documentation 4.6.2 Importance of Documentation 4.6.3 Types of Documentation 4.7 Program Specification 4.8 System Flow Chart 4.8.1 Elements of System Flowcharts 4.8.2 Examples of System Flowchart 4.9 Data Flow Diagram 4.9.1 Element of DFD 4.9.2 Leveling the data flow diagram 4.9.3 Idea for Drawing DFD 4.9.4 Examples of DFD
Unit 5: Client Server and Web-based Programming Technology	**8 hours**
1) Explain computer and communication, stand-alone and client server computing, two and three-tier architecture of client server system, web-based programming 2) Describe the role of platform independent feature of software	5.1 Standalone Programming concept 5.1.1 Introduction 5.1.1 Advantage and limitations of Standalone Programming 5.2 Client Server Programming (CSP) 5.2.1 Introduction to Client server programming 5.2.2 Advantages and Disadvantages of CSP 5.2.3 Distinguish between Standalone and Client Server Programming 5.2.4 Client Server Programming Architecture 5.2.5 Some of the client server programming and their applications 5.5 Web-based Programming 5.5.1 Introduction to web, versions of web 5.5.2 Advantages and Disadvantages of web-based programs 5.5.3 Architecture of web-based programming 5.5.4 Web-based System and Programming 5.5.5 Some important web-based programing and their applications 5.6 Introduction to mobile computing languages and their application 5.7 Requirement of Platform-Independents system

5. Evaluation System and Students' Responsibilities

Evaluation System

The internal evaluation of a student may consist of assignments, attendance, term-exams, presentation etc. The tabular presentation of the internal evaluation is as follows:

Internal Evaluation	Weight	Marks	External Evaluation	Marks
Theory		**50**	**Semester End**	**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 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 Books

1. P.K. and Priti Sinha, *Foundations of Computing*, BPB Publications, Third Edition

References Books

1. Er. Hari Bhandari*, A Text Book of Programming Logics and techniques*, OCEM Publication, Second Edition
2. V.K Jain, *Computer Fundamentals*, BPB Publications

Pokhara University **Faculty of Science and Technology**		
Course Code: CMP 111 (1 Credit)		Full Marks: 100
Course Title: **Computer Application Workshop (1-0-3)**		Pass Mark: 45
Nature of the Course: Practical		Total Lectures: 30 hours
Level: Bachelor	Year : I / Semester : I	Program: Bachelor of Computer Application

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 to install/update operating system and various application software, manage data backup and restore operations on a computer system

2. Methods of Instruction

Specific instructional techniques: Lab works, Project works

3. Contents in Detail

Specific Objectives	**Contents**
- Recognize the computer hardware and accessories.	**Unit 1: Introduction to Desktop/Workstation Assembly (2)** 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 and laptop with power supply, internal cabling, Motherboard and PCI Device	**Unit 2: Assembling Desktop and Laptop Computer (4)** 2.1 Assembling steps and precautions. 2.2 Setting of motherboards, memory, hard disk, processors, CDROM, SMPS and other devices.
- Install OS (Linux and Windows) in Desktop/Laptop	**Unit 3: Laptop/Desktop OS Installation (6)** 3.1 BIOs setting, firmware types, BOOT configuration. 3.2 Hard disc partitioning. 3.3 OS installation (Linux, Windows). 3.4 User account management 3.5 Printer, Scanner installation

- Install and update application software and utility software. - Scan and remove viruses from computer system.	**Unit 4: Software Installation (5)** 4.1 Installation of application and utility software. 4.2 Update of firmware patches, 4.3 Anti-virus installation and scanning.
- Recognize computer networks-wired PAN, CAN, LAN and Wireless LAN configuration.	**Unit 5: Networking and Internet Setup (5)** 5.1 Introduction to Computer Network, Network topologies, wired and wireless networking media. 5.2 Network cabling, cable types and connectors. 5.3 IP Address setting, Sharing of Printer/ Network Devices 5.4 Basic wireless AP configuration.
- Study of server of Mail/Data/Domain/FTP. - Apply FTP/SCP Client tool to upload/download files to FTP server.	**Unit 6: Server Installation and Backup (3)** 6.1 Introduction to server. 6.2 Study of different server (Email, Data, Domain, FTP).
- Use the basic troubleshooting tools and utilities.	**Unit 7: Hardware and Software Troubleshooting (5)** 7.1 Basic Repair and Maintenance of Desktop/Laptop. 7.2 Replacement of Passive Components. 7.3 Uses of basic software troubleshooting commands. 7.4 Uses of trouble shooting tools and utilities.

4. Evaluation system and Students' Responsibilities

Evaluation System

The evaluation of a student may consist of assignments, attendance, test-exams, term-exams, lab reports and projects etc. The tabular presentation of the evaluation is as follows:

External Evaluation	**Marks**	**Internal Evaluation**	**Weight**	**Marks**
Semester-End examination	0	Attendance & Class Participation	5%	100
		Assignments/Lab Report	10%	
		Quizzes/MCQ	10%	
		Project Work/Lab Test	10%	
		Viva	15%	
		Internal Exam	50%	
	0	**Total Internal**		**100**
Full Marks: 0 + 100 = 100				

Student 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.

5. Prescribed Books and References

References

1. Rosch, W. R. *The Winn L. Rosch Hardware Bible*. USA: Sams.
2 Bose, A. K. *Hardware and Software of Personal Computers*. New Delhi: New Age International.
3. Rajaraman, V. and Adabala, N. *Fundamental of Computers*. New Delhi: PHI Learning.

Pokhara University
Faculty of Science and Technology

Course Code: ENG 122 (3 Credits)		Full Marks: 100
Course Title: **Business and Technical Communication (3-3-1)**		Pass Mark: 45
Nature of the Course: Theory and Practical		Total Lectures: 48 hours
Level: Bachelor	Year: I / Semester: II	Program: BCA

1. Course Description:

This course is designed for the foundations of communication through writing letters, memos, emails, preparing reports and proposals, seminar papers, structuring of essay and verbal and nonverbal communications. The course includes a practicum component in that students towards the end of the semester to submit a portfolio that includes a range of writing assignments such as technical correspondence and job application process. This course follows various teaching and learning methodologies specially, the seminar model of classroom teaching with discussions and presentations using modern tools and techniques. And it offers conceptual insight into the field, principles of communication design, style, grammar and various writing skills of general and academic interests.

2. General Objectives:

The general objectives of this course are as follows:

- To be able to comprehend and take notes after listening.
- To develop the ability to deliver technical knowledge orally in English.
- To fasten reading skills in technical and non-technical reading materials
- To be able to write proposals, reports, letters, description on technical talks.

3. Methods of Instructions:

- Lecture and discussion
- Demonstration
- Presentation
- Explanation and illustration
- Group and individual work
- Project work
- Self-study etc.

It is expected that students are fully engaged with the teacher in subject matters and lessons to excel their interactive and presentation skills.

4. Course Content in details	
Specific Objectives	**Course Content**
	Unit 1: Introduction to Communication 3 hrs.
• Explain the communication and its processes • Describe the principles of communication	• Definition of communication • Process of Communication • 7 C's Principles of communication
	Unit 2: Technical communication Process 8 hrs
• Define technical communication with	• Definition

its forms and types • Explain the role competency and contrast with its core competencies along with attitudes, values and personality. • Describe the skill sets for technical writers	• Evolution of technical communication • Forms of technical communication • Types of technical communication • Importance of technical communication • Definition of Competency • Attitudes, values and personality • Core competencies in technical communication • Skill sets for technical writers
	Unit 3: Technical communication process 5 hrs
• Describing the technical communication process at two different levels- document and collaboration processes.	• The Document Process: - ❖ Planning ❖ Writing the draft, ❖ Revising ❖ Delivery • The Collaboration Process ❖ The pros and cons of collaboration ❖ Collaborative writing process ❖ Strategies for effective collaborative writing
	Unit 4: Correspondence 8 hrs
• Comparing the techniques of writing letters to apply for job • Describe the role of correspondence in the form of memoranda and letters	**Correspondence** • **Memo writing** ❖ Meaning and definition of memo writing ❖ Purpose of memos ❖ Format of memo • **Letter Writing** ❖ Introduction ❖ Features of letter ❖ Types of letters ❖ Job application- CV ❖ Letter of inquiry ❖ Quotation letter ❖ Order Letter ❖ Claim Letter ❖ Adjustment Letter
	Unit 5: Interviews, Meetings and Minutes 7 hrs
• Highlight the nature and methods of effective job interviews, meeting and minutes	**Interview:** • Introduction • Effective techniques for interview • Interview and body language **Meeting:** • Introduction • Purposes of Meeting • Notice (format of Meeting) • Minutes of Meeting

	- Purpose of Minutes - Format of Minutes
	Unit 6: Technical Talk and Non-verbal Communication 3 hrs
- Explain the methods to develop the presentation skill and strategies - Describe the non-verbal communication and its forms	- Oral communication - Oral presentation - Types of non-verbal communication - Importance of non-verbal communication - Characteristics of effective presentation
	Unit 7: Writing Proposals and Reports 8 hrs
- Explain effective and efficient methods for writing the reports and proposals	**Proposal writing** - Introduction - Components of proposals: Title, Introduction, Statement of Problem, Literature Review, Methodology, Budgeting, Output - Types of proposal **Report Writing** - Introduction - Different types of Report - Elements of Report :Title, Acknowledgement, Abstract, Introduction, Methodology, Results/Findings, Conclusion and recommendation, References
	Unit 8: Real-Life Scenarios and Role-Playing 3 hrs
- Enhance the quality of life job interviews - Explain how to resolve IT related conflicts involving group discussions	- Mock job interviews - Negotiating contracts and agreements - Resolving IT-related conflicts - Group discussions on IT trends and innovations
	Unit 9: Grammar and Usage 3 hrs
- Describe grammatical items and transform the sentences as required. - Explain the varieties of English with focus on usage	- Phrases, clauses and sentences - Functional and structural division of sentences - Transformation of sentences - Voice- Active and Passive - Varieties of English

Evaluation System

The internal evaluation of a student may consist of assignments, attendance, term-exams, lab reports and projects etc. The tabular presentation of the internal evaluation is as follows:

5. Evaluation System and Students' Responsibilities

Evaluation System

The internal evaluation of a student may consist of assignments, attendance, term-exams, lab reports and projects etc. The tabular presentation of the internal evaluation is as follows:

Internal Evaluation	**Weight**	**Marks**	**External Evaluation**	**Marks**
Theory		**30**	**Semester End**	**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				

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.

8. Prescribed Books and References

Text Book:

1. Business Communication for Success, UNIVERSITY OF MINNESOTA LIBRARIES PUBLISHING EDITION, 2015, ISBN -13

References:

1. Adhikari, Dharma, and Upadhyaya, Phanindra*, Technical Communication*, Buddha Publication, 2012
2. Adhikari, Dharma, and Upadhyaya, Phanindra*, Technical Communication-II*, Buddha Publication

Pokhara University
Faculty of Science and Technology

Course Code: MTH 132 (3 credits)		Full Marks: 100
Course Title: **Mathematics II (3-3-0)**		Pass Mark: 45
Nature of the Course: Theory		Total Lectures: 48 hours
Level: Bachelor	Year: I / Semester: II	Program: Bachelor of Computer Application

1. Course Description:

This course covers fundamental of integrals, application of integration, differential equations, vector Space, complex numbers and function of complex variables, sequence and series and Fourier series which are essential as mathematical foundation for computing.

2. General Objectives:

The general objective of this course is to provide the students with basic mathematical skills required to understand Computer Application Courses

3. Methods of Instructions:

Lecture, Tutorial, Discussion, Assignments and Practical works.

4. Contents in Detail

Specific Objectives	Contents
Explain • Indefinite • Definite • Improper and • Double integration • Symbolic calculation of integration using any software tools (MATLAB/Mathematica/Octave etc.)	**Unit 1: Fundamental of integrals [10 Hrs]** 1.1 Introduction 1.2 Indefinite integrals 1.3 Techniques of integration 1.3.1 Integration by substitution 1.3.2 Integration by parts 1.3.3 Integration by partial fractions 1.4 Definite integrals 1.5 Improper integrals 1.6 Beta and Gamma functions 1.7 Double integral (concept only)
• Evaluate area and volume by integration	**Unit 2: Application of integration [7Hrs]** 2.1 Introduction 2.2 Application in economics (Determination of total cost and total revenue function) 2.3 Area between the curves 2.4 Arc length of curves 2.5 Volume of solid of revolution (Disks and Washers) 2.6 Area of surface of revolution, 2.7 Consumer's surplus and producer's surplus
• Solve first and second order	**Unit 3: Differential equations [7 Hrs]**

differential equations.	3.1 Introduction 3.2 Order and degree of ordinary differential equations. 3.3 Solution of differential equations of first order by 3.3.1 Separation of variables 3.3.2 Homogeneous 3.3.3 Linear 3.3.4 Equation reducible to linear form (Bernoulli's equation) 3.3.5 Linear and exact differential equations 3.4 Second order homogenous ODE with constant coefficients. 3.5 Second order Non homogenous ODE (Concept only)
• Solve the problem related to Vector spaces, subspaces, linear dependent and independent, and Eigen values and Eigen vectors	**Unit 4: Vector Space [6 Hrs]** 4.1 Introduction 4.2 Vector spaces and subspaces with example 4.3 Linear combination of vectors 4.4 Linear dependence and independence of vectors 4.5 Basis and dimension of vector space 4.6 Eigen values and Eigen vectors.
• Solve and analyze complex number related problems	**Unit 5: Complex numbers and Function of complex variables [7Hrs]** 5.1 Introduction 5.2 Algebra of the complex numbers 5.3 Geometric representation of complex numbers 5.4 Conjugate and absolute values of complex numbers 5.5 Polar form of complex numbers 5.6 Complex variables and function of complex variables 5.7 Analytic functions 5.8 Necessary and sufficient conditions for f(z) to be analytic (without proof) 5.9 Harmonic functions 5.10 Conformal mappings
• Find Sum of series • Expand function in series	**Unit 6: Sequence and series [6 Hrs]** 6.1 Introduction 6.2 Arithmetic and Geometric series 6.3 Sum of finite natural numbers 6.4 Sum of square of first 'n' natural numbers 6.5 Sum of cubes of first 'n' natural numbers. 6.6 Convergence of geometric series 6.7 Taylor series (one and two variables) 6.8 Maclaurin series.

Find Fourier series, Fourier sine and cosine series	**Unit 7: Fourier series [5 Hrs]** 7.1 Introduction 7.2 Periodic functions and trigonometric series 7.3 Fourier series 7.4 Fourier sine and cosine series

5. List of Tutorials	
SN	**Tutorials**
1.	Problems of indefinite, definite and improper integration.
2.	Area, volume, Consumer's surplus and producer's surplus.
3.	Solution of first and second order differential equations.
4	Problem related to Vector spaces, subspaces, linear dependent and independent, and Eigen values and Eigen vectors.
5	Solve complex numbers & complex function related problems
6	Sum of series and expansion of function in series
7	Fourier series, Fourier sine and cosine series

6. List of Practical's by using MATLAB/Mathematica /other software tools	
SN	**Practicals**
1.	Integration by symbolic calculations
2.	Visualization of figure for surface area and volume.

7. Evaluation System and Students' Responsibilities

Evaluation System

The internal evaluation of a student may consist of assignments, attendance, presentation, term-exams etc. The tabular presentation of the internal evaluation is as follows:

Internal Evaluation	**Weight**	**Marks**	**External Evaluation**	**Marks**
Theory		**50**	**Semester End**	**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 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. Erwin Kreyszig ***Advance engineering Mathematics,***
2. Thomas and Finney ***Calculus*** Pearson

References:

1. D.R. Bajracharya, R.M. Shrestha &et al, ***Basic mathematics I, II***, Sukunda Pustak Bhawan, Nepal
2. Budnick F.S. ***Applied Mathematics for Business Economics and the Social sciences*** MCGraw-Hill Ryerson Limited
3. K.K. Shrestha & R. K. Thagurathi, Applied Mathematics
4. Rudra Pratap ***Getting Started with MATLAB***, Oxford University Press 2010

Pokhara University
Faculty of Science and Technology

Course Code: ACC 131 (3 Credit)		Full Marks: 100
Course Title: **Financial Accounting (3-3-1)**		Pass Mark: 45
Nature of the Course: Theory and Practice		Total Lectures: 48 hours
Level: Bachelor	Year: I / Semester: II	Program: Bachelor of Computer Application

1. Course Description:

This course aims to explore the fundamental principles and practices of Financial Accounting. This comprehensive course covers the entire accounting cycle, including inventory management and bank reconciliation statements. In this course, students will learn in-depth about journals, ledgers, and trial balances; the accounting equation; income statements, statements of retained earnings, and balance sheets; cash flow statements; inventory management; and bank reconciliation statements. By the end of this course, students will have a strong grasp of financial accounting concepts and will be well-equipped to analyze financial statements, make informed business decisions, and pursue further studies in accounting or related fields. Whether students aspire to become accountants, entrepreneurs, or simply have an interest in financial literacy and software development, this course will provide them with valuable skills for the business world. In summary, this course provides students with a solid foundation in the aspects of financial reporting and analysis.

2. General Objectives:

Studying Financial Accounting typically involves several general objectives that help students gain a comprehensive understanding of financial reporting and analysis. This course is designed with the following general objectives:

- To make students able to understand financial transactions.
- To enable the students to prepare and interpret key financial statement.
- To make students familiar with accounting standards such as GAAP, IFRS, NAS and so on.
- To acquaint the students with the various types of activities conducted in an organization.
- To familiarize the students with the conceptual foundation of accounting.
- To acquaint the students with basics of corporate reporting.
- To enable students to process and record business transactions.
- To make students understand the use of computer in the field of financial accounting.

3. Methods of Instructions:

3.1. General Instructions:

- Lecture
- Group discussion
- Question-answers
- Home Assignment and Self Study
- Classroom Presentation
- Problem Solving

3.2. Specific Instruction:

After completing each unit, students must present their views and opinions related to the taught subject matters**.**

4. Course Contents in Details	
Specific Objectives	**Contents**
Explain the ccounting principles and theories.	**Unit I: The Conceptual Foundation of Accounting (7 Hours)**
	1.1 Accounting as a Language of Business 1.2 Forms of Business Organizations 1.3 Types of Activities Performed by Business Organizations 1.4 Users of Accounting Information: Internal and External 1.5 Qualitative Characteristics of Accounting Information 1.6 The Accounting Profession – Role and Activities of an Accountant 1.7 The Accounting Framework – GAAP 1.8 Accounting Terminology 1.9 Accounting Information System in Modern Business Organizations 1.10 Use of Computers in Accounting Process
Describe the financial information clearly to stakeholders and ensure adherence to accounting standards and regulations.	**Unit II: Basics of Corporate Reporting (5 Hours)**
	2.1 Legal Requirement of Accounting: Provisions of Company Act Relating to Accounting 2.2 Introduction to Accounting Standards: IFRS and NAS 2.3 Annual Report 2.3.1 Definition 2.3.2 Major Components/Elements of Annual Report 2.4 Financial Statements 2.4.1 Definition 2.4.2 Objectives of Financial Statement (Primary & Secondary) 2.4.3 Components of Financial Statement 2.4.3.1 Income Statement 2.4,3.2 Statement of Retained Earnings 2.4.3.3 Balance Sheet 2.4.3.4 Statement of Cash Flow 2.4.3.5 Statement of Changes in Stockholders' Equity 2.5 Accounting Policies and Notes 2.6 Introduction to Audit 2.7 Legal Provisions Regarding Audit of Accounts in Nepal
Explain in precise systematic documentation of all financial activities within an organization.	**Unit III: Processing and Recording Business Transactions (6 Hours)**
	3.1 The Basic for Recording Transactions 3.1.1 Sources of Accounting Information 3.1.1.1 Types of Source Document 3.1.1.2 Role/ Importance of Source Documents 3.1.2 Recording of Transaction and Evens 3.1.2.1. External Transactions (Events) 3.1.2.2 Internal Transactions (Events) 3.1.3 The Accounting Equation and Analysis of Transactions 3.1.3.1 Steps Involved in Developing Accounting Equation 3.2 The Double Entry System 3.2.1 The Account and its Analysis

	3.2.2 Debits and Credits and its Rules 3.2.3 The Journals (General Journals and Compound Journals0 3.2.4 General Ledger (Standard Format and T-Account) 3.2.5 Normal Account Balances 3.2.6 Trial Balance (Meaning/ Objectives/ and Preparation) 3.2.7 Use of Excel in Processing Business Transactions
Describe an organization's profitability, health, financial performance, sources, and utilization of fund over a specific period through the income statement, balance sheet, and cash flow statement.	**Unit IV: Preparation of Financial Statement (15 Hours)**
	4.1 Income Statement 4.1.1 Concept of Income Statement 4.1.2 Basic Concepts, Principles, and Conventions used in Income Statement 4.1.3 Major Components of Income Statement 4.1.4 Preparation of Income Statement (vertical multi-step format) 4.1.5 Statement of Retained Earnings 4.2 Balance Sheet 4.2.1 Concept of Balance Sheet 4.2.2 Basic Concept, Principles, and Conventions used in Balance Sheet 4.2.3 Major Components of Balance Sheet 4.2.4 Preparation of Balance Sheet 4.2.5 Use of Computers in Preparation of Income Statement and Balance Sheet 4.3 Cash Flow Statement 4.3.1 Meaning of Cash Flow Statement 4.3.2 Purpose of the Cash Flow Statement 4.3.3 Importance of Cash Flow Statement 4.3.4 Classification of Cash Flow Activities 4.3.5 Preparation of Cash Flow Statement 4.3.6 Two Methods of Reporting Cash Flow from Operating Activities 4.3.7 Accruals and Cash Basis Accounting 4.3.8 Reconciling Cash Flow under Operating Activity using Indirect Method 4.3.9 Use of Computers in Preparation of Cash Flow Statement 4.4. Financial Ratios based on Financial Statements 4.4.1 Liquidity Ratios 4.4.2 Leverage Financial Ratios 4.4.3 Efficiency Ratios 4.4.4 Profitability Ratios
Explain "cash and cash equivalents" in financial management, enabling effective liquidity management strategies and minimizing financial risk for the organization.	**Unit V: Accounting for Cash and Cash Equivalent (6 Hours)**
	5.1. Components of Cash and Cash Equivalent 5.2 Need for Adjustment to Accounting Records 5.3 Preparation of the Bank Reconciliation Statement 5.4 Petty Cash 5.5 Balance Sheet Presentation of Cash and Cash Equivalent 5.6 Internal Control System 5.7 Cash Control (Receipt and Disbursement)

	Unit VI: Inventories and Cost of Goods Sold (9 Hours)
Compare the inventory management techniques and its impacts on profitability and operational efficiency of the organization.	6.1 Introduction to Inventory 6.2 Meaning and Nature of Inventory 6.3 Determining Inventory Quantities 6.4 Cost Included in Inventory 6.5 Cost of Goods Sold Models 6.6 Inventory System: Periodic and Perpetual 6.7 Inventory Costing Methods with Periodic System 6.8 Inventory Costing Methods with Perpetual System 6.9 The Choice of Inventory Method 6.10 Inventory Valuation and Income Measurement 6.11 Disclosure in the Financial Statement 6.12 Accounting Principles Related to Inventory 6.13 Methods of Inventory Estimation 6.14 Analysis of Inventory

5. List of Tutorials	
SN	**Tutorials**
1.	Identifying financial issues, which are against GAAP, IFRS, and NAS.
2.	Searching annual report of any organizations and analyzing overall contents of the report.
3.	Collecting source documents and analyzing how an organization records transaction based on source documents.
4.	Preparing journals, ledgers, and trial balance from different types of economic transactions.
5.	Preparing income statement, statement of retained earnings, and balance sheet.
6.	Preparing cash flow statement.
7.	Solving different types of problems related to financial ratios and interpreting them.
8.	Evaluating inventory using different methods under periodic and perpetual system.
9.	Solving different issues related to bank reconciliation statement.

6. List of Practical	
SN	**Practical**
1.	Preparing income statement, balance sheet, and cash flow statement using accounting software.
2.	Prepare and present a case study on accounting software used by any public or private organization.

7. Evaluation System and Students' Responsibilities

Evaluation System

The internal evaluation of a student may consist of assignments, attendance, term-exams, lab reports and projects etc. The tabular presentation of the internal evaluation is as follows:

Internal Evaluation	**Weight**	**Marks**	**External Evaluation**	**Marks**
Theory		**30**	**Semester End**	**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				

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.

8. Prescribed Books and References

Text Books:

1. Porter, G.A., & Norton, C.L. *Financial Accounting*: *The Impact on Decision Makers.* USA: The Dryden Press

References:

1. Hermanson, H.R. and Edwards, D.J. Financial Accounting: *A Business Perspective.* USA: Von Holffmann Press.
2. Kimmel, P.D., Weygandt, J.J., & Kieso, D.E. *Financial Accounting*. New Delhi: Wiley India Pvt. Ltd.
3. Narayanswamy, R. *Financial Accounting: A Managerial Perspective.* New Delhi: Prentice Hall of India.
4. Koirala, M.P., Acharya, C., Sharma, L.P.B., Sharma, N., & Gautam, C.M. *Financial Accounting*. Kathmandu: Buddha Academic Enterprises.
5. Nepal Accounting Standard (NASs).
6. International Accounting Standards (IASs) / International Financial Reporting Standards (IFRSs).

Pokhara University Faculty of Science and Technology		
Course Code: CMP 118 (3 Credits)		Full Marks: 100
Course Title: **Programming in C (3-3-3)**		Pass Mark: 45
Nature of the Course: Theory + Practical		Total Lectures: 48 hours
Level: Bachelor	Year: I / Semester: II	Program: Bachelor of Computer Application

1. Course Description

This course is designed to encompass the concepts of the C programming through a combination of theory and practical components. This course covers syntax, data types, control structures, functions, arrays, pointers, file handling, and dynamic memory allocation. The course aims to equip students with the skills necessary to write efficient and error-free C programs and apply their knowledge to real-world programming challenges.

By the end of this course, students will be able to write efficient and error-free program codes using C programming language, understand the underlying principles of the language, and apply knowledge to solve real-world programming challenges. This course will lay a solid foundation to understand programming structure and coding techniques and further support pursuing a career in software development, systems programming, or any other field that requires C programming skills.

2. General Objectives

- To provide the comprehensive understanding of the programming language.
- To introduce the develop proficiency in writing structured and error-free programs.
- To utilize different data types and control structures effectively in programming.
- To gain hands-on experience with arrays, pointers, and file handling in C.
- To enhance programming skills through a series of practical exercises and assignments, encouraging the application of learned concepts in real-world scenarios.
- To develop the ability to critically analyze and solve practical programming problems using the programming language, fostering a strong foundation for future programming endeavors.

3. Methods of Instruction

These are sample Strategies,which teachers can use to accelerate the attainment of the various course outcomes.

- Use of Video/Slides/Animations to explain functioning of various concepts.
- Incorporate collaborative learning in the class.
- Use interactive methods through demonstration and group discussion
- Adopt Problem Based Learning to fosters student's Analytical skills, develop design thinking skills such as the ability to design, evaluate, generalize, and analyze information
- Use visualization tool (for Eg. https://pythontutor.com/visualize.htm) in order to visualize the operations of C Programs

4. Course Contents in Details

Specific Unit wise Objective	**Course Contents**
Unit-1 Introduction to C Programming	**3 hours**
▪ Review about algorithm and flowchart. ▪ Describe history and features of C ▪ Explain the basic construct of structured programming	1.1 Review of Flowchart and Algorithms 1.2 History of C programming language 1.3 Features of C programming language 1.4 Structure of a C program 1.5 Preprocessor directives
Unit-2 Variables, Operators, and Datatypes	**7 hours**
▪ Explain the basic constructs of C language	2.1 C character set, Tokens 2.2 Keywords and reserved words 2.3 Identifiers with naming convention.

	2.4 Constants and Variables 2.4.1 Introduction to Constants and Variables 2.4.2 Variable Declaration 2.4.3 Variable types 2.5 Datatypes 2.6 Operator 2.6.1 Types 2.6.2 Precedence and associativity of operators 2.7 Formatted and Unformatted I/O functions 2.8 Comments- Single line and Block
Unit-3 Control Statements	**8 hours**
▪ Describing the role of control statement with its types	3.1 Clean code practices 3.2 Sequential control Statement 3.3 Selection Control Statement 3.3.1 Decision-making with if-else statements (if, if-else, else if) 3.3.2 Nested if 3.3.3 Conditional Operator 3.3.4 Switch statement for multi-choice decisions 3.4 Iteration control Statement 3.4.1 Looping with for, while and do-while loops 3.4.2 Nested Loop 3.5 Jump Statements 3.6 Best practices for writing structured code.
Unit-4 Arrays and Strings	**10 hours**
▪ Describe the arrays and their declaration, accessing and manipulation in C ▪ Introduce the strings and string manipulation functions ▪ Explain the role of multi-dimensional arrays and its implementation	4.1 Introduction to arrays and their declaration 4.2 Accessing array elements and array indexing 4.3 Manipulating array elements 4.4 Multi-dimensional arrays 4.5 Working with strings and string manipulation functions
Unit-5 Functions	**7 hours**
▪ Describing the basics of functions declaration, definition, calling and returning ▪ Compare the pass by value and pass by address ▪ Explain the role of recursion and pass arrays/strings to functions	5.1 Introduction to function 5.1.1 Types of function (Library and User defined) 5.2 Declaration, definition, and calling 5.3 Function arguments and return values 5.4 Function prototypes: 5.4.1 No Arguments and No return values 5.4.2 Arguments but No return Values 5.4.3 Arguments with return values 5.4.4 No arguments but return a value 5.5 Recursive function. 5.6 Iteration versus recursion 5.7 Passing arrays and strings to functions
Unit-6 Structure and Union	**4 hours**
• Describe the user defined data types • Compare features and function of Structure and Unions • Introduce code modularity and maintainability, reusability and reduce	6.1 Introduction to Structure and Union 6.2 Declaration, accessing members, initialization, size. 6.3 Nested structures 6.4 Arrays of structures

complexity.	6.5 Passing structures to functions, returning structures from functions. 6.6 Difference between structure and union
Unit-7 Pointers	**3 hours**
▪ Describe the role of pointers with its operator. ▪ Introduce dereferencing and access memory location ▪ Explain the role of pointer arithmetic ▪ Compare the features and functions for dynamic memory allocation	7.1 Introduction to pointers and pointer declaration 7.2 Dereferencing pointers and accessing memory locations. 7.3 Returning multiple values using pointers 7.4 Pointer arithmetic's 7.5 Pointers and arrays. 7.6 Pointers to structures 7.7 Pointer as a function argument 7.8 Dynamic Memory Allocation
Unit-8 File Handling	**6 hours**
▪ Discuss the role of file and file handling in C with its modes and functions	8.1 Opening, reading, writing, and closing files using different techniques 8.2 Sequential and random access to files 8.3 Error handling and file input/output operations 8.4 File handling best practices

5. Laboratory and Assignments

The instructor should design laboratory and assignments by taking into consideration the necessary knowledge, technological aspects, practicality, and current trends related to the topic. The assignment guidelines should cater to the average students while also providing the opportunity to engage and challenge the more advanced ones. It is advisable for the instructor to create multiple sets of assignments and distribute them among different groups of students. It is also preferable that the assignments are based on real-world problems or applications. Students should be encouraged to utilize Hungarian notation, practice proper indentation, and include comments in their code.

Furthermore, the instructor can assign a mini-project that showcases the application of the learned concepts in a real-life scenario. Additionally, they may assign an assignment or mini-project that aligns with the respective branch of study, going beyond the syllabus.

6. Evaluation System and Students' Responsibilities

Evaluation System

The internal evaluation of a student may consist of assignments, attendance, term-exams, lab reports and projects etc. The tabular presentation of the internal evaluation is as follows:

Internal Evaluation	**Weight**	**Marks**	**External Evaluation**	**Marks**
Theory		**30**	**Semester End**	**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				

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.

7. Reference Books:

1. S.K. Srivastava and Deepali Srivastava: C in Depth
2. Kely and Pohl: A book on C
3. Wait, Mitchell, Steven Prata and Donald Martin: C primer Plus
4. Yeswant Kanetkar: Let us C

Pokhara University Faculty of Science and Technology		
Course Code: ELX 112 (3 Credits)		Full Marks: 100
Course Title: : Microprocessor and Computer Architecture (3-3-1)		Pass Mark: 45
Nature of the Course: Theory + Practical		Total Lectures: 48 hours
Level: Bachelor	Year: I / Semester: II	Program: Bachelor of Computer Application

1. Course Description

This course is designed to encompass the fundamental concepts of 8085 and 8086 with assembly level programming. It also provides major notions for the design of processing unit, control unit architectures. It presents the various computer arithmetic algorithms to solve the computer arithmetic problems. This course also introduces memory and input/output organization and the advance computer architecture. After completion of this course, students can design simple CPU and Control Unit.

2. General Objectives

The major objectives of this course are

- To acquaint the students with basic concepts of basic architecture details of microprocessor (8085/8086)
- To acquaint the students with concepts of assembly level programming using 8-bit microprocessor
- To acquaint the students with the fundamentals of computer systems.
- To acquaint the students with the knowledge of computer architecture and associated processing, control unit and ALU unit of very simple central processing unit.
- To apprise the students with the architectural and associated components of computer systems.
- To aware the students about the architecture of the computer systems available in the market.

3. Methods of instructions

Lecture, Project work and Practical

4. Content in details

Specific objectives	**Contents**
• Familiarize with fundamentals of microprocessor-based system along with their real-world applications.	**Unit 1 An overview of computer and Microprocessor[3 Hrs]** 1.1. Brief overview of microprocessor and microcontroller 1.2. Organization of computer systems 1.3. Architecture of computer 1.4. Applications of microprocessor and microcontroller
• Describe the architecture of 8085 microprocessor and basic assembly level programming. • Explain the machine cycles involved during the execution of instructions.	**UNIT 2 Architectural and assembly language programming of 8085 [12 Hrs.]** 2.1 Pin Configuration, 2.2 Functional Block Diagram 2.3.1 Timing and Control Unit, 2.3.2 Registers, 2.3.3 Data and Address Bus, 2.3.4 ALU 2.3.5 I/O Interface 2.3 Intel 8085 instructions types 2.3.1 Data transfer instructions 2.3.2 Arithmetic instructions 2.3.3 Logic instructions 2.3.4 Decision and Branching instructions

	2.3.5 Machine control instruction 2.4 Operation Code and Operands, 2.5 Addressing Modes, 2.6 Interrupts and Flags 2.7 Instructions Types and Data Flow inside 8085, 2.8 Timing Diagram (two examples of 4, 7, 10 and 13 T-states) 2.8 Basic Assembly Language Programming Using 8085 Instruction Sets
• Familiarize with the architecture of 16 bit microprocessor 8086. • concept of memory segmentation and pipelining in modern processor.	**UNIT 3 Overview of 8086 microprocessor [5 Hrs]** 3.1 Features of 8086 microprocessor 3.2 Functional diagram of 8086 microprocessor 3.3 Registers and Flags 3.4 ALP Development Tools: Editor, Assembler and linker
• Familiarize with CPU Fundamental	**UNIT 4 CPU Fundamental [4 Hrs]** 4.1 CPU organization/Structure 4.2 Register organization and data path 4.3 Arithmetic and Logic units 4.4 Design principal for modern system
• Examine the basic structure of a micro-sequencer and Hardwired Control Unit	**UNIT 5 Control Unit Design [6 Hrs]** 5.1 Control of the processor 5.2 Hardwired Control Unit(Control unit inputs and logic) 5.3 Microprogramed control units(Micro instruction and its types) 5.4 Architecture of micro programmed control unit 5.5 Microinstruction sequencing and execution 5.6 Application of hardwired and micro programmed control units
• Understand the representation of binary numbers in signed and unsigned notation along with the algorithms used for the basic arithmetic operations.	**UNIT 6 Computer Arithmetic [6 Hrs]** 6.1 Numeric format and representation of binary number in signed and unsigned notation 6.2 Addition and subtraction in signed and unsigned notation. 6.3 Shift and add multiplication algorithm, Booth's algorithm. (signed and unsigned)
• Review memory Hierarchy of computer system and study the concept of associative and cache memory in real world scenario.	**UNIT 7 Memory Organization [4 Hrs]** 7.1 Memory hierarchy 7.2 Memory interfacing diagram (RAM and ROM with 8085) 7.3 Associative memory 7.4 Cache Memory and mapping techniques
• Familiarize with serial and parallel communication interfaces and introduce various methods for improving I/O performances.	**UNIT 8 Input/Output Organization [4 Hrs]** 8.1 Serial and parallel communication interfaces 8.2 Programmed I/O 8.3 Interrupts, types of interrupts, Interrupt processing, Interrupt Hardware and priority 8.4 Direct Memory Access, I/O Processors
• Understand with the concept of instruction pipelining and multicore architecture in modern processor.	**UNIT 9 Advance Architectures [4 Hrs]** 9.1 RISC and CISC Fundamentals 9.2 Instruction Pipeline, Register window 9.3 Flynn's Taxonomy, MIMD system topologies and architectures 9.4 Introduction to multicore architecture

5. Laboratory Works

The laboratory works should include following concepts:

1. Assembly language program using 8085 microprocessor trainer kit or any software simulator.
   1.1 Use of all types of instructions and addressing modes. Programs including basic arithmetical, logical, looping, bitwise and branching instructions.
2. Computer Arithmetic can be implemented with C/C++.

6. List of Tutorials:

The various tutorial activities that outfits this course should cover all the content of this 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 hrs should be conducted to cover all the content of this course:

A. Discussion based Tutorials [2 hrs]

1. Comparative analysis of Von-Neumann, Harvard and modified Harvard Architecture.
2. Applications of microprocessor and microcontroller based system in real world.
3. Hardware and programmable module of 8 bit microprocessor.
4. Interpret the concept of memory segmentation and pipelining in 16 bit microprocessor.
5. Review each entity memory hierarchy for modern processors.
6. Interfacing of DMA, interrupt controller and UART with 8085 microprocessor.
7. Comparative analysis of different aspects of computing system as defined in Flynn's Classification.

B. Assembly level programming tutorials [3 hrs]

1. Assembly level programming illustrating data transfer instructions, arithmetic instructions, logical instructions and branching instructions in 8085 microprocessor.
2. Enlighten PUSH and POP instructions using appropriate assembly level programs.
3. Assembly level programming for simple sequencing program, multiplication, division, table processing and sorting.
4. Timing diagram for different types of instructions and RTL of each machine cycle involved during the execution of instructions.

C. Design tutorials[3 hrs]

1. Design a CPU for any given registers set, instruction set and state diagram. Show the RTL code for each execution cycle.
2. Develop a control unit for any given state diagram.
3. Design a micro-sequencer control unit for any given specifications following design procedure.

D. Computer Arithmetic tutorials [3 hrs]

1. Perform arithmetic addition and subtraction in signed and unsigned notation for any given numbers.
2. Perform Multiplication operation for any given numbers using shift-add multiplication algorithm and Booth's algorithm.
3. Perform Division operation for any given numbers using restoring and Non restoring Division algorithm.

E. Project work [4 hrs]

Develop a case study report about any of the modern Advance superscalar processors. It should include the architecture of processor, control unit, memory as well as input output organization in detail. An oral presentation with the submission of report should be a part of work and must be included as a component for evaluation.

7. Evaluation system and Students' Responsibilities

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 Marks		**50**		
Full marks=50+50				

Students 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 Examination. Students are advised to attend all the classes, formal exam, and test 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 Text Books and references

Text Books:

1. Gaonkar, Ramesh S., Microprocessor Architecture, Programming, and Applications with 8085, Prentice Hall. New Delhi
2. Stallings, W., "*Computer Organization and Architecture*", Eighth Edition, 2011, Pearson.

References:

1. Hall, Douglas V. Microprocessor and Interfacing programming and Hardware, McGraw Hill, New Delhi
2. Carpineili, John D., Computer system Organization and Architecture, Addison Wesley. Pearson Education Asia (LPE),2001
3. Malvino: Digital Computer Electronics and Introduction to Microcomputers
4. Dougals V. Hall: Microprocessor and Interfacing programming and Hardware, McGraw Hill.
5. Mano, M.M., "*Computer Systems Architecture*", Third Edition, 2011, Pearson.
6. Tanenbaum, A.S., "*Structured Computer Organization*", Fourth Edition, 2003, Pearson Education.
7. Rajaraman, V. et all, "*Computer Organization and Architecture*", 2011, PHI.
8. Sima, D. et all, "*Advanced Computer Architecture*", 2000, Addison Wesley.

Pokhara University **Faculty of Science and Technology**		
Course Code: PRJ 151		Full Marks: 100
Course Title: Project I (1-1-2)		Pass Mark: 45
Nature of the Course: Theory and Practical		Total Lectures: 48 hours
Level: Bachelor	Year : I / Semester : II	Program: Bachelor of Computer Application

1. Course Description:

This course is designed to provide students with hands-on practices in C programming through the implementation of programming logic and techniques, problem-solving, and efficient coding practices. Students will build upon their foundational knowledge of C programming and expand their abilities by applying concepts to real-world scenarios. Through the project work, students will be able to transform their theoretical knowledge into practical implementation.

2. General Objectives:

1. To develop practical skills for implementation of the C programming constructs.
2. To use and implement C language to develop real world applications.
3. To provide practical knowledge for planning and formulating project and its design, development, documentation and presentation.

3. Procedure:

The Project I is focused to implement the programming knowledge and skill that students acquired from the C programing language. A group of maximum four students can be formed to work in their project I. Students are themselves responsible to form the group and select project on any domain of common interest. Project supervisor(s) shall be assigned for guidance, tutoring and mentoring of the project activities. The project shall be evaluated by evaluation panel that includes internal evaluator(s) and/or supervisor and external examiner(s) appointed by the respective department. Appointment of External Examiner is mandatory for final defense of the project. Each group is required to complete all project work phases and develop software product that is SMART (Specific, Measurable, Achievable, Relevant, and Time-bound.

4. Project Phases:

The entire project work shall be divided into three phases and evaluation shall be done accordingly:

Phase I – Proposal Submission & Defense

In Phase I, students are required to form a team comprising of maximum 4 team members and draft a conceptual framework for their project work in the form of a Proposal. The project team must submit and present the proposal in presence of Examiner/Supervisor in a formal presentation and must be approved.

Phase II – Mid Term Defense

The project team needs to present the work progress to Examiner(s)/Supervisor and demonstrate as required. Till the mid-term defense, students must have finished the design phase including the overall system/architectural design and validation scheme. The Mid-Term defense can be scheduled after 4 weeks of project proposal defense.

Phase III – Final Submission and Defense

Before the semester ends, students must submit and defend the project final work (oral presentation and demonstration) in presence of the evaluation panel inclusive of external examiner. The project work should be presented in complete format that include project requirement analysis, design, coding and testing. Students are required to submit the final report 1 week prior to the final defense. Students must submit the hardcopy/softcopy of the project in the standard format prescribed by respective department of the college.

5. Evaluation Criteria:

The overall project evaluation shall be cumulative of Phase I, Phase II and Phase III evaluations. Phase I and II shall be evaluated by internal examiner and Phase III evaluation shall be done by evaluation panel inclusive of external examiner. The evaluation shall be done based on following criteria.

Phase I (20%)

Tasks required to be accomplished in Phase I.

- Feasibility Study
- Requirements Analysis and Specification
- Project plan
- Creativity, Innovativeness and Usefulness of the Idea

Phase II (30%)

Tasks required to be accomplished in Phase II.

- System/Architectural Design
- Progress
- Level of achievement
- Group/Team Effort
- Ability to propose solutions

Phase III (50%)

- Presentation Skills (5 Marks)
- Project Completeness and Demonstration (10 Marks)
- Final Output (10 Marks)
- Viva-Voce (10 Marks)
- Documentation (15 Marks)

6. Documentation Structure

The documentation structure defines the components that must be included in the final report after the completion of the Project I. It may include the following components:

Preliminaries

- Cover Page
- Acknowledgment
- Letter of Approval
- Abstract
- Table of Content
- List of Figures
- List of Tables
- List of Abbreviation

Chapter 1 : Introduction

1. Background
2. Problem Statement
3. Objective
4. Project Schedule

Chapter 2 : System Analysis & Design

1. Feasibility study
2. Algorithm
3. Flowchart
4. Context Diagram
5. Data Flow Diagram
6. User interface

Chapter 3 : Development and Testing

1. Implementation (Development Environment & Operation Procedure)
2. Verification and Validation

Chapter 4 : Results & Conclusion

References (Respective school/department may decide which citation and reference notation to choose in the report.)

Annex (It may include implementation of main algorithm(s) of project, screenshots of user-interface and/or any other detail information that the project report requires to present.)

The documentation structure may slightly differ based on the nature of the project that the students have chosen. If so, students should take the approval from the respective department and/or their supervisor.

Pokhara University
Faculty of Science and Technology

Title of the Course: **Object Oriented Programming using Java**

Course code: CMP215	Full marks: 100
Nature of Course: Theory and Practical	pass marks: 45
Semester: III	Time per period: 1 hour
Level: Bachelor	Total Lectures: 48 Hrs

Program: BCA

1. COURSE DESCRIPTION

This course equips students with comprehensive knowledge of various Object-Oriented aspects of Java, guiding them through fundamental OOP concepts such as Classes, Objects, Inheritance, Interfaces, Polymorphism, and Packages. It also covers the basics of graphical user interfaces. The skills acquired in this course will enable students to develop standalone applications for real-world use.

By the conclusion of this course, students will have the ability to write efficient, error-free Java programs, establishing a strong foundation for future endeavours.

2. GENERAL OBJECTIVE

- To acquaint the students with basic knowledge of java environment.
- To familiarize the Students with Object Oriented Programming Concepts and Techniques.
- To apprise the basic JAVA concepts
- To make the students knowledgeable about classes, objects and class libraries
- To acquaint the students to develop simple applications with OOPS concepts
- To illustrate the concepts of Inheritance and polymorphism.
- To introduce interfaces, packages, JavaFX and Socker programming basic concepts.
- To enhance programming skills through a series of practical exercises and assignments, encouraging the application of learned concepts in real-world scenarios.

3. CONTENTS IN DETAIL

Title of the Course	**OBJECT ORIENTED PROGRAMMING USING JAVA**	
CREDIT:3 Practical:3	SEMESTER: III	CCODE:CMP 215
Specific Objective	**CONTENTS**	**LECTURE HOURS**
• To introduce the basic concepts of Object-Oriented Programming • To make user friendly in JAVA environment • To demonstrate basic structure of the program	**UNIT- 1: Introduction to Java Programming** 1.1 History and Features of Java 1.2 C++ vs Java 1.3 Introduction 1.3.1 JDK 1.3.2 JVM 1.3.3 Bytecode 1.3.4 Java Environment. 1.4 Procedure Oriented Vs Object Oriented 1.5 Basic Concepts of Object-Oriented Programming 1.6 Structure of Java Programming, Naming conversions 1.7 Compiling and executing simple program 1.8 Scanner class	**4 hrs**
• To introduce the Fundamental of JAVA language • To Describe the concept of Arrays and Strings	**UNIT-2: Java Fundamental Concepts** 2.1 Java Tokens, Keywords, Statements 2.2 Constants and Variable 2.3 Data Types 2.4 Operators 2.5 Control Statements 2.7.1 Branching 2.7.2 Looping Statements 2.7.3 Jump Statements 2.6 Arrays in Java 2.8.1 One dimensional 2.8.2 Two dimensional 2.7 Java Strings 2.9.1 String Length 2.9.2 Concatenation 2.9.3 String Comparison 2.9.4 String Buffer 2.8 Methods in Java 2.10.1 Predefined methods 2.10.2 User defined methods 2.10.3 Static method	**7 hrs**
• To Describe and familiarize	**UNIT-3: Java Object Oriented Programming** 3.1 Fundamental of Class 3.2 Object fundamental 3.3 new Keyword	**8 hrs**

the concept of Class and Object, methods in real life Application • To demonstrate java. Lang package	3.4 Member methods of a class 3.5 Instance method 3.6 Constructor and its types 3.7 Finalize method 3.8 Abstraction 3.9 Encapsulation 3.10 Using this keyword 3.11 Method overloading 3.12 Wrapper classes 3.13 Autoboxing and unboxing 3.14 Visibility Modes	
• To demonstrate and apply the Inheritance and interface concept in application	**Unit-4: Inheritance, Polymorphism, Interface and Packages** 4.1 Inheritance 4.1.1 Using Extends 4.1.2 Member access 4.1.3 Super classes and sub classes 4.1.4 Single Inheritance 4.1.5 Multi-level Inheritance 4.1.6 Hierarchy Inheritance 4.2 Constructor and inheritance 4.3 Polymorphism-Method overriding 4.4 Abstract class 4.5 Using final keyword 4.6 Interface 4.6.1 Creating and implementing an interface 4.6.2 Extending interface 4.7 Packages 4.7.1 Creating Packages 4.7.2 Accessing a Package	**8 hrs**
• To provide the knowledge how threads work in the real time applications • To demonstrate the java. Utility package	**UNIT-5: Multithreading, Java Collections** 5.1 Introduction-Multithreading 5.1.1 Thread class and Runnable Interface 5.1.2 Creating Threads 5.1.3 Stopping and Blocking a Thread 5.1.4 Life Cycle of a Thread 5.1.5 Using Thread Methods 5.1.6 Thread Priority 5.2 Collection Framework in Java 5.2.1 Overview of java collection framework 5.2.2 Commonly used collection classes- ArrayList, Vector	**6 hrs**
• To familiarize the students with the exception handling	**Unit-6-Exception Handling, Java IO and Streams** 6.1 Managing Errors and Exceptions 6.2 Syntax of Exception Handling Code 6.3 Types of Exceptions 6.4 Using try and catch keywords	**7 hrs**

process in real life applications • To discuss the role of Stream and file handling	6.5 Multiple catches 6.6 Using Finally Statement 6.7 Throwing Exceptions 6.8 Streams: Byte Streams and Character Streams 6.9 Reading / Writing Console Input / Output 6.10 Reading and Writing files,	
• To Describe and gain knowledge of database connectivity in java	**Unit-7-Java JDBC** 7.1 JDBC introduction 7.2 JDBC drivers 7.3 Connecting to Database 7.4 Querying a database and processing the results	**4 hrs**
• To introduce GUI based programming.	**Unit-8- JavaFX a** 8.1 Introduction to JavaFX 8.2 JavaFX Architecture 8.3 JavaFX Application Basics-Line, rectangle and text	**4 hrs**

4. Methods of Instruction

The general teaching pedagogy that can be followed by teaching faculties for this course includes class lectures, laboratory activity, group discussions, case studies, guest lectures, research work, project work, assignments.

- Lecturer method need not to be only traditional lecture method, but alternative effective teaching methods could be adopted to attain the outcomes.
- Use of Video/Slides/Animations to explain functioning of various concepts.
- Discuss how every OOPS concept can be applied to the real world; it helps to improve the students' understanding.
- Group discussion is must for the each and every topic.
- Guest lectures will help the students to gain more knowledge.
- Project work convert the student ideas into practical
- Assignments helps the students to recall every topic.

5. List of Practical

1. Write a java Program for (Fibonacci Series, Factorial number, Palindrome)
2. Write a java Program to Swap Two Numbers
3. Write a Java program to add two binary numbers.
4. Find the smallest and largest element from the array
5. Write a Java program to reverse a string.
6. Write a java program to count number of words in a given text
7. Designed a class that demonstrates the use of constructor.
8. Write a java program to implement single level inheritance and interface.
9. Write a java program to implement method overriding
10. Create a package, add the necessary classes and import the package in java class.

11. Write a java program to implement thread life cycle.
12. Write a java program to implement exception handling.
13. Write a java program to implement Array list and vector.
14. Write a Java program that demonstrates JavaFX with Line, Rectangle and Text.

By incorporating all the conceptual details of OOP in Java develop a mini project so that students practical marks is evaluated through project.

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	5
		Discipline and Class performance	5
		Assignments, Presentation and Practical	15
		Internal Term Exam	25
Total External	50	Total Internal	50
Full Marks 50+50 = 100			

Students' Responsibilities:

To be eligible for the Semester End Examinations, 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.

7. Prescribed Books and References

Recommended Text book

1. Herbert Schildt, Java the Complete Reference, 11th Edition, Tata McGraw-Hill Publishing Co. Ltd.

2. E. Balagurusamy, *"Programming with Java: A Primer"*, Edition, Tata McGraw Hill Publication.

Reference Books

1. Y. Daniel Liang, Introduction to Java Programming Brief Version, 8th Edition Kindle Edition.

2. Cay S. Horstmann, Core Java Volume I–Fundamentals, 11th Edition, Pearson 2019.

Pokhara University
Faculty of Science and Technology

Course Code.: CMP 227	Full marks: 100
Course title: **Data Structure and Algorithm**	Pass marks: 45
Nature of the course: Theory and Practical	Time per period: 1 hour
Year, Semester: 3rd	Total periods: 48
Level: Bachelor	Program: BCA

1. Course Description

This course introduces students to fundamental data structures and algorithms. It covers core concepts, analysis techniques, and practical implementation. Students will learn to design efficient algorithms, analyze their performance, and apply them to problem-solving. The course emphasizes hands-on practice through coding assignments and projects.

2. General Objectives

- To familiarize students with fundamental data structures (stacks, queues, linked lists, trees, graphs) and algorithms, including sorting and searching techniques.
- To equip students to analyze algorithm efficiency using time complexity metrics (Best, Worst, and Average case).
- To enhance problem-solving and critical thinking skills through practical assignments, projects, and the comparison of recursive and iterative methods.
- To prepare students for industry challenges by integrating modern tools, techniques, and the application of fundamental algorithms to modern technological fields.

2. Contents in Detail

This section contains the details to be taught under the course. Normally, the contents can be divided into 5 to 8 units, and each unit can be assigned 5 – 8 teaching hours.

Specific objectives	Contents
• Understand how data structures and algorithms (DSA) contribute to effective problem-solving. • Describe and categorize abstract data types (ADTs), structures, and basic data types.	**Unit 1: Introduction to Data Structures and Algorithms (7 hrs)** 1.1. Introduction to Data Structures 1.1.1 Data Types and Data Structures 1.1.2. Classification of Data Structures 1.1.3. ADTs (Abstract Data Types) 1.2 Introduction to Algorithms 1.2.1 Definition and Characteristic 1.2.2. Need/Importance of DSA

• Learn the basics of algorithms, such as their meaning and significance. • Use time complexity and asymptotic notations (Big O, Omega, Theta) to analyze algorithms. • Learn about the impact of dynamic memory allocation on performance.	1.3. **Algorithm design** Divided and Conquer, Greedy Algorithm, Backtracking, Dynamic Programming 1.4 Algorithm Analysis 1.3.1. Time Complexity 1.3.2. Best, Worst and Average Case and Rate of Growth 1.3.3. Asymptotic Notation (Big O, Omega and Theta)
• Understand the concept of stacks as an ADT. • Master stack operations and array-based implementation. • Apply stacks for practical applications like expression evaluation. • Grasp the concept of queues as an ADT. • Implement various queue types (linear, circular, priority). • Understand the concept of recursion and its role in problem-solving. • Differentiate between recursive and iterative approaches. • Apply recursion to solve problems like factorial, Fibonacci, GCD, and Tower of Hanoi. • Analyze the relationship between recursion and stack operations.	**Unit-II: Stack, Queues and Recursion (8 hrs)** 2.1. Concept of stacks and Stack ADT 2.1.1. Stack operations 2.1.2. Stacks implementation using arrays 2.1.3. Applications of stacks: 2.1.3.1. Conversion from infix to postfix 2.1.3.2. Evaluation of postfix 2.1.4. Recursion 2.1.4.1. Concept of recursion 2.1.4.2. Recursive vs. iterative solutions 2.1.4.3. Recursive problems: Factorial, Fibonacci sequence, GCD, Tower of Hanoi 2.1.5. Recursion and stack 2.2. Concept of queues and Queue ADT 2.2.1 Queue operations 2.2.2. Queue Implementation using arrays 2.2.3. Types of Queues: 2.2.4. Linear and circular 2.2.5 Introduction to priority queues and double ended queue.
• Understand the concept of lists and Abstract Data Types	**Unit-III: Linked List (8 hrs)**

(ADTs), including array-based implementation. • Define linked lists and explore their basic operations. • Learn about different types of linked lists: Singly Linked List, Doubly Linked List, and Circular Linked List. • Perform basic operations in linked lists such as node creation, insertion, and deletion at various positions. • Implement stacks and queues using linked lists, with a focus on doubly and circular linked lists.	3.1 List –Definition and List ADT, array Implementation 3.2 Linked List – Definition and its operations 3.3 Types of Linked Lists: 3.3.1 Singly Linked List 3.3.2 Doubly Linked List 3.3.3 Circular Linked List 3.4 Implementation of Singly Linked Lists: 3.4.1 Node Creation 3.4.2 Node Insertion (Beginning, End, Specified Position) 3.4.3 Node Deletion (Beginning, End, Specified Position) 3.6 Linked List implementation of Stack and Queue
• Understand the fundamental concepts and terminology of tree data structures. • Implement and analyze various tree traversal algorithms for binary trees. • Design and manipulate binary search trees, including insertion, deletion, and searching operations. • Explain the need for balanced trees and understand the basic principles of AVL trees. • Introduce the concept of multi-way search trees, specifically B-trees, and their applications. • Apply tree structures to solve real-world problems, such as data compression using Huffman coding.	**Unit-IV Trees (8 hrs)** 4.1 Tree concepts and terminology 4.2 Binary trees: 4.2.1 Definition and types 4.2.2 Tree representations (array-based, linked-based) 4.2.3 Tree Traversal (PreOrder, PostOrder and InOrder) 4.2.4 Construction of binary tree from traversal 4.2.5 Application of Tree: Huffman Algorithm 4.3 Binary search trees: 4.3.1 Construction of BST 4.3.2 Basic Operation on BST (Insert, delete, search) Node 4.4 Introduction to balanced trees 4.4.1 Need for balanced trees 4.4.2 Definition of AVL (Adelson-Velskii and Landis) trees, insertion, deletion and Rotation operations 4.5 B-tree Definition and application

• Understand the fundamental concepts of sorting and its importance in computer science. • Analyze and implement various sorting algorithms, including simple and advanced techniques. • Compare the efficiency of different sorting algorithms using time complexity analysis. • Recognize the appropriate sorting algorithm for specific data sets and problem requirements. • Implement and understand the workings of distribution-based sorting algorithms. • Appreciate the connection between heap sort and priority queues (Binary heap max-min) in data structures.	**Unit-V: Sorting (6 hrs)** 5.1 Definition and Classification of Sorting algorithm (internal, external, stable, unstable) 5.2 Bubble Sort 5.3 Selection Sort 5.4 Insertion Sort 5.5 Merge Sort 5.6 Quick Sort 5.7 Radix Sort 5.8 Heap Sort as priority queues
• Understand the concept and applications of searching algorithms. • Implement and analyze Binary Search for efficient data retrieval. • Learn the fundamentals of Hashing, including Hash Functions and Hash Tables. • Explore Collision Resolution Techniques: Separate Chaining, Linear Probing,	**Unit-VI : Searching Algorithm and Hashing (4 hrs)** 6.1Concept of Searching algorithms 6.2 Linear and Binary Search 6.3 Hashing: 6.3.1Concept of Hashing, Hash functions, Hash Table 6.4 Collision resolution techniques: 6.4.1. Open Hashing: Separate Chaining 6.4.2. Closed addressing : Linear Probing , Quadratic Probing, Double Hashing 6.5. Load Factor and Rehashing

Quadratic Probing, and Double Hashing. • Understand Load Factor and Rehashing to optimize hash table performance.	
• Understand graph concepts, representations, and types. • Master graph traversal techniques (DFS, BFS, topological sort). • Apply graph algorithms for finding minimum spanning trees (Kruskal's, Prim's) and shortest paths (Dijkstra's). • Analyze graph problems using algorithm design techniques (divide and conquer, greedy, backtracking, dynamic programming). • Apply graph concepts to real-world problems and applications.	**Unit VII: Graphs (4 hrs)** 7.1. Graph concepts and representations (adjacency matrix, adjacency list) and types 7.2. Graph traversals: 7.2.1. Depth-first search (DFS) 7.2.2. Breadth-first search (BFS) 7.2.3. Topological Sort 7.3. Minimum Spanning Tree: Kruskal’s Algorithm and Prim’s Algorithm 7.4. Shortest path problems:- Dijkstra's algorithm
• Understanding the main types, including Divide and Conquer, Greedy Algorithms, and Backtracking. • Exploring data structures optimized for machine learning tasks. • Introduction to algorithms designed for quantum computing environments.	**Unit VIII:**

• Examining recent advancements and trends in data structure development.	

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

3. **Methods of Instruction**

   The Data Structures and Algorithms course will employ a balanced approach to instruction, combining theory with practical application. Interactive lectures will introduce core concepts, supplemented by in-class problem-solving and hands-on laboratory work. Algorithm visualization tools and real-world case studies will enhance understanding and demonstrate the relevance of DSA in modern technology. Collaborative group discussion will encourage teamwork and tackle complex problems, while online resources will support self-paced learning. This multi-faceted approach aims to develop both theoretical understanding and practical skills, preparing students for real-world software development challenges.

4. **List of Practical (C and/or Java*)**
   1. Implement stack, linear and circular queue using arrays.
   2. Implement Infix to postfix conversion using stack
   3. Implement recursive algorithms- Factorial, Fibonacci series and Tower of Hanoi
   4. Implement singly link list.
   5. Construct binary search trees and implement traversal algorithms.
   6. Implement graph traversal algorithms (DFS, BFS).
   7. Implement sorting algorithms (quick sort, merge sort, heap sort).
   8. Implement searching algorithms (sequential, binary search).
   9. Implement hash system (separate chaining, liner and quadratic).
   10. Implement Dijkstra's algorithm for shortest path.
   11. Implement Kruskal's or Prim's algorithm for minimum spanning tree.

*It is strongly recommended that students avoid using the default Java libraries for data structures (e.g., java.util.ArrayList, java.util.LinkedList, java.util.HashMap, java.util.Collections etc.). Instead, students should focus on manually implementing these data structures from scratch. While the use of Java’s built-in libraries for data structures is discouraged during this course, students are encouraged to explore these libraries outside of the course to understand how they are implemented and optimized in Java.

5. **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.

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				

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.

6. Prescribed Books and References

Text Book

- Langsam, Y., Augustin, M.J. and Tanenbaum, A.M: Data Structure Using C and C++, Prentice Hall of India
- Rowe, G.W.: Introduction to Data Structure and Algorithms with C and C++, Prentice Hall of India
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). *Introduction to algorithms*. MIT press.
- Any java base book

Reference books

- Kruse, R. L., & Ryba, A. J. (1998). *Data structures and program design in C++*. Prentice Hall, India..
- Brassard, G., & Bratley, P. (1996). *Fundamentals of algorithmics*. Prentice-Hall, India.

Any Data Structure and Algorithm books using C and/or Java

Pokhara University
Faculty of Science and Technology

CMP 221.3 System Analysis and Project Management (3-0-1)

Course Code	: CMP 221.3	Full Marks	: 100
Course Title	: System Analysis and Project Management	Pass Marks	: 45
Nature of the Course	: Theory and Practical	Total Lecture Hours	: 48
Level	: Bachelor	Program	: BCA

Evaluation:

	Theory	Practical	Total
Sessional (Internal)	40	10	50
Final (external)	50	-	50
Total	100	-	100

1. Course Description

This course covers the essentials of system analysis, design, implementation, and project management. Students will learn about system concepts, the System Development Life Cycle (SDLC), and the role of system analysts. Key topics include requirement analysis techniques, system design principles, input/output design, database design, and system architectures. The course also addresses system implementation processes such as testing and user training, and explores project management tools and techniques for planning, monitoring, and controlling projects. Practical case studies will enhance students' ability to apply these concepts effectively in real-world scenarios.

2. Course Objectives

The general course objectives are to enhance the ability of students with the conceptual of fundamental knowledge about System Analysis Design and Project Management. After completing this course, the students will be able:

1. To understand the fundamental concepts of systems, their characteristics, and the System Development Life Cycle (SDLC).
2. To apply techniques for system requirement analysis, including fact-finding, modeling with DFDs and ERDs, and use case modeling.
3. To grasp the principles of system design, including design principles, input/output design, database design, and system architecture.
4. To learn the processes and challenges of system implementation, including testing, documentation, and user support.
5. To explore project management fundamentals, including project lifecycle stages, planning tools, and documentation.
6. To manage project scheduling, monitoring, controlling, and risk management effectively.

3. Methods of Instruction

The course is taught in English, with faculty members employing a range of teaching methodologies tailored to student needs. These include traditional methods such as lectures and discussions, as well as innovative approaches like Project-Based Learning (PBL), Flipped Classroom, Active Learning, and Gamification. The aim is to engage and motivate students through various instructional strategies. Assessment methods include theoretical and practical evaluations, VIVA, tests, assignments, project work, and terminal examinations, allowing faculty to gauge student understanding comprehensively.

4. **Learning Outcomes**

Upon completion of the course, students will demonstrate a comprehensive understanding of system concepts and the System Development Life Cycle (SDLC), effectively apply requirement analysis techniques to define and model system requirements and utilize system design principles to develop and implement functional systems. They will be proficient in executing system implementation processes, including testing, documentation, and user training, while also addressing implementation challenges. Furthermore, students will apply project management tools and techniques to plan, document, and manage projects, and will be adept at scheduling, monitoring, and controlling projects using methodologies such as Earned Value Management (EVM) and Key Performance Indicators (KPIs).

5. **Course Contents in Details**

<table>
<tr><td colspan="2"><b>Unit-1 Topic: Introduction to System Analysis and Design 5 hours</b></td></tr>
<tr><td><b>Specific Unit wise Objective</b></td><td><b>Course Contents</b></td></tr>
<tr><td>This unit has the following objectives:
1. To introduce the concept of a system, its characteristics, and types (open, closed, physical, abstract).
2. To explain the System Development Life Cycle (SDLC) and its various stages and models (Waterfall, Spiral, Prototyping, RAD, Agile).
3. To highlight the role, skills, and responsibilities of a system analyst and their interaction with stakeholders.
4. To apply the theoretical knowledge to a simple case study, demonstrating basic system analysis and design concepts.</td><td>1.1 Concept of System
1.1.1. Introduction
1.1.2 Characteristics
1.1.3 Types of Systems (Open, Closed, Physical, Abstract)
1.1.4 Subsystem
1.2 System Development Life Cycle (SDLC)
1.2.1 Overview of SDLC
1.2.2 Stage of SDLC
1.2.3 Model of System Development (Overview of Waterfall, Spiral, Prototyping, RAD, Agile)
1.3 System Analyst
1.3.1 Introduction of Programmer and System Analysist
1.3.2 Skills and Responsibilities of System Analyst
1.3.3 Interaction with Stakeholders
1.4 Case Study</td></tr>
<tr><td colspan="2"><b>Unit-2 Topic: System Requirements and Analysis 10 hours</b></td></tr>
<tr><td>This unit has the following objectives:
1. To understand system requirement analysis and its significance.
2. To apply fact-finding techniques like interviews and document reviews.
3. To identify and differentiate functional and non-functional requirements.
4. To create and interpret Data Flow Diagrams (DFD), Entity-Relationship Diagrams (ERD), and Use Case Models.
5. To explore prototyping and methodologies such as RUP and CASE tools.</td><td>2.1 Definitions of System Requirement and Analysis
2.2 Fact-Finding Techniques
2.2.1 Interviews, questionnaires, document review, observation
2.3 System Requirements
2.3.1 Functional and non-functional requirements
2.4 Data Flow Diagrams (DFD)
2.4.1 Introduction
2.4.2 Components
2.4.3 Levels and balancing
2.5 Entity-Relationship Diagram (ERD)
2.5.1 Introduction
2.5.2 Components
2.5.3 Cardinalities and Relationship
2.6 Use Case Modeling
2.6.1 Introduction
2.6.2 Actors
2.6.3 Use case Descriptions and Diagrams
2.7 Prototyping
2.7.1 Introduction and Purposes
2.7.2 Advantages and Limitations
2.7.3 Rational Unified Process (RUP), Computer Aided Software Engineering (CASE), Overview of CASE Approach, Classification of CASE tools.
2.8 Case study</td></tr>
</table>

Unit-3 Topic: System Design and Development	**9 hours**
This unit has the following objectives: 1. To understand the fundamentals of system design and development. 2. To apply basic system design principles such as modularity, abstraction, and reusability. 3. To design effective input and output interfaces including forms, reports, and user interfaces. 4. To grasp key concepts in database design, including normalization and data integrity. 5. To explore various system architectures like client-server, n-tier, and cloud. 6. To learn the processes involved in system implementation, including coding, testing, and deployment.	3.1 Introduction of system Design and Development 3.2 System Design Principles 3.2.1 Basic System Design Principles: Modularity, Abstraction, Encapsulation, Cohesion, Coupling, Separation of Concerns, Reusability, Scalability, Maintainability, Performance, Security 3.3 Input and Output Design 3.3.1 Introduction 3.3.2 Forms, Reports, User interface Design 3.4 Database Design 3.4.1 Introduction 3.4.2 Normalization, Data Integrity, Database Design, DBMS 3.5 System Architecture 3.5.1 Introduction 3.5.1 Client Server, N-Tire, Cloud Architecture 3.6 System Implementation 3.6.1 Introduction 3.6.2 Coding, Testing, Integration, Deployment 3.7 Case Study
Unit-4 Topic: System Implementation and Maintenance	**8 hours**
This unit has the following objectives: 1. To understand the key aspects and processes of system implementation. 2. To learn about software testing, installation, and system documentation. 3. To explore user training, support, and organizational challenges in system implementation. 4. To grasp the fundamentals of system maintenance, including its execution and the upkeep of information systems.	4.1 System Implementation 4.1.1 Introduction of system Implementation 4.1.2 Software Application Testing and Installation 4.1.3 Documenting the System 4.1.4 Training and Supporting the Users 4.1.5 Organization Issues/Challenges in System Implementation 4.2 System Maintenance 4.2.1 Introduction of System Maintenance 4.2.2 Maintaining Information System 4.2.3 Conducting System Maintenance
Unit-5 Topic: Introduction to Project Management	**9 hours**
This unit has the following objectives: 1. To understand the fundamentals of project management, including scope, time, cost, and quality. 2. To learn about the stages of the project lifecycle: initiation, planning, executing, monitoring, and closing. 3. To explore project planning tools and techniques such as Gantt Charts, CPM, and PERT. 4. To understand the Work Breakdown Structure (WBS), its advantages, disadvantages, and creation process. 5. To review project documentation essentials, including the project charter and project plan.	5.1 Introduction, Objective, Importance, Key Elements (Scope, Time, Cost, Quality) 5.2 Project Life Cycle 5.2.1 Introduction 5.2.1 Stage of Project Lifecycle (Initiation, Planning, Executing, Monitoring, Closing) 5.3 Project Planning 5.3.1 Introduction 5.3.2 Tools and Techniques for Project Planning (Gantt Charts, Critical Path Method (CPM), Program Evaluation Review Technique (PERT) 5.4 Project Management Tools 5.4.1 Introduction to Work Breakdown Structure 5.4.2 Advantages and disadvantages 5.4.3 Importance and Creation 5.5 Project Documentation 5.5.1 Introduction 5.5.2 Project Charter 5.5.3 Project Plan

	5.6 Simple Case Study
Unit-6 Topic: Project Scheduling Monitoring and Controlling 7 hours	
This unit has the following objectives: 1. To understand the basics of project scheduling, monitoring, and controlling. 2. To learn time management techniques including task estimation and schedule development. 3. To explore resource management strategies for human, financial, and technological resources. 4. To grasp risk management concepts, including risk identification, analysis, and mitigation strategies. 5. To apply project monitoring and control techniques such as Earned Value Management (EVM) and Key Performance Indicators (KPIs)	6.1 Introduction to Project Scheduling, Monitoring and Controlling 6.2 Time Management 6.2.1 Introduction 6.2.2 Estimating Task Management, Developing Schedules 6.3 Resource Management 6.3.1 Introduction 6.3.2 Managing human, financial and Technological Resources 6.4 Risk Management 6.4.1 Introduction 6.4.2 Identifying Risks, Risk Analysis, Risk Mitigation Strategies 6.5 Project Monitoring Techniques 6.5.1 Introduction 6.5.2 Earned Value Management (EVM), Key Performance Indicators (KPIs) 6.6 Project Control 6.6.1 Introduction 6.6.2 Schedule Control, Scope Control, Change Management

Prescribed Books and References

References Books

1. Jeffery Whitten, Lonnie Bently, *System Analysis Design Methods* 7/E McGraw-Hill
2. Jeffery A. Hoffer, Joey George, Joe Valacich, *Modern System Analysis and Design*, 6/E Prentice Hall India.
3. J.F. George and J. s. Valacich, *Modern System Analysis and Design*, 8/E. Pearson Education Inc., 2017
4. J. Satzinger, R. Jackson & S. Burd, System Analysis and Design in a Changing World, 7/E, Cengage Larning, 2016
5. Murch Richard, *Project Management Best Practices for IT Professionals,* Prentice Hall

6. Evaluation System

Student academic performance is assessed through internal and external methods. Internal evaluation is carried out by faculty members, monitoring day-to-day student progress. On the other hand, external evaluation is conducted by the university, involving semester-end examinations worth 100 marks. This evaluation system operates as follows:

Internal Evaluation	**Weight**	**Marks**	**External Evaluation**	**Marks**
Theoretical Subjects		**50**	**Semester End**	**50**
Daily attendance (Regularity and Punctuality)	10%			
Assignments	20%			
Discipline (Dress, Behavior, outlooks)	10%			
VIVA and Class Performance	10%			
Presentation skill	10%			
Internal Examination	40%			
Practical Subjects		**50**		
Daily attendance (Regularity and Punctuality)	10%			
Assignments	10%			

Discipline (Dress, Behavior, outlooks)	5%			
VIVA and Class Performance	10%			
Presentation skill	5%			
Internal Examination	40%			
Practical Exam/Project Work	20%			
Full Marks: 50 + 50 = 100				

7. Students' Responsibilities

To be eligible for the Semester End Examinations, students must achieve a minimum of 45% marks in internal evaluation including both theoretical and practical sections. Additionally, they must maintain an attendance of at least 80% in their classes. Failure to meet these criteria will result in a "NOT QUALIFIED" (NQ) status, rendering the student ineligible for the Semester End Examinations. Students are strongly advised to attend all classes, formal exams, tests, and fulfill all requirements within the stipulated time frame. It is essential for students to fulfill all course requirements to successfully complete the course.

Pokhara University
Faculty of Science and Technology

Course Code.:	Full marks: 100
Course title: **Web Technology I**	Pass marks: 45
Nature of the course: Theory/Practical/Theory & Practice	Time per period: 1 hour
Year, Semester: Third	Total periods: 45
Level: Bachelor	Program: BCA

1. Course Description

Web Technologies-I is designed to give students a solid foundation in web development. Students will start by exploring the evolution of the web and understanding key concepts like client-server architecture and web protocols. The course will cover HTML and CSS, teaching students how to create and style web pages. They will also dive into JavaScript, learning both basic and advanced features to make web pages interactive. Version control with Git will be introduced to help manage projects effectively. Additionally, students will learn about web design principles, including UI/UX, wireframing, and SEO, to ensure their websites are functional and user-friendly.

This course will be delivered through a combination of lectures, hands-on coding exercises, and practical projects. Students will engage in interactive sessions that involve building real-world web applications and practicing new skills through lecturer and assignments. Each unit will include both theoretical explanations and practical tasks to reinforce learning and ensure students can apply their knowledge effectively.

2. General Objectives

- To help students understand the history and structure of the web, along with basic web protocols and client-server interactions.
- To teach students how to create web pages using HTML, including working with elements, attributes, and various components like lists, links, images, and forms.
- To enable students to style and layout web pages using CSS, covering basics, the CSS box model, responsive design, and advanced layout techniques like Flexbox and Grid.
- To build students' JavaScript skills, including how to write scripts, handle events, and use core programming concepts.
- To introduce students to more advanced JavaScript topics, such as ES6+ features, asynchronous programming, and working with JSON and AJAX.
- To provide an overview of JavaScript frameworks and libraries, including jQuery, React and introduce popular frameworks like Angular, and Vue.js.
- To equip students with essential version control skills using Git and GitHub, including basic commands and collaboration techniques.
- To offer a foundation in web design principles, including UI/UX design, wireframing, typography, color theory, usability, accessibility, and SEO.

3. Contents in Detail

Specific Objectives	Contents
– To understand the evolution of the web, client-server architecture, and the roles of web browsers and servers. – To explain basic web protocols, DNS, and the differences between client-side and server-side scripting languages	**Unit I: Introduction to Web Technologies (4 hours)** 1.1 Evolution of the Web 1.1.1 History of web 1.1.2 WWW 1.1.3 URL 1.1.4 Version of web 1.1.5 Search engine 1.1.6 Website and Web application 1.2 Client-Server Architecture 1.3 Overview of Web Browsers and Web Servers 1.3.1 Web browse, cross browser compabilities 1.3.2 Web server 1.4 Basic Protocols 1.4.1 Short overview of basic protocols 1.4.2 HTTP/HTTPS 1.4.3 HTTP request and resoponse 1.5 DNS and its hierarchy 1.6 Client side and server side scripting language
– To understand role of HTML in web design. – To learn the basics of HTML, including its history, structure, and common tags and attributes. – To create lists, links, images, tables, and forms, and use semantic HTML for better web organization.	**Unit II: HTML (8 hours)** 2.1 Introduction 2.1.1 History and evolution of HTML 2.1.2 Concept of tag 2.1.3 HTML document structure 2.1.4 Html elements and attribbutes 2.1.5 Basic text formatting tags 2.2 Lists, Links and Images 2.2.1 Unordered, ordered and description lists, nested list 2.2.2 Creating hyperlinks, types of link 2.3 Adding images 2.4 Table and Forms 2.4.1 Html table elements and attributes, nested table, mearging row and columns, grouping section of table 2.4.2 common form elements like text fields, textareas, selects, buttons,radio button etc. 2.4.3 input date, time 2.4.4 form methods 2.5 Semantic HTML 2.5.1 Header, footer, nav, article, section, time, progress 2.6 Audio and video elements 2.7 div, span, aside, iframe, canvas

- To provide a comprehensive understanding of styling web pages using CSS, including its syntax, box model, and layout techniques	**Unit III: CSS (8 hours)** 3.1 Need of CSS 3.2 CSS Basics 3.2.1 CSS syntax, selectors, declarations, types, groping selectors, pseudo, comments, 3.2.2 CSS Specificity 3.2.3 Internal vs external CSS 3.2.4 Style text fonts, color, links etc 3.3 CSS Box Model 3.3.1 Box model properties for padding, border, width etc 3.3.2 Centering content with margins 3.3.3 Element backgrounds 3.4 Flexbox and Grid Layouts 3.5 Responsive Design Principles 3.5.1 Working with different screen size 3.5.2 Media queries in CSS 3.6 Brif about CSS Preprocessors (e.g., SASS, LESS)
- To introduce the core concepts of JavaScript, including variables, data types, and control structures, and to show how to use functions, manipulate the DOM, and handle events to make web pages interactive.	**Unit IV: JavaScript Basics (8 hours)** 4.1 Introduction to JavaScript 4.2 Embeeding JavaScript to a webpage 4.2.1 External Script 4.2.2 Internal Script 4.2.3 Using onClick(), onSubmit(), onLoad() etc. 4.3 Variables, Data Types, Operators, Expressions 4.4 Java Script Programming Construct: 4.4.1 Assignment, data declaration, 4.4.2 if, if else, switch 4.4.3 while, for, do while 4.4.4 label, break, continue 4.4.5 Arrays 4.5 Functions and Scope 4.6 Dialog boxes 4.7 Document Object Model 4.8 Event handling.
- To dive into advanced JavaScript features such as ES6+ syntax, promises, and async/await, and to understand how to handle errors and work with JSON and AJAX for more sophisticated web interactions.	**Unit V: Advanced JavaScript Concepts (6 hours)** 5.1 ES6+ Features: 5.1.1 Let, Const, Arrow Functions, Promises, etc. 5.1.2 Objects 5.2 Regular Expressions

	5.3 Asynchronous Programming (Async/Await) 5.3.1 Promises 5.3.2 Async/Await 5.3.3 Fetch 5.4 Error Handling 5.5 Introduction to JSON 5.5.1 JSON Syntax and Structure 5.5.2 Simple Example 5.6 Introduction to AJAX
- To familiarize yourself with JavaScript libraries like jQuery for easier DOM manipulation and to explore popular frameworks/library such as React, Angular, and Vue.js.	**Unit VI: Introduction to JS Frameworks and Libraries (5 hours)** 6.1 Jquery() 6.1.1 Why jQuery 6.1.2 Jquery fundamentals 6.1.3 Manipulating DOM elements. 6.2 Overview of Popular Frameworks/Library (e.g., React, Angular, Vue.js) 6.3 Choosing the Right Framework for Projects
- To explore the principles of UI/UX design, including wireframing and responsive design, and to apply concepts of typography, color theory, usability, accessibility, and SEO to create well-designed, user-friendly websites	**Unit 8: Web Design Concepts (6 hours)** 8.1 Principles of User Interface Design 8.2 Introduction to UI/UX Design 8.2.1 Wireframe 8.2.2 User Story 8.2.3 Responsive design 8.2.4 Introdiction to Tools for Wireframing 8.2.5 Low-Fidelity and High-Fidelity Wireframes 8.3 Typography and Color Theory 8.4 Usability and Accessibility Guidelines 8.5 SEO

4. Methods of Instruction

The course uses lectures, discussions, group work, lab sessions to blend theoretical knowledge with practical application, ensuring students gain both understanding and hands-on experience in web development. Assignments will be carefully designed by the teacher focusing real-world projects to reinforce learning, with clear instructions and deadlines.

5. List of Practical

Students should perform following HTML lab tasks:

- In the head section of a web page, make sure to:
  - Add a title tag to set the page’s title, which appears in the browser tab.
  - Link to external stylesheets using relative paths to apply the desired styles.

    - Include meta tags to specify important information such as the character encoding, description, keywords, author, and viewport settings for responsive design.
- In the body of a web page, include:
    - A table with headers, rows, and cells, adjusting attributes for spanning and sizing.
    - Text, images, sound clips, and videos with proper controls, sizing, and alternate text.
    - The `<div>` tag for styling and layout.
    - Text elements using tags like `<h1>`, `<p>`, and `<li>`.
    - Classes and styles for formatting, including lists.
- Create hyperlinks from text and images to:
    - bookmarks on the same page
    - other locally stored web pages
    - a website using the URL
    - send mail to a specified email address
    - to open in a specified location (the same window, a new window)
- Create form for:
    - Login and signup purpose
    - Design registration form including Textbox, TextArea, Select, Checkbox, Radio button, Button elements and group them using fieldset and legend.
- Use <div><span> elements to break down in different blocks.

CSS Labs

- Set background properties including colour and images, font properties, style table including size, background colour, horizontal and vertical alignment, spacing, padding, borders.
- Create external styles to be tagged in a web page including h1, h2, h3, p, li.
- Attach comments to an external stylesheet.
- Center a div with its elements.
- Design navigation menu for web page with proper stylling, when user put mouse over the menu the menu should be hilighted.
- Apply css to image, links, list
- Create product grid including product name, price, image, add to cart button and add effects when user put crouser on the product. Also make it responsive.

JavaScript:

- Link an external JavaScript file and manipulate DOM elements.
- Add and remove textbox from web page with add and delete button.
- Write a script to input information like Name, Address, Contact, Age abd display them in proper format when user clicks the submit button.
- Validate form inputs with JavaScrips
    - Validate login and signup form. the password should contain atleast letters, symbole and number and should be minimum 10 charactors, username field cannot be blank, to sign up user must be audult.

- Load another page inside a page when user click on load button without reloadind a page.
- Manupulate DOM emelents using jQuery.

VCS:

- Start a Git repository with **git init**, stage files with **git add**, and commit with **git commit** make some changes and push with **git push**

UI/UX:

Prepare Low-Fidelity wireframe of products and product details page.

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	5
		Lab tasks	10
		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.

7. Prescribed Books and References

Text Book

Reference Books

Felke-Morris, T. (2020). *Web development and design foundations with HTML5*. Pearson.
Simpson, K. (2021). *You don't know JS yet: ES6 & beyond*. Independently published
Chacon, S., & Straub, B. (2014). *Pro Git*. Apress.

Pokhara University
Faculty of Science and Technology

CMP 350.3

Course code	: CMP 350 (3 Credits)	Full marks	: 100
Course title	: Operating Systems	Pass marks	: 45
Nature of the course	: Theory and Practical	Total lectures	: 48 hours
Level	: Bachelor	Program	: BCA

Evaluation:

	Theory	Practical	Total
Sessional (Internal)	30	20	50
Final (external)	50	-	50
Total	80	20	100

1. Course Description

This course is designed to provide the concepts of operating system to various system environments. It cinites system software, internal structures, functions and security features of operating system to run the application software and perform various tasks with optimized throughput. This course also introduces the emerging new trended operating system for distributed environment like cloud and mobile systems. After completion of this course, students can select, apply and implement the operating system features and functions for the best utilization of the device specification.

2. General Objective (limit to only 3 objectives)

- To acquaint the students with structure of operating systems and their functionality.
- To acquaint the students with basic concepts of resource allocation and management.
- To develop the skills in students to select and measures the optimal resource allocation schedules.
- To acquaint the students with the knowledge of process and thread, I/Os, Memory, CPU, disk management, network security systems.
- To acquaint the students with basic concepts of operating systems new trends such as real-time, distributed, cloud systems and mobile.

3. Methods of Instruction

- Lecture
- Tutorial
- Practical and case studies

4. Contents in Detail

Specific Objectives	Contents
• Familiarize with basic concepts of Operating systems, and its structures. • Understanding the features of generation and types of operating systems.	**Unit 1: Basics of Operating System** [6 hours] 1.1 Introduction, goals of operating system 1.2 Operating-System Structures and functions 1.3 Types of Operating Systems *Batch Processing, Multiprogramming, Multiprocessing, Networking, Real Time (RTOS), Distributed, Embedded system* 1.4 System Interface, System Calls 1.5 Virtual Machine

• Familiarize with Task, Process and threads • Implement of resource allocation techniques • Understanding the mutual exclusion for resource utilization	**UNIT- II: Process, Threads and Scheduling** [13 hours] **2.1 Process** 2.1.1 The Process Model 2.1.2 Process Creation and Process Termination 2.1.3 Process Hierarchies 2.1.4 Process States 2.1.5 Implementation of Processes **2.2 Threads** 2.2.1 Thread Usage 2.2.2 Threads Models (Many-to-one model, One-to-One Model, Many-to many model) 2.2.3 User Space and Kernel Space Threads 2.2.4 Hybrid Implementations 2.2.5 Difference between Threads and Processes **2.3 Inter-process Communication** 2.3.1 Race Conditions 2.3.2 Critical Regions 2.3.3 Mutual Exclusion with Busy Waiting 2.3.4 Sleep and Wakeup, Semaphores, Mutexes, Monitors, Message Passing 2.3.9 Avoiding Locks: Read-Copy-Update **2.4 Process Scheduling** 2.4.1 Basic Concept 2.4.2 Type of scheduling (Preemptive scheduling, Non-preemptive scheduling, batch, Interactive, real time scheduling), 2.4.3 Scheduling Criteria or performance analysis, Scheduling Algorithm (Round-robin, First come first served, Shortest-job- first, Shortest process next, Shortest remaining Time next, real time, priority fair share, guaranteed, Lottery scheduling) **2.5 Deadlocks** 2.5.1 System Resources: Preemptable and Non-preempable 2.5.2 Method of handling Deadlocks, 2.5.3 Deadlock prevention 2.5.4 Deadlock avoidance: Banker's Algorithm, *2.5.5* Protection- *System Protection, Goals of Protection, Principles of Protection, Domain of Protection, Access Matrix, Implementation of Access Matrix, access Control list* **2.6** Research on Processes and Threads
• Conceptualize the role and working procedure of memory • Familiarizing with virtual memory management • Understanding the page replacement algorithms • Understand the mechanism file and filing	**UNIT- III: Storage Management** [13 hours] **3.1 Memory Management** 3.1.1 Logical & physical Address Space 3.1.2 Swapping 3.1.3 Contiguous Allocation 3.1.4 Paging, Structure of Page Table 3.1.5 Segmentation, Segmentation with Paging **3.2 Virtual Memory** 3.2.1 Background 3.2.2 Demand Paging, 3.2.3 Performance of Demanding Paging, 3.2.4 Page Replacement, Page Replacement Algorithms,

	Allocation of Frames, 3.2.5 Thrashing. **3.3. File System Interface and Implementation** 3.3.1 File System Interface- *The Concept of a File, Access methods, Directory Structure, File System Mounting, File Sharing, Protection* 3.3.2 File System Implementation- *File System Structure, File System Implementation, and Allocation methods, Free-space Management, Directory Implementation, Efficiency and Performance*
• Understand the role of input/output devices • Understand the different approaches for optimal output	**Unit IV : Input/output Management** [7 hours] 4.1 Principles of I/O Hardware- *I/O Device, Device Controller, Memory Mapped I/O, Direct Memory Access* 4.2 Principles of I/O Software- *Goals of I/O Software, Polled I/O verses Interrupt Driven I/O, Character User Interface and Graphical User Interface, Device Driver, Device Independent I/O Software, User-space I/O Software* 4.3 Mass Storage Structure - Overview of Mass Storage Structure, Disk Structure, Disk Attachment, Disk Scheduling, Disk Management, Swap space Management Redundant Array of Inexpensive Disks , RAM Disks, Optical Disk
• Visualize of operating system structure and functionality	**Unit V : Case Study** [5 hours] **5.1 Linux -** Design principles, Inter-process communication, Kernel modules, Network structure, and Security in Linux/Windows **5.2 Windows -** Design principles, Programmer interface, System components, Security level in Linux/Windows
• Familiarize with current trends of operating systems	**Unit VI : New Trend in Operating System** [4 hours] 6.1 Concept, character and role of Distributed, Cloud, Mobile and Multimedia operating systems 6.2 Memory wall and bottleneck for operating system

5. Practical Works

Laboratory work should cover the operating system structure and functions of any two popular operating system. It also insist the students to design model of operating system with the reference of open source guideline. Students should complete the following tasks in laboratory:

1. Understanding and running all the internal command and external commands in Microsoft Disk Operating Systems.
2. Installation and user, application management in Windows (current version)
3. Simulation of Process Scheduling Algorithms
4. Simulation of Page Replacement Algorithms
4. Simulation of Disk Arm Scheduling Algorithms
5. System Administration (user, disk, role, etc.) in any open source operating system.

6. Evaluation System and Students' Responsibilities

Evaluation System

The internal evaluation of a student may consist of assignments, attendance, internal assessment, labreports, 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.

7. Prescribed Books and References

Books

1. Andrew S. Tanenbaum, Herbert Bos, *"Modern Operating Systems"*, Pearson, 4th Edition, ISBN-10: 0-13-359162-X. 2011

References

1. A. Silberschatz, P.B. Galvin, G. Gagne *"Applied Operating System Concepts*", Wiley, 8th Ed.
2. D. M. Dhamdhere , "*System Programming and Operating System*" - Tata McGraw-Hill, 20
3. Andrew S. Tanenbaum, *"Distributed Operating System*", Pearson
4. Naresh Chauhan, Principles of Operating Systems, Oxford University Press 2014, Oxford University Press, ISBN-13: 978-0-19-808287-3

Pokhara University
Faculty of Science and Technology

Course Code.: CMP 323	Full marks: 100
Course title: **Software Engineering**	Pass marks: 45
Nature of the course: Theory/Practical/Theory & Practice	Time per period: 1 hour
Year, Semester:……………	Total periods: 45
Level: Bachelor	Program: BCA

1. Course Description

This course includes the topics that provide fundamental concept and standard of software engineering so that students will be able to develop software and/or handle software project using the global standard of software.

2. General Objectives

To provide the students with the basic competencies required to identify requirements of developed system.

To understand and document the system design and maintain a developed system.

3. Contents in Detail

This section contains the details to be taught under the course. Normally, the contents can be divided into 5 to 8 units, and each unit can be assigned 5 – 8 teaching hours.

Course Detail

Specific Objectives	Course Content
• Define Software and sotware characteristics • Various types of software. • Explain various characteristics of software. • Explain the attributes of good software and software engineering. • Identify key challenges that software engineering are facing. • Explain professional practice.	**Unit1 : Introduction (4 hrs)** 1.1 Introduction to Software, Types of Software, Characteristic of Software, Attributes of Good Software. 1.2 Introduction to Software Engineering, Costs, Key Challenges that Software Engineering Facing 1.3 Different between System Engineering and Software Engineering, Professional Practice.
• Define Software process. • Describe the waterfall model, evolutionary development. • Describe component based software engineering and process iteration, incremental delivery and software development and also describe their advantages and disadvantages. • Describe the agile methods and its importance of extreme programming.	**Unit 2 : Software Development Process Model (8 Hrs)** 2.1 Software Process, Software Process Model: The Waterfall Model, Evolutionary Development, Component Based Software Engineering(CBSE), Process Iteration, Incremental Delivery, Spiral Development, Rapid Software Development, Agile Methods, Extreme Programming, Rapid Application Development 2.2 Software Prototyping; Rational Unified Process (RUP), Computer Aided Software Engineering (CASE); Overview of CASE Approach, Classification

• Explain rapid application development. • Describe software prototyping and rational unified process. • Explain computer aided software engineering and the various CASE tools.	of CASE tools.
• Describe system and software requirements and the types of software requirements. • Explain the functional and nonfunctional requirements • Explain requirement validation and requirement specification. • Define feasibility.	**<u>Unit 3 : Software Requirement Analysis and Specification (10 Hrs)</u>** 3.1 System and Software Requirements, Types of Software Requirements: Functional and Non Functional Requirements, Domain Requirements, User Requirements, Elicitation and Analysis of Requirements. 3.2 Overview of Techniques, View Points, Interviewing, Scenarios, Use-Case Ethnography, Requirement Validation, Requirement Specification, Feasibility.
• Define software design and concept • Describe architectural design. • Explain repository model,client server model,layered model • Explain modular decomposition • Explain procedural design using structural methods. • Explain user interface design and human computer interaction. • Describe design notation.	**<u>Unit 4 : Software Design (10 Hrs)</u>** 4.1 Design Concept; Abstraction, Architecture, Patterns, Modularity Cohesion, Coupling Information Hiding ,Functional Independence,Refinement 4.2 Architectural Design Repository Model Client Server Model , Layered Model, Modular Decomposition; 4.3 Procedural Design Using Structural Methods, User Interface Design, Human Computer Interaction, ,Information Presentation , ,Interface Evaluation ,Design Notation
• Define coding and programming languages used for software development. • Explain various development tools And good programing practices.	**<u>Unit 5 : Coding (2 Hrs)</u>** 5.1 Programming Language and Development Tools 5.2 Selecting Languages and Tools Good Programming Practices
• Explain software testing. • Differentiate verification and validation. • Explain various techniques of testing and various level of testing. • Design test cases. • Define product and its process. • Explain quality standards. • Describe capability maturity model.	**<u>Unit 6 : Software Testing and Quality Assurance (6 Hrs)</u>** 6.1. Verification and Validation , Techniques of Testing; Black-box and White-box Testing, 6.2. Inspections 6.3. Level of Testing ; Unit Testing, Regression Testing; **6.4.** Design of Test Cases, Quality Management Activities, Product and Process ,Quality Management Activities , Product and Process, Quality Standards, ISO9000 , Capability Maturity Model(CMM);
• Explain evolution nature of software and different types of maintenance. • Explain functionality addition or modification. • Describe re-engineering.	**<u>Unit 7 : Software Maintenance (3 Hrs)</u>** 7.1. Evolving Nature of Software, Different Types of Maintenance ; Fault Repair, Software Adaptation 7.2. Functionality Addition or Modification; Maintenance Prediction, Re-Engineering, Configuration Management (CM), Importance of CM, Configuration Items, Versioning;

• Define configuration management. • Explain importance of configuration management.	
• Explain the need for the project management in software project. • Explain management activities and • need of project scheduling. • Explain risk management.	**Unit 8: Management Software Projects (2 hrs)** 8.1. Needs for the Proper Management of Software Projects, Management Activities; Project Planning Estimating Costs, Project Scheduling, Risk Management, Managing People.

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

4. Methods of Instruction

Specify the main methods of instruction such as lecture, discussion, group work, lab work, field visit, etc., pedagogical approach including the rationale for why students will benefit from the course.

5. List of Practical

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	5
		Field visit and field report	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.

7. Prescribed Books and References

Text Books

1 Roger S.Pressman, "Software Engineering: A Practitioner 's approach", 6th Edition , McGraw Hill International edition,2005

Reference Books

1 Ali Behforooz , and Frederick J.Hudson, "Fundamental of Software Engineering", OUP,1996

2 Ian Sommerville," Software Engineering", 9th Edition Addison-Wesley, 2010.

3 Rajib Mall," Fundamental of Software Engineering",3rd Edition, 2010.

4 Pankaj Jalote, "An Integrated Approach to Software Engineering ", 2nd Edition, Springer, 1997.

Pokhara University
Faculty of Science and Technology

CMP 226.3 Database Management System (3-0-3)

Course code	: CMP 226 (3 Credits)	Full marks	: 100
Course title	: Database Management System	Pass marks	: 45
Nature of the course	: Theory and Practical	Total lectures	: 48 hours
Level	: Bachelor	Program	: BCA

Evaluation:

	Theory	Practical	Total
Sessional (Internal)	30	20	50
Final (external)	50	-	50
Total	80	20	100

1. Course Description

This course is designed to encompass the fundamental concepts of database management system. These concepts include the concept of database design, database languages and the database system implementation. This course presents the introductory concepts of database security, query processing and optimization, transactions and their concurrency control and recovery policies. This course also introduces the emerging new trended databases such as NoSQL and Blockchain databases. After completion of this course, students can design and implement a database system to develop a software application.

2. General Objective

- To acquaint the students with the knowledge of relational database design using ER Model.
- To develop the skills in students to design normalized relational database required for a specified application.
- To acquaint the students with the knowledge of database security, query processing and optimization, files and record organizations, transaction, concurrency control, data recovery mechanisms.
- To acquaint the students with concepts of NoSQL databases and Block-chain Databases.

3. Methods of Instruction

- Lecture
- Tutorial/Discussion/Readings
- Practical works and Project works.

4. Contents in Detail

Specific Objectives	Contents

• Familiarize with basic concepts of database systems, and its architecture. • Explain the concept for database components and their roles	**Unit 1: Introduction** [7 hours] 1. Database Management System purpose and applications, 2. Database Systems vs File Systems, Storage Management and Indexing 3. View of Data- Data Abstraction (Physical, logical and view level, Data Independence) 4. Instances and Schemas, 5. Database Languages (DDL, DML and DCL) 6. Database and Application Architecture- Database System Architecture and Database Application Architecture (two-tier and three-tier)
• Familiarize with ER Model and Relational Model and relational algebra. • Design the relational database using ER model.	**Unit 2: ER and Relational Model** [8 hours] 1. Introduction to ER Model: 2. Entity sets, attributes and values, Relationship sets- participation, entity's role, descriptive attributes, degree of relationship set, Mapping Cardinalities, Attributes- simple, composite, single-valued, multi-valued, derived, Entity-Relationship (ER) Diagram, Specialization, Generalization, and Aggregation 3. Key and its types 4. Reducing ER diagrams to Relational Schema 5. Structure of Relational Databases, Database Schema, Schema Diagrams 6. Relational Algebra
• Implement and write DDL and DML queries in the SQL.	**Unit 3: Structured Query Language** [6 hours] 1. Structured Query Language (SQL)- SQL DDL and DML 2. Basic Structure of SQL Queries, DDL queries, Basic Operations (Rename, String, Attribute Specification in the select clause, order by, where-clause), Set Operations, Null values, Aggregate Functions, 3. Nested Queries, 4. Join Expressions (Natural Join, Join Conditions, Outer Joins), 5. Views, stored Procedures

• Apply the integrity constraints to implement database securities. • Normalize the database to a defined normal form.	**Unit 4: Relational Database Design** [6 hours] 1. Integrity constraints- Domain Constraints, Entity Integrity Constrains, Referential Integrity Constraints, Assertions and Triggers 2. Features of Good Relational Designs 3. Functional dependencies and Armstrong's Axioms 4. Closure of a Set of Functional Dependencies and Closure of Attribute Sets 5. Database Normalization and Normal Forms- 1NF, 2NF, 3NF and BCNF 6. De-normalization for Performance
• Understand the mechanism of query processing and need of query optimization. • Familiarize with different file organization methods.	**Unit 5 : Query Processing and Optimization** [5 hours] 1. Introduction to Query Processing 2. Equivalence of Expressions 3. Query Cost Estimation 4. Query evaluation and execution plan 5. Query Optimization
• Understand the concepts of transactions, need of atomicity, durability and isolation, serial schedule. • Familiarize with the need of Concurrency control.	**Unit 6 : Transactions and Concurrency Control** [5 hours] 1. Transaction Concepts 2. Transaction Model and State Diagram 3. ACID properties of transaction 4. Serializability- conflict and view serializability 5. SQL Standard Isolation Levels 6. Concurrency Control- Lock-Based Protocols
• Understand the recovery algorithms and techniques to protect and recover the data from various failures. • Understand the need of database security. • Understand the role of access control, authorization, views and encryption mechanism to provide database security.	**Unit 7 : Security and Crash Recovery** [6 hours] 1. Security and integrity violations 2. Access control and authorization, views 3. Failure classification 4. Recovery and Atomicity- log records, database modification, concurrency control and recovery, transaction commit, Redo and Undo Transactions using Log, Check Points, 5. Recovery Algorithm Using Log Records- Transaction Rollback, Recovery after a System Crash, Optimizing Commit Processing

• Understand the concepts of NoSQL, ODBMS, Distributed Databases	**Unit 8 : Emerging Trend in Databases** [5 hours] 1. NoSQL Databases- Characteristics, Categories, Advantages 2. Object Oriented Database and ORM 3. Parallel and Distributed Databases 4. Data Warehouse and Data Mining

5. Practical Works

Laboratory work of 45 hours per group of maximum 24 students should cover the database design, use of database languages and database system implementation using any relational database management system such as MS SQL or MySQL or Oracle etc. Students should complete the following tasks in laboratory:

SN	Tasks to Complete
1	Introduction to MS SQL (or any RDBMS), its datatypes and its installation.
2	All SQL DDL operations studied in Unit 2 such as creating database, creating tables, delete database, drop table, alter etc.
3	All SQL DML operations studied in Unit 2 such as database modification operation- insert into, delete, update etc.
4	Implementing Join Expressions (Natural Join, Join Conditions, Outer Joins)
5	Implementing Stored Procedures.
6	Illustration and implementation of Views.
7	Implementing Integrity constraints (Domain Constraints, Entity Integrity Constrains, Referential Integrity Constraints)
8	Implementing Assertions and Triggers.
9	Implementation of Database Security and Privileges: Grant and Revoke Commands, Commit and Rollback Commands.
10	Connecting database with connection string using any standard programming language and executing SQL queries.

Students should submit a project work that uses all the knowledge obtained from this course to design and implement a database system for any application that students chose. The students should design the database using ER model and present using ER diagram which are then reduced to relational schema. The students should apply the constraints studied in this course including triggers. The database should be in at least 3NF. The marks for the practical evaluation must be based on the project work submitted by students.

6. 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. Silberschatz, A., Korth, H. F., & Sudarshan, S. (2011). *Database system concepts.* McGraw Hill.

References

1. Majumdar, A. K., & Bhattacharyya, P. (1996). *Database Management Systems*. McGraw-Hill.
2. Elmasri, R., & Navathe, S. B. (1994). Fundamentals of Database Systems Benjamin Cummings. *Redwood City, CA*.

3. Everest, G. C. (1986). *Database management*. McGraw-Hill, Inc.

Pokhara University
Faculty of Science and Technology

Course Code.:	Full marks: 100
Course title: **Computer Graphics and Multimedia Technology**	Pass marks: 45
Nature of the course: Theory/Practical	Time per period: 1 hour
Year, Semester:II/IV	Total periods: 45
Level: Bachelor	Program: BCA

1.Course Description

To introduce the students to the basics of Graphics hardware , various drawing algorithms ,animation and multimedia at the industry level.

2.General Objectives

To introduce the hardware and concept of object rendering.
To introduce the concept of 2D and 3D Graphics Algorithm
To familiarize students with the knowledge of multimedia and animation.

3. Contents in Detail

Specific Objectives	Contents
Familiarize with the basic Graphics hardware and interfacing and software	Unit 1: Introduction to Graphics (6 hrs) 1.1 Application of Computer Graphics 1.2 CRT Monitor working 1.3 Display Technology (Raster and Random) 1.4 Color Generation Technique in CRT 1.5 Frame Buffer Organization and Video Controller 1.6 Recent Trend virtual Reality and Augmented Reality
Scan conversion of Graphics Primitive line and Circle.	Unit 2: Scan Conversion (6 hrs) 2.1 Importance of Scan Conversion 2.2 Line Drawing Method: DDA and BLA 2.3 Mid-point Circle Generation Algorithm
Importance of Basic and Advance transformation of solid body.	Unit 3: Two Dimensional Transformation and Window Viewing (7 hrs) 3.1 2D Geometric Transformation (Translation, Rotation, Scaling) 3.2 Homogeneous Coordinate Representation 3.3 Successive Transformation 3.4 Window to Viewport Transformation 3.5 Cohen Sutherland Line Clipping Algorithm

Introduce the concept of 3D Graphics and 3D object representation projection and Hidden Suraface Removal	Unit 4: Three Dimensional Graphics and Hidden Surface Removal (6 hrs) 4.1 3D Geometric Transformation 4.2 3D Object Representation- Polygon Surface and Polygon Mesh 4.3 Parallel and Perspective Projection 4.4 Hidden Surface Removal 4.5 Back Face Detection, Z-Buffer Method, Scan Line Method
Effect of Light in a computer Scene and Shading techniques	Unit 5: Illumination Model and Shading (6hrs) 5.1 Basic illumination Model 5.2 Specular Reflection and Phong Model 5.3 Polygon Rendering Methods (Flat, Goraud and Phong shading)
Introduction to Multimedia and media representation and various different kind of authoring tools and Animation	Unit 6: Multimedia System and Representation (8 hrs) 6.1 Introduction to Multimedia 6.2 Different kind of Image Format 6.3 Sound and Audio 6.3.1 Digital Audio Formats 6.3.2 MIDI hardware, software and Messages 6.4 various Multimedia Authoring Tools
Introduction to video and various compression Technology and Animation	Unit 7: Multimedia Compression Technique (6 hrs) 7.1 Video 7.1.1 Digital Video Formats(AVI and MOV) 7.2 Compression Techniques for Audio and video (Spatial and Temporal) 7.3 Animation and its Types

4.Methods of Instruction

The method of instruction should be lectures assisted with practical , tutorial and group work project.

5. List of Practical

1. To draw a straight line using Digital Differential Analyzer and Bresenham's Line Drawing Algorithm
2. Digitize a circle using mid-point circle Drawing Algorithm
3. Translation,Scaling and Rotaion in 2D
4. 2D Screen Viewing Transformation
5. To draw a graphical output primitives using Open Gl
6. To implement and test the Image Compression Algorithhm (JPEG Compression)

6.Evaluation System and Students Responsiblities
In addition to the formal end-semester exam(s) ,the interal (fromative) evaluation of a student may consists of tutorial,assisgments, lab reports , projects, class participation and presentation etc.

Internal Evaluation	Weight	Marks	External Evaluation	Marks
Theory		30		
Attendance and Class Participation		10%		
Assignments		20%		
Presentation		10%		
Internal Assessment		60%		
Practical		20		
Attendance and Class Participation		20%	End Semester Examination	50
Lab Report/Project Report		30%		
Mini-Project		50%		
Total Internal		50		

7.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. Hearn D & Baker M.P (1997) Computer Graphics, C Version . Pearson Education India
2. Buford, J.F.K(2002), Multimedia Systems, Pearson ,India

Reference Books

1.Malay K. Pakhira Computer Graphics Multimedia and Animation ,Second Edition PHI Publication

Pokhara University
Faculty of Management Studies

Course Code.: MTH 320	Full marks: 100
Course title: **Probability and Statistics (3-0-0)**	Pass marks: 45
Nature of the course: Theory & Practice	Time per period: 1 hour
Semester:……………	Total periods: 45
Level: Bachelor	Program: BCA

1. Course Description

This course aims to provide Bachelor of Computer Application (BCA) students with a solid understanding of applied statistics, as well as practical skills. It covers statistical concepts including data presentation, measures of central tendency, dispersion, correlation and regression analysis, probability, probability distributions, sampling, estimation, and hypothesis testing. The course focuses on hands-on learning, with laboratory sessions where students can use statistical analysis software tools. Having a mastery of these statistical tools is essential for creating effective dashboards in computer applications, software, web development, and research projects.

2. General Objectives

The course is designed with the following general objectives:

- To provide students with a fundamental understanding of Statistics, Probability, and Sample Survey techniques, particularly in relation to Computer Applications.
- To equip students with the ability to compute various statistical measures of central tendency and dispersion for given data sets.
- To develop students' skills in utilizing statistical tools and techniques that are relevant to computer applications and research work.
- To familiarize students with statistical software used for data analysis, such as Microsoft Excel, SPSS, or R Studio.
- To enable students in interpreting statistical values and results.
- To make students competent in understanding and applying the concepts of hypothesis testing and significance tests.

3. Contents in Detail

Specific Objectives	**Contents**
• Acquaint fundamental concepts of statistics, its broad applications in various fields. • Familiarize different types of variables and data. • Make knowledgeable about the various scales of measurement and their significance in statistical analysis. • Equip with the understanding of key elements of statistical analysis.	**Unit I: Introduction (3 hours)** 1.1 Introduction to statistics, 1.2 History and its scope of statistics 1.3 Variable and data, types of variables (quantitative and categorical), types of data (quantitative and qualitative), measurement and scaling, types of scale(nominal, ordinal, interval and ratio), fundamental elements of statistical analysis.

• Enable in distinguishing between primary and secondary data sources and understand their uses in research. • Familiarize with various methods of primary data collection, • Acquaint with the fundamental principles and approaches of experimental and survey research methods. • Develop skills in preparing data for analysis	**Unit II: Data collection (4 hours)** 2.1 Sources of data (Primary and Secondary) 2.2 Primary data collection methods (Field survey, Observation, Experiment) and tools (Questionnaire, Observation Checklist. 2.3 Introduction to Experimental research and survey research 2.4 Data preparation - editing, coding, and transcribing
• Enhance proficiency in organizing and presenting data through the utilization of diverse tabular formats including both manually and the utilization of computer software. • Equip with the skills to effectively visualize data using different graphical methods both manually and using computer software.	**Unit III: Presentation of data (4 hours)** 3.1 **Tabular Presentation:** Steam-and-leaf display, frequency distribution, relative frequency distribution, cumulative frequency distribution, bivariate frequency distribution. 3.2 **Data visualization**: Bar diagram(simple, multiple, sub-divided, percentage), pie chart, histogram, frequency polygon, frequency curve, ogive and time-plots.
• Enable to compute and interpret various measures of central tendency and perform these calculations using computer software. • Make proficient in calculating and interpreting partition values both manually and through computer software. • Develop ability to compute and interpret measures of variation both manually and computer software. • Familiarize and calculate with the concepts of distribution shape	**Unit IV: Summary Measures (5 hours)** 4.1 **Measure of central tendency**: mean (Arithmetic, Geometric and Harmonic), median, mode and mid-hinge 4.2 **Partition Values:** Quartiles, Deciles, Percentiles 4.3 **Measures of variation:** range, interquartile range, semi-inter quartile range standard deviation, Variance and coefficient of variations. 4.4 **Shape of the distribution:** five-number summary, box-and-whisker plot, concept of skewness and kurtosis.
• Familiarize with the basic concepts of probability. • Equip with an understanding of fundamental probability rules and apply them in different scenarios of problem solving. • Make knowledgeable about the concept of conditional probability and Bayes' Theorem, and to enable to apply these concepts in problem-solving.	**Unit V: Probability (5 hours)** 5.1 **Introduction of Probability:** Basic concepts, Counting rules, Objective and subjective probability, Marginal and joint probability 5.2 **Basic Probability:** Addition rule, Multiplication rules, 5.3 **Conditional Probability:** Concept of Conditional Probability, Bays' Theorem
• Introduce to random variables and distinguish between discrete and continuous random variables. • Familiarize with probability distributions. • Learn Binomial, Poisson, and Normal distributions, including the normal approximation of binomial and Poisson distributions.	**Unit VI: Discrete Probability Distribution (10 hours)** 6.1 **Random variables**: Introduction to random Variable, types of random variable (discrete and continuous), Mean and standard deviation of discrete random variables, Mathematical expectation of discrete random variable 6.2 **Probability distribution:** Concept of probability distribution, types of probability distribution (discrete and continuous), Probability mass function, Probability density function, probability distribution function,

	Binomial distribution, Poisson distribution and Normal distribution, Normal approximation of binomial and Poisson distribution
• Acquaint with the concepts of population and sample, and enable to understand and apply various sampling methods in statistical analysis. • Calculate and interpret confidence intervals for means and proportions, • Gain hands-on experience using computer software to analyze and interpret data.	**Unit VII: Sampling and Estimation of Population Parameters (5 hours)** 7.1 **Sampling:** Population and sample, different types of sampling methods, 7.2 **Estimation:** Law of large numbers, Central limit theorem, point and interval estimation, Statistical confidence, Confidence intervals, Confidence interval estimation for mean and proportion.
• Provide with an understanding of hypothesis testing, including steps involved, and the connection between confidence intervals and hypothesis testing. • Equip with skills to conduct and interpret tests for a single mean and double means using z-tests and t-tests both manually and using computer software. • Execute and interpretate tests for a single proportion and comparing two proportions. both manually and using computer software. • Acquaint with the Chi-square test for independence and gain practical experience with computer software.	**Unit VIII: Hypothesis Testing (6 hours)** 8.1 **Introduction to Testing of hypothesis: Statistical hypothesis, steps in hypothesis testing,** P-value approach to hypothesis testing, Connection between confidence intervals and hypothesis testing, 8.2 **Test of significance of mean:** Test of single mean, comparing two means (z and t-test) 8.3 **Test of proportion:** Test of single proportion, Comparing two proportions 8.4 **Test of independence using Chi-square test**
• Analyze correlations using scatter diagrams, Pearson's correlation coefficient, and Spearman's rank correlation. • Performing simple linear regression analysis using the least squares method to fit models. • Calculate and interpret the Coefficient of Determination and Standard Error, and analyze regression output with the help of statistical software.	**Unit X: Correlation and Regression Analysis** (**6 hours)** 9.1 **Correlation Analysis**: Scatter diagram, Correlation Coefficient and its Properties, Spearman’s rank correlation, Pearsons correlation coefficient. 9.2 **Simple Linear Regression Analysis:** Model fit using least square method, Coefficient of Determination and Standard Error, interpretation of output of regression analysis using statistical software

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

4. Methods of Instruction

This course will consist of lectures, discussions, and lab work. The lectures will focus on providing theoretical computational knowledge in statistics and probability, while the discussions will encourage students to apply the material within the context of computer science. The lab work is crucial as it allows students to practice statistical concepts using various software tools such as Microsoft Excel, SPSS, R Studio, or Python. This blended approach aims to enhance students' understanding of the theory and foster the development of practical skills for data analysis in real-world applications.

6. Praticals

Evaluation System

5. 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
		Practical evaluation	10
		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.

6. Prescribed Books and References

Text Book

1. Levin, Richard I. And David S. Rubin: Statistics for Management, Prentice-Hall of India

Reference Books

1. Fundamentals of Mathematical Statistics – 1 st Edition S.C.Gupta, V.K. Kapoor and S Chand .
2. Introduction to Probability & Statistics – 4 th Edition J.Susan Milton, Jesse C. Arnold Tata McGraw Hill.
3. Fundamentals of Statistics (7th edition), S C Gupta, Himalaya Publishing house
4. Probability and Statistical Inference (9th Edition), Robert V. Hogg & Elliot Tanis & Dale Zimmerman, Pearson.

Pokhara University
Faculty of Science and Technology

Course Code.:	Full marks: 100
Course title: **Web Technology II**	Pass marks: 45
Nature of the course: Theory/Practical/Theory & Practice	Time per period: 1 hour
Year, Semester: Fourth	Total periods: 45
Level: Bachelor	Program: BCA

1. Course Description

This course is designed to give students hands on knowledge of server side development. After completing this course students will not only have theoretical knowledge but also practical, hands-on experience in building, securing, and deploying web applications, making them well-prepared for the challenges of professional web development.

The course covers essential topics such as server architecture, web frameworks, database integration, web services, security best practices, and deployment strategies. Students will gain hands-on experience in building, securing, and deploying web applications.

2. General Objectives

- To provide knowledge about the fundamentals of server-side development.
- To provide the strong foundation of the serverside programming language.
- To enable students to develop web applications using the framework.
- To enhance students' ability to integrate and manage databases in web applications.
- To introduce students to the basics of Content Management Systems (CMS).
- To develop skills in creating and consuming web services and APIs.
- To equip students with the knowledge to implement security best practices in web applications.
- To prepare students for the deployment of web applications to production environments.

3. Contents in Detail

Specific Objectives	**Contents**
– To understand how web technologies work and the roles of client-side vs. server-side. – To learn about HTTP request and response message format and different web servers. – To explore server architectures and backend programming languages. – To understand MVC design in web development.	**Unit I: Introduction to Server-Side Development (4 hours)** 1.1. Overview of Web -I 1.2. HTTP request message, HTTP response Message 1.3. Differences between client-side and server-side technologies 1.4. Web Clients 1.5. Web Server 1.5.1. Role of the server in a web application 1.5.2. Overview of web servers - Apache

	- Nginx - IIS 1.6. Server-side architecture 1.6.1. Monolithic 1.6.2. Microservices 1.7. Introduction to backend programming languages (Python, PHP, Java) 1.8. Understanding MVC (Model-View-Controller) architecture
– To set up Python and learn its basics. – To gain knowledge about Python data types, functions, and control structures. – To gain knowledge about lists, dictionaries, files, and understand OOP concepts. – To understend working with CSV and JSON files.	**Unit-II: Server-Side Programming with Python (8 hours)** 2.1 Setting up the Python environment 2.2 Introduction to Python syntax and basic operations 2.3 Data Types and Variables 2.3.1. int, float, str, bool, list, dict, tuple, set 2.3.2. Variable assignment and naming conventions 2.3.3. Type conversion 2.4 Control Structures 2.4.1 Conditional statements: if, elif, else 2.4.2 Loops: for and while 2.4.3 List comprehensions 2.5 Introduction to functions 2.5.1 Defining functions, parameters, return values 2.5.2 Scope and global variables 2.6 Working with Python Data Structures (1 hour) 2.6.1 Lists 2.6.2 Creating, accessing, and modifying lists List methods: append(), remove(), pop(), etc. 2.6.3 Dictionaries: Key-value pairs, adding and accessing items Dictionary methods: get(), keys(), values() 2.6.4 Tuples and Sets: Creating and using tuples and sets 2.6.5 Differences between lists, tuples, and sets 2.7 File Handling in Python 2.7.1 Introduction to File Operations 2.7.2 Opening and closing files 2.7.3 File modes 2.7.4 Reading from Files Reading entire files, line-by-line reading 2.7.5 Using with statement for file operations 2.7.6 Writing to Files: Writing data to files, Appending data to existing files

	2.8 Working with CSV and JSON Files 2.8.1 Reading and writing CSV files using csv module 2.8.2 Handling JSON files using json module 2.9 OOP Concepts
– To understand the importance of framework in web development – To get started with Flask and understand its core features. – To manage HTTP requests, form submissions, cookies, and sessions. – To explore additional Flask features like file uploads and sending emails.	**Unit III: Working with Web Framework (6 hours)** 3.1 Framework overview: 3.2 Introduction to Flask Framework 3.3 Setting up a Flask environment 3.4 Flask basics: Routes, Views, and Templates 3.5 Handling HTTP requests and responses 3.6 Building Basic Web Functionality 3.6.1 Implementing form submission and input validation 3.6.2 Understanding and managing cookies and sessions 3.7 Flask redirect, Message flashing , File upload 3.8 Sending Email
– To connect web application with database – To Perfoem basic CRUD operations and manage database transections.	**Unit IV: Database Integration (7 hours)** 4.1 Introduction to Databases 4.2 Overview of relational (SQL) vs. NoSQL databases 4.3 Database Connectivity with Flask 4.3.1 Connecting Flask applications to databases using SQLAlchemy 4.4 Performing CRUD operations using HTML template 4.5 Handling database connections and transactions
- To peovide the knowledge about CMS and its importance in web. - Provide the knowledge of the popular frameworks.	**Unit V: Introduction to Content Management Systems (CMS) (3 hours)** 5.1 Overview of popular CMS platforms (WordPress, Drupal, Joomla) 5.2 Advantages of using CMS in web development 5.3 Use cases and scenarios where CMS is beneficial 5.4 Brief about Web-programming frameworks: php, java, python frameworks.
- To Understands concepts like SOA,SOAP, API , JSON and XML . - To be able to write REST API.	**Unit VI: Web Services and APIs (5 hours)** 6.1 Introduction to Web Services 6.2 Brief about service-oriented architecture 6.3 SOAP 6.4 RESTful web services 6.6 RESTful APIs vs. SOAP

	6.6 JSON and XML data formats 6.7 Building and Consuming APIs 6.7.1 Creating simple RESTful APIs using Flask
– To know web security basics and common vulnerabilities. – To learn security principles and best practices. – To implement protection measures for your web apps.	**Unit VII: Security in web applications (8 hours)** 7.1 Web application security fundamentals 7.1.1 Foundations of security 7.1.2 Vulnerabilities 7.1.3 Threats 7.1.4 Attacks, 7.2 Security Principles: 7.2.1 Least Privilege 7.2.2 Defense in Depth 7.2.3 Fail Securely 7.3 Common web vulnerabilities 7.3.1 SQL Injection, 7.3.2 XSS 7.3.3 CSRF 7.3.4 DoS 7.3.5 Live examples and demonstrations of vulnerabilities 7.4 Security best practices for web development 7.4.1 Input Validation and Output Encoding 7.4.2 Secure Authentication and Authorization Multi-Factor Authentication (MFA),Role-Based Access Control (RBAC), Secure Session Handling, Session Fixation and Hijacking Prevention, Error Handling and Logging, Secure File Uploads 7.4.3 SSL/TLS and HTTPS 7.5 Firewalls, proxies, and VPNs 7.6 Host and Network Security threats and countermeasures 7.7 Application Security threats and countermeasures 7.8 Design guidelines for secure web applications
– To provide knowledge of deployment and deployment platforms – To teach students to applu SSL certificates to web app or website	**Unit VIII: Web Application Deployment (4 hours)** 8.1 Introduction to Deployment 8.2 Deployment platforms 8.3 Basic differences between development and production environments. 8.4 Setting up environment variables and basic production settings.

	8.5 Introduction to using Gunicorn to serve Flask apps. 8.6 Setting up basic HTTPS using a free certificate (e.g., Let's Encrypt). 8.7 Brief introduction to monitoring and logging. 8.8 Importance of security in deployment.

4. Methods of Instruction

The course uses lectures, discussions, group work, lab sessions to blend theoretical knowledge with practical application, ensuring students gain both understanding and hands-on experience in web development. Assignments will be carefully designed by the teacher focusing real-world projects to reinforce learning, with clear instructions and deadlines.

5. List of Practical

?

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	5
		Lab tasks	10
		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.

7. Prescribed Books and References

Text Book

Reference Books

Grinberg, M. (2018). Flask Web Development: Developing Web Applications with Python (2nd ed.). O'Reilly Media.

Sweigart, A. (2020). The Big Book of Small Python Projects: 81 Easy Practice Programs. No Starch Press.

Beazley, D. (2020). *Python distilled*. Addison-Wesley.

Pokhara University
Faculty of Science and Technology

Course Code.: PRJ 251	Full marks: 100
Course title: **Project II**	Pass marks: 45
Nature of the course: Theory/Practical/Theory & Practice	Time per period: 1 hour
Year, Semester: Fourth	Total periods: 45
Level: Bachelor	Program: BCA

1. Course Description

Project-II is an immersive, hands-on course designed for BCA fourth-semester students to translate their classroom knowledge into a tangible, real-world application. This course offers students the opportunity to take the theories and skills they've been learning and apply them to a meaningful project, whether it's a web or desktop application. Working independently or in small teams, students will develop a software solution that not only addresses a real-world problem but also showcases their ability to manage and execute a project from start to finish. Guided by a faculty supervisor, students will navigate through the entire software development lifecycle, gaining valuable experience and insights along the way. This course is valued at 2 credits.

2. General Objectives

- To empower students to apply the knowledge and skills gained throughout the BCA program to create a practical software application.
- To encourage students to think critically and solve problems creatively as they work on their projects.
- To help students develop essential project management, teamwork, and communication skills that are crucial for their future careers.
- To enable students to produce a functional, user-friendly, and well-documented software application that meets specific needs.
- To provide students with hands-on experience in every phase of project development, from planning and design to implementation and testing.

3. Project Nature

The project should focus on the development of a web or desktop application. Students are encouraged to choose a project that they are passionate about and that allows them to demonstrate their skills in areas such as user interface design, backend development, and database management. Whether building a web platform or a desktop tool, the project should be original, substantial, and reflective of the students' capabilities in software development and project management.

4. Project Milestone

The schedule can be given to the students according to the following phases.

I. **Team formation (Week 1)**

- Students will form a team consisting of a maximum of 3 members based on their interests and skills.
- Teams should be balanced to ensure diverse skill sets and effective collaboration.

II. **Project Proposal Submission**: **(Week 2-4)**

- Students will start by defining the project's scope, objectives, and deliverables.
- A detailed project proposal must be submitted, outlining the project idea, timeline, and required resources.

Proposal Format:

1. Introduction
2. Aim and Scope
3. Objectives
4. Current Problem and Proposed Solution
5. High level requirements
   - 5.1 Functional
   - 5.2 Non functional
   - 5.3 Requirement prioritization table
6. Design
   - 6.1 UML Diagram/ERD, DFD

7. Development cost estimation
8. Detail Timeline
9. Conclusion
10. References

III. **UI/UX Design (Week 5-6)**

- Students should begin by gathering user stories to understand the needs, goals, and pain points of potential users. These user stories will guide the design and development of the user interface and user experience, ensuring the final product meets the end users' expectations.
- User Interface designed with any UI/UX Design tool.
- Present and defend the UI/UX design.

IV. **First Milestone (Week 7-8)**

- Students should present their project progress. At this stage 60% of the work should be completed.

V. **Second Milestone (Week 9)**

- Students should present their project progress. At this stage 100% of the work should be completed.

VI. Final Defense (Week 10)

- Final Project demonstration/presentation with documentation.
- The final defense includes a demonstration of the project, focusing on both functionality and documentation quality.

7 Guidelines for Project using Framework

- It is suggested to develop an application using emerging frameworks.
- It is recommended that the team should be of 2-3 students. Teams cannot exceed 3 members to ensure effective collaboration and manageability.
- The project should be free from plagiarism of any kind. Students are expected to produce their own work and properly attribute any external resources or inspiration.
- Coding standards should be followed meticulously. At the minimum, the code should be self documented, modular, and should use the meaningful naming convention.
- It is advisable that object-oriented methodology is used with reusability of classes and code, etc.
- Mentor/ Internal guides (i.e. the faculty members) must devote time, allocated as per the time table to guide the students for the project. The time allocation will be in accordance with the teaching scheme for 4th semester project.

8 Report format:

Title Page
Original Copy of the Approval
Certificate of Authenticated work
Role and Responsibility Form
Abstract
Acknowledgement
Table of Contents
Table of Figures
List of Abbreviations
List of Tables

CHAPTER 1: INTRODUCTION

1.1 Background
1.2 Objectives
1.3 Purpose, Scope, and Applicability
    1.3.1 Purpose
    1.3.2 Scope and Limitation
    1.3.3 Applicability
1.4 Achievements
1.5 Organization of Report

CHAPTER 2: SURVEY OF TECHNOLOGIES

Review of the similar/relevant projects

CHAPTER 3: REQUIREMENTS AND ANALYSIS

3.1 Problem Definition
3.2 Requirements Specification
3.3 Planning and Scheduling
3.4 Software and Hardware Requirements
3.5 Preliminary Product Description
3.6 Conceptual Models

CHAPTER 4: DESIGN

4.1 Introduction
4.2 System Design
4.3 Database design
4.4 Interface Design
4.5 Summary

CHAPTER 5: IMPLEMENTATION AND TESTING

5.1 Implementation Approaches
5.2 Coding Details and Code Efficiency
    5.2.1 Code Efficiency
5.3 Testing Approach
    5.3.1 Unit Testing
    5.3.2 Integrated Testing
    5.3.3 Beta Testing
5.4 Modifications and Improvements
5.5 Test Cases

CHAPTER 6: RESULTS AND DISCUSSION

6.1 Test Reports
6.2 User Documentation

CHAPTER 7: CONCLUSIONS

7.1 Conclusion
    7.1.1 Significance of the System
7.3 Recommendation

REFERENCES

IEEE refrecence styple

Note: 3 copies of the report should be submitted to the department of computer application with proper binding. The report should be approved and signed by an external, internal examiner and head of department.

9 Evaluation system and Students' Responsibilities

Marking Rubric:

S.N.	**Criteria**		**Marks**
1	Proposal defense		20%
2	Main Objective of Application should be properly functional and dynamic.		30%
3	Login/Register and other user management function should be fully functional.		5%
4	Other features of application should be fully functional and dynamic.		15%
5	User Interface of application should be up to standard.		10%
6	Presentation and Documentation		10%
7	Usability of application in market and teamwork evaluation		10%
	Total		**100%**

***Note :** Criteria No. 1, 2 and 3 are mandatory and incase on incompletion of any one of them will result in **incomplete** project marking.

Evaluation committee:

- Program Coordinator
- Project Supervisor
- Internal Examiner
- External Examiner (Should be assigned by university)

Students' Responsibilities

Students are responsible for actively participating in all project phases, collaborating effectively within their teams, and adhering to deadlines. They must ensure the quality and originality of their work, maintain thorough documentation, and regularly communicate with mentors. Ethical conduct, problem-solving, and innovation are key, alongside continuous learning and effective presentation skills. These responsibilities aim to equip students with practical experience and essential skills for their future careers.

11. Document Formatting Guidelines

Page Numbering:

Use Roman numerals (i, ii, iii, etc.) for pages from the certificate page to the list of tables, figures, abbreviations, and approvals.

Main Content: Use Arabic numerals (1, 2, 3, etc.) starting from Chapter 1.

Placement: Page numbers should be centered at the bottom of the page.

Page Size and Margins:

Page Size: A4

Margins:

- **Top**: 1 inch (2.54 cm)
- **Bottom**: 1 inch (2.54 cm)
- **Left**: 1.25 inches (3.17 cm)
- **Right**: 1 inch (2.54 cm)

Paragraph Style:

Alignment: Justified

Spacing: 1.5 line spacing

Text Font:

Font Type: Times New Roman

Font Size: 12 for body text

Section Headings:

Chapter Headings: Font size 16, bold

Section Headings: Font size 14, bold

Sub-section Headings: Font size 12, bold

Figures and Tables:

Alignment: Centered on the page

Captions:

- **Figure Captions**: Bold, font size 12, centered below the figure
- **Table Captions**: Bold, font size 12, centered above the table

Pokhara University
Faculty of Science and Technology

Course Code.: MTH230 (3 Credit)		Full Marks: 100
Course Title: Numerical Methods (3-0-1)		Pass Marks: 45
Nature of the Course: Theory/Practical		Total Lectures: 48 hours
Level: Bachelor	Year: III / Semester: V	Program: Bachelor of Computer Application

1. Course Description:

This course covers how a computer can be used to solve the problems that may not be solvable by calculus and algebra. It develops the problem-solving techniques required to study other courses and mathematical relationship that can be used to simulate real –world situation.

2. General Objectives:

The general objectives of this course are:

1. To solve elementary matrix, arithmetic problems analytically and numerically.
2. To introduce numerical methods for interpolation, regressions and finding of roots.
3. To find the solution of ordinary and partial differential equations.
4. To provide knowledge of relevant high level programming language for computing, implementing, solving and testing of algorithms.

3. Methods of Instructions:

Lecture
Tutorial/Discussion/Readings
Practical works and Project works

4. Course Contents:

Specific Objectives	**Contents**
Unit 1: Solution of Non-linear equation	**10 hours**
• Solve none linear equations by different methods of various fields and comparison of their convergence.	1.1. Introduction, Importance of Numerical Methods 1.2. Approximation and Errors in computation. 1.3. Bisection Method. 1.4. Secant method. 1.5. Newton Raphson method (for one variable) 1.6. Fixed point iterative method. 1.7. Rate of convergence and comparisons of these Methods.
Unit 2: Interpolation and Approximation	**8 hours**
• Develop mathematical relationship between observations to simulate real world situation.	2.1. Lagrange interpolation 2.2. Finite differences (forward, backward, and divided difference) 2.3. Newton's Interpolation (forward, backward). 2.4. Least square method of fitting linear and nonlinear curve for discrete data and continuous function. 2.5. Spline: Interpolation (Cubic Spline)
Unit 3: Numerical Differentiation and Integration	**8 hours**
• Solve the not solvable problem of calculus.	3.1. Numerical Differentiation formulae 3.2. Maxima and minima

	3.3. Newton-Cote general quadrature formula 3.4. Trapezoidal, Simpson’s 1/3, 3/8 rule 3.5. Romberg integration 3.6. Gaussian integration (2-point and 3-point formula)
Unit 4: Solution of system of linear algebraic equations	**10 hours**
• Solve linear equations by different techniques.	4.1. Gauss elimination method. 4.2. Gauss-Jordan method 4.3. The inverse of a matrix. 4.4. Ill-conditioned system of linear equations. 4.5. LU Factorization method (Dolittle’s, Crout’s, Cholesky’s) 4.6. Iterative methods (Jacobi method, Gauss-Seidel method) 4.7. Eigen value and Eigen vector using Power method
Unit 5: Solution of ordinary differential equations	**6 hours**
• Solve the ordinary differential equations which may exist in real world but not solvable by calculus.	5.1. Overview of solution of differential equations 5.2. Taylor’s series method. (for first and second order) 5.3. Euler’s method (for first and second order) 5.4. Heun’s method. (for first and second order) 5.5. Runge-Kutta methods (for first and second order) 5.6. Solution of boundary value problem by finite difference method and shooting method.
Unit 6: Numerical solution of Partial differential Equation	**6 hours**
• Solve the partial differential equations which exist in real world but not solvable by calculus.	6.1. Classification of partial differential equation (Elliptic, parabolic, and Hyperbolic) 6.2. Solution of Laplace equation (standard five-point formula with iterative method) 6.3. Solution of Poisson equation (finite difference approximation) 6.4. Solution of one-dimensional Heat equation by Schmidt method.

5. Practical

The following practical will be conducted by using MATLAB/C/C++ or any other relevant high level programming languages.

List of Practical

SN	List of Practical
1.	Solution of nonlinear equations.
2.	Interpolation and regression.
3.	Differentiation and Integration.
4.	Linear system of equations and power method.
5.	Ordinary differential equations.

6. Evaluation System and Students’ Responsibilities:
6.1 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.

Internal Evaluation	**Weight**	**Marks**	**External Evaluation**	**Marks**
Theory		**30**	**Semester End**	**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**		**50**
Full Marks: 50 + 50 = 100				

6.2 Students' Responsibilities:

To be eligible for the Semester End Examinations, 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.

7. Prescribed Books and References:

Text Books:

1. C.F. Gerald and P.O. Wheatley ***Applied Numerical Analysis (7th edition),*** New York.
2. Dr. B.S. Grewal, ***Numerical Methods in Engineering and Science,*** Khanna Publication, (10th edition)

References:

1. E. Balagurusamy ***Numerical Methods,*** New Delhi; Tata McGraw Hill, 2010
2. Richard L. Burden, J. Douglas Faires, "Numerical Analysis 7th edition" , Thomson / Brooks/Cole
3. S.S. Sastry ***Introductory Methods of Numerical Analysis (4th edition),*** Prentice-Hall of India, New Delhi, 2008.
4. Dr. V. N. Vedamurthy & Dr. N. Ch. S. N. Iyengar ***Numerical Methods,*** Noida, Vikash Publication House 2009.
5. Rudra Pratap ***Getting Started with MATLAB,*** Oxford University Press 2010

Pokhara University
Faculty of Science and Technology

Course Code.: CMP 317 (3 Credits)		Full Marks: 100
Course Title: Dot Net Technology (3-0-3)		Pass Marks: 45
Nature of the Course: Theory/Practical		Total Lectures: 48 hours
Level: Bachelor	Year: III / Semester: V	Program: Bachelor of Computer Application

1. Course Description:

This course provides a comprehensive introduction to the .NET ecosystem, focusing on essential programming and application development skills. Its major intent is to equip students with a solid understanding of .NET Core and C# programming fundamentals, enabling them to build, manage, and deploy modern .NET applications. The course covers key areas including .NET Core architecture, C# language basics, console applications, dependency management with NuGet, and data access using Entity Framework Core. It includes hands-on experiences with ASP.NET Core for web API development, control flow in C#, exception handling, and LINQ for data manipulation. Delivery approaches include a mix of self-learning modules, interactive and experiential exercises to enhance understanding and practical application. Students will engage in project-based assignments and use Visual Studio IDE to apply their knowledge, ensuring a practical and immersive learning experience.

2. General Objectives:

The general objectives of this course are:

- To familiarize students with the evolution, core principles, and differences between .NET Framework and .NET Core.
- To enable students to understand .NET Core's cross-platform, Common Language Runtime (CLR) and core .NET Core libraries.
- To guide students in using Visual Studio IDE for project creation, management, and version control.
- To equip students with skills in implementing control flow statements, object-oriented, exception handling, handling input/output and command-line arguments.
- To familiarize students with dependency management using NuGet, including adding packages and managing versions.
- To provide skills in modeling data, defining entities, and performing CRUD operations using Entity Framework Core.
- To introduce students to building dynamic Web APIs, including RESTful design principles, HTTP methods, and versioning strategies.

3. Methods of Instructions:

1. Lectures and discussion,
2. Practical/Group Work,
3. Experiential Exercises,
4. Self-Learning Modules.

4. Course Contents:

Specific Objectives	**Contents**
Unit 1: Introduction to .NET	**4 hours**
• Describing the evolution and core principles, key features, cross-platform capabilities and application structure of .NET Core. • Explore the role of the Common Language Runtime (CLR) in .NET execution,	1.1. Overview of .NET and its evolution 1.1.1. Core principles of the .NET Framework 1.1.2. Relationship between .NET Framework and .NET Core 1.1.3. Key features and benefits of the .NET platform 1.1.4. Role of .NET in modern application development

including memory management and JIT compilation. • Get acquainted with Visual Studio IDE, including project creation, management.	1.1.5. Brief history of .NET 1.2. Introduction to .NET Core 1.2.1 .NET Core 1.2.2 Key differences between .NET Framework and .NET Core 1.2.3 Cross-platform capabilities of .NET Core 1.2.4 .NET Core application structure 1.3. Common Language Runtime (CLR) 1.3.1 Role of CLR in .NET Execution 1.3.2 Memory management and garbage collection 1.3.3 JIT compilation, CTS and its importance 1.4. .NET Core libraries 1.4.1 Core libraries included in .NET Core 1.4.2 Namespace structure 1.4.3 Fundamental classes and their functionalities
Unit 2: C# Programming Fundamentals	**8 hours**
• Explain structure of a C# program and perform I/O operations with comments and documentation. • Implement control flow using selection, iteration, and jump statements. • Understand object-oriented programming (OOP) concepts, including classes, objects, inheritance, polymorphism, encapsulation, and abstraction. • Work with methods, properties, and indexers in C#. • Handle exceptions using try-catch-finally blocks, throw and create custom exceptions, and understand exception hierarchies.	2.1. C# language basics 2.1.1. C# Language History & Evolution 2.1.2. C# Syntax and Keywords 2.1.3. Basic Program Structure 2.1.4. I/O Operations, Comments and Documentation 2.2. Data types, variables, operators 2.2.1. Value Types & Reference Types 2.2.2. Variable Declaration and Initialization 2.2.3. Operators & its types 2.2.4. Type Casting and Conversation 2.3. Control flow statements 2.3.1. Selection, Iteration and Jump Statements 2.4. Object-oriented programming concepts (OOPs): 2.4.1. Introduction to OOPs, Classes and Objects 2.4.2. Methods, Properties & Indexers 2.5. Exception handling 2.5.1. Introduction to exception 2.5.2. Try-catch-finally blocks 2.5.3. Throwing exceptions and custom exceptions 2.5.4. Exception Hierarchies
Unit 3: Basic .NET Core Applications	**8 hours**
• Perform input/output operations in a console application. • Implement command-line arguments in a .NET Core console application. • Implement error handling and validate user input in console applications. • Understand the structure and role of the .NET Core project file (csproj). • Build, run, debug, and publish .NET Core applications for different platforms.	3.1. Console applications: Input/output, command line arguments 3.1.1. Creating a console application project 3.1.2. Basic console application structure 3.1.3. I/O operating in Console 3.1.4. Using Command-line arguments 3.1.5. Error handling and user input validation 3.2. Building and running .NET Core application 3.2.1. Understanding the .NET Core project file(csproj) 3.2.2. Building the application for different target platforms 3.2.3. Running the application from command line 3.2.4. Debugging and Publishing .NET Core application

Unit 4: Advanced .NET Core Concepts	**4 hours**
• Describe the role of NuGet packages in a project. • Explain the concept of assemblies and their role in .NET. • Implement best practices for exception handling in .NET Core. • Develop and use custom exception types in applications.	4.1. Dependency management with NuGet 4.1.1. Introduction to NuGet packages 4.1.2. Adding NuGet packages to a project 4.1.3. Managing package version 4.2. Class libraries and namespaces 4.2.1. Creating class library projects 4.2.2. Organizing code using namespaces 4.2.3. Assembly concept and its role in .NET 4.2.4. Referencing class libraries in other projects 4.3. Implementing Exception handling 4.3.1. Best practices for exception handling 4.3.2. Custom Exception Types
Unit 5: Data Access with Entity Framework Core	**8 hours**
• Explore the benefits of using EF Core in .NET Core applications. • Compare code-first and database-first development approaches. • Implement CRUD operations (Create, Read, Update, Delete) using EF Core. • Develop databases and manage data using EF Core. • Master LINQ syntax for data manipulation. • Implement complex queries, including joins, projections, filtering, and grouping using LINQ to Entities.	5.1. Introduction to Entity Framework Core (EF Core) 5.1.1. Overview of ORM 5.1.2. Basic concept and architecture of EF Core 5.1.3. Benefits of using EF Core 5.1.4. Relationship with .NET Core 5.2. Data modeling concepts: Entities, relationships 5.2.1. Entity Framework Core Data Model 5.2.2. Defining entities and properties 5.2.3. Primary Keys and Foreign Keys 5.2.4. Relationships: 5.2.5. Navigation Properties 5.3. Working with databases: CRUD operations 5.3.1. Connecting to a database using EF Core 5.3.2. Creating new records 5.3.3. Retrieving data using LINQ 5.3.4. Updating and Deleting records 5.4. LINQ queries for data manipulation 5.4.1. Introduction to Language Integrated Query 5.4.2. Basic LINQ Syntax 5.4.3. Quering data using LINQ to Entities 5.4.3. Projecting, filtering, ordering, grouping 5.4.4. Joining data from multiple tables
Unit 6: ASP .NET Core Web Development	**8 hours**
• Describe the ASP.NET Core MVC framework. • Explore Razor syntax and page structure. • Implement model binding and data access in Razor Pages. • Implement Create and manage controllers to handle user requests. • Develop views using Razor templates and data binding techniques. • Create and validate HTML forms in ASP.NET Core. • Apply Dependency Injection (DI) principles for loose coupling.	6.1. Introduction to ASP.NET Core MVC 6.1.1. ASP.NET Core MVC framework 6.1.2. MVC architecture 6.2. Razor Pages for simple web applications 6.2.1. Razor syntax and page structure 6.2.2. Model binding and data access in Razor 6.2.3. Creating dynamic and interactive pages 6.3. Routing and controllers 6.3.1. Routing mechanisms in ASP.NET core MVC 6.3.2. Routes and URL patterns 6.3.3. Creating controllers to handle user requests 6.3.4. Controller actions and action methods 6.3.5. Returning different types of responses

	6.4. Views and data binding 6.4.1. Roles of views in presenting data to the user 6.4.2. Using Razor templates to build views 6.4.3. Data binding as controllers to views 6.4.4. Model formatting and dynamic content 6.5. Forms and validation 6.5.1. Creating HTML forms for user input 6.5.2. Handling form submissions in controllers 6.5.3. Implementing data validation techniques 6.5.4. Error handling and validation messages 6.6. Overview of Dependency Injection (DI) concepts 6.6.1. Dependency Injection (DI) principles 6.6.2. Benefits of using DI for loose coupling 6.6.3. Built-in DI container in ASP.NET Core 6.6.4. Registering and injecting dependencies 6.6.5. Implementing custom services, repositories
Unit 7: Building Modern Web APIs	**8 hours**
• Implement Web APIs APIs. • Explain best practices for clean and maintainable API design. • Explor RESTful API principles and design. • Implement HTTP methods (GET, POST, PUT, DELETE) in RESTful APIs. • Develop APIs using the ASP.NET Core framework.	7.1. Introduction to Web APIs 7.1.1. Web APIs and its purposes 7.1.2. Advantages of using Web APIs 7.1.3. Common use cases for Web APIs 7.1.4. Design considerations for APIs 7.2. Designing RESTful APIs 7.2.1. REST principles 7.2.2. Resources identifiers in RESTful APIs 7.2.3. HTTP methods *(GET, POST, PUT, DELETE)* 7.2.4. Designing URLs for API endpoints 7.2.5. Versioning strategies for Web APIs 7.3. ASP.NET Core Web API framework 7.3.1. Introduction to ASP.NET Core Web API 7.3.2. Building web API project in ASP.NET Core 7.3.3. Action methods for API requests. 7.3.4. Returning data from controllers JSON, XML 7.3.5. Handling input parameters and data binding 7.4. Data serialization (JSON, XML) 7.4.1. Importance of data serialization in Web APIs 7.4.2. JSON structure and serialization techniques 7.4.3. Introduction to XML 7.4.4. XML structure and serialization techniques
5. Practical Works:	
1. Develop console application using C# with .NET Core 2. Create .NET Core class libraries 3. Build database-driven applications using Entity Framework Core and LINQ 4. Develop web applications with ASP.NET Core MVC or Razor Pages 5. Create Simple Web APIs using ASP.NET Core	
6. Evaluation System and Students' Responsibilities:	
6.1 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.

Internal Evaluation	**Weight**	**Marks**	**External Evaluation**	**Marks**
Theory		**30**	**Semester End examination**	**50**
Attendance / Class Participation	10%			
Assignments	20%			
Project Work/Presentations	10%			
Term Exam	60%			
Practical		**20**		
Attendance and Lab Participation	10%			
Lab Report	20%			
Lab Examination	40%			
Viva Examination	30%			
Total Internal Marks		**50**		
Full marks=50+50				

6.2 Students' Responsibilities:

To be eligible for the Semester End Examinations, 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.

7. Prescribed Books and References:

1. Mueller, J. (2021). *C# for Dummies*. Wiley.
2. MacDonald, M. (2018). *Beginning ASP.NET in C#*. Wrox.
3. Nagel, C., & Glynn, J. (2021). *Professional C# and .NET Core*. Wrox.

Pokhara University **Faculty of Science and Technology**	
Course Code.: CMP 336 (3 Credits)	Full Marks: 100
Course Title: Data Communication and Computer Networks (3-0-2)	Pass Marks: 45
Nature of the Course: Theory/Practical	Total Lectures: 48 hours
Level: Bachelor | Year: III / Semester: V	Program: Bachelor of Computer Application

1. Course Description:

This course is designed to provide students with the fundamental concepts of data communication and the architecture, protocols, and technologies that underpin modern computer networks. It covers the principles of digital communication, network models, transmission media, and network devices, focusing on the OSI and TCP/IP models. Students will explore key topics such as LANs, WANs, IP addressing, routing, switching, and network security. Through practical exercises and simulations, the course aims to equip students with the skills to design, implement, and manage robust and efficient computer networks, preparing them for careers in network administration and IT infrastructure management.

2. General Objectives:

The general objectives of this course are as follows:

- To equip students with in-depth insights into the basic principles of data communication and networking, including architecture, protocols, and services.
- To build a strong grasp of essential concepts focusing on the OSI and TCP/IP models.
- To develop hands-on abilities in configuring and managing network and intermediate devices such as routers and switches.
- To equip students with knowledge of IP addressing schemes, subnetting, and the implementation of routing protocols

3. Methods of Instructions:

- Lecture, Discussion, Readings, Practical works, Case Studies and Project works

4. Course Contents:

Specific Objectives	**Contents**
Unit 1: Data Communications	**8 hours**
• Familiarize students with the basic concept of Data Communications, data transmission techniques and impairments, with performance parameters and related theorems.	1.1 Definition, Components of Data Communication 1.2 Data Transmission Modes: Simplex, Half-Duplex, Full-Duplex 1.3 Data Representation: Analog and Digital 1.4 Analog and Digital Transmission 1.5 Serial and Parallel Communication 1.6 Asynchronous and Synchronous Transmission 1.7 Transmission Impairments 1.8 Bandwidth, Throughput, Baud Rate, Jitter 1.9 Nyquist and Shannon Capacity Theorem 1.10 Digital Transmission Techniques

Unit 2: Multiplexing and Switching	**4 Hours**
- Explore the various types of multiplexing and switching techniques, and their usage in data communication.	2.1 Multiplexing and its types- FDM, TDM, WDM 2.2 Switching and its types- Circuit, Packet and Message switching
Unit 3: Reference Model	**6 Hours**
- Elaborate the concept of Protocol, Protocol architecture along with OSI, TCP/IP and Internet Protocol Stack.	3.1 Protocols and Layered Architecture 3.2 OSI Reference Model and functions 3.3 TCP/IP Protocol 3.4 Cross layers 3.5 Devices in different layers 3.6 Software defined networks
Unit 4: Physical Layer Design Issues	**4 Hours**
- Analyze different transmission media, including both guided and unguided methods, and explore the standards for twisted pair cabling, labeling, and testing.	4.1 Network Topologies 4.2 Transmission Medium 4.2.1 Guided Transmission Media (Twisted Pair, Coaxial, Optical Fiber) 4.2.2 Unguided Transmission Media (Bluetooth, Wi-Fi, Cellular, Radiowave, Microwave, Satellite) 4.3 Twisted Pair Cabling, labelling and Testing Standards
Unit 5: Data Link Layer Design Issues	**6 Hours**
- Explore basics of error detection and correction techniques with data link layer protocols.	5.1 Framing 5.2 Error Detection (Parity, Checksum and CRC) 5.3 Error Correction (FEC, ARQ) 5.4 Flow Control Protocols (Stop and Wait, Go-Back-N, Selective Repeat) 5.5 ATM, HDLC, PPP
Unit 6: Network Layer Design Issues	**8 Hours**
- Gain understanding of network layer design, focusing on logical addressing, subnetting, routing protocols, and the transition from IPv4 to IPv6, including migration strategies and the role of Network Address Translation (NAT) in modern networks.	6.1 Logical Addressing (IPv4/IPv6) 6.2 Subnetting and Prefix Delegation 6.3 Routing Protocols (Static, RIP, EIGRP and OSPF) 6.4 IPv4 Frame Format 6.5 IPv6 Frame Format 6.6 IPv6 Migration Strategies 6.7 Network Address Translation
Unit 7: Transport and Application Layer	**6 Hours**
- Explore the transport and application layer Protocols and processes.	7.1 Process to Process Communication 7.2 TCP, UDP and Sockets 7.3 Three Way Handshake 7.4 Client Server Computing Model 7.5 Application Layer Protocols (DNS, DHCP, HTTP, SMTP, POP, IMAP) 7.6 Congestion Control Approaches
Unit 8: Network Management and Security	**6 Hours**
- Delve into cryptographic techniques,	8.1 Network Management

the functionality of VPNs and firewalls necessary for network management and security.	8.2 Infrastructure of Network Management, FCAPS 8.3 Cryptography (Symmetric and Asymmetric) 8.4 Virtual Private Networks 8.5 Firewalls 8.6 Layered Approach of Network Security

5. List of Practical:

SN	Practical Works
1.	Network Cabling (Crossover and Straight Through)
2.	Configure IP addresses and Subnetworks
3.	Basic Network Commands (ipconfig, ping, netstat, arp, tracert, telnet, nslookup)
4.	Network Design and Visualization in Simulation Software
5.	Enterprise LAN Architecture, DNS and VLAN
6.	Deployment of Routing Protocols (Static, RIP, EIGRP and OSPF), NAT
7.	Deployment of Virtual Private Networks (VPN)
8.	Analyzing Network Traffic Using Wireshark

6. 6. Evaluation System and Students' Responsibilities:

6.1 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.

Internal Evaluation	Weight	Marks	External Evaluation	Marks
Theory		**30**	**Semester End examination**	**50**
Attendance / Class Participation	10%			
Assignments	20%			
Project Work/Presentations	10%			
Term Exam	60%			
Practical		**20**		
Attendance and Lab Participation	10%			
Lab Report	20%			
Lab Examination	40%			
Viva Examination	30%			
Total Internal Marks		**50**		
Full marks=50+50 =100				

6.2 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 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.

7. Prescribed Books and References:

Text Books:

1. B.A. Forouzan, *Data Communications and Networking*, 5th edition. McGraw-Hill Education, 2012.

References:

1. J. F. Kurose and K. W. Ross, *Computer Networking: A Top-Down Approach*, 7th ed. Pearson, 2017.
2. A.S. Tanenbaum and D. J. Wetherall, *Computer Networks*, 5th ed. Pearson, 2010.
3. W. R. Stevens, *TCP/IP Illustrated, Volume 1: The Protocols*, 1st ed. Addison-Wesley Professional, 1994.
4. T. Lammle, *CCNA Routing and Switching Study Guide*, 2nd ed. Sybex, 2016.
5. Lowe, *Networking All-in-One For Dummies*, 8th ed. Wiley, 2018.

Pokhara University
Faculty of Science and Technology

Course Code: ELE322 (3 Credits)	Full Marks: 100
Course Title: Research Methodology (3-0-1)	Pass Marks: 45
Nature of the Course: Theory/Practical	Total Lectures: 48 hours
Level: Bachelor | Year: III / Semester: V	Program: Bachelor of Computer Application

1. Course Description:

The course is a basic one and it should be treated at an introductory level of research methodology and method of doing practical knowledge. It introduces students to the fundamentals of research methodology and helps students to build a strong foundation in research methodology, enabling them to conduct independent research and contribute to their field of study by considering research ethics

2. General Objectives:

The general objectives of this course are:

- To provide the concept the basic concepts and principles of research, including the scientific method and ethical considerations
- To familiarize the students with different research design and gain skills in collecting, analyzing, and interpreting data using various tools and techniques
- To make the students familiar with the proposal and report writing along with publication process.

3. Methods of Instructions:

1. Lecture and discussion
2. Paper reviews
3. Presentation
4. Paper development

4. Course Contents:

Specific Objectives	Contents
Unit 1: Introduction	**8 hours**
• Explain the research process, including the sequential steps involved in conducting research from start to finish	1.1. Definition, objectives, characteristics, purpose and significance of research 1.2. Importance of research: emphasizing its role in innovation, problem-solving, and advancing technology 1.3. Types of research: fundamental research, applied research, experimental research, descriptive research, analytical research, empirical research, qualitative and quantitative approaches 1.4. Overview of the research process and its steps of research process: highlighting its key stages, such as problem identification, literature review, data collection, analysis, and reporting
Unit 2: Literature Review and Problem Statement	**8 hours**
• Describe the technique and source conducting literature review.	2.1. Conducting a literature review: techniques and sources 2.2. Identifying research gaps.

• Explain the concept of research gaps and develop the ability to analyze existing research to spot areas that are underexplored	2.3. Formulating problem statement, research questions, and hypotheses in scientific settings 2.4. Research objectives: align with the problem statement/ research questions and hypotheses, ensuring clarity and feasibility.
Unit 3: Research Design	**12 Hours**
• Provide the clear concept of different variable and measurement scale • Develop the ability to select and implement appropriate research designs based on the research objectives.	3.1. Meaning, purpose, principles and importance of research design 3.2. Element of research design 3.3. Types of research design and their uses in research 3.4. Sources of error in research design 3.5. Features of a good research design 3.6. Types of research design: exploratory, descriptive experimental, observational, and survey research: strengths and weaknesses of each 3.7. Qualitative method, quantitative method, and mixed method
Unit 4: Data Collection and Analysis	**10 Hours**
• Explain to perform collection of data using appropriate techniques. • Interpret data, draw logical conclusions, and relate findings to the research objectives.	4.1. Meaning, nature, and types of data collection; secondary and primary data-definition, sources, characteristics, advantages and disadvantage 4.2. Primary data collection methods: surveys, questionnaires, interviews, observation, and experiments 4.3. Secondary data collection: academic journals, books, online databases, government reports, and industry publications; evaluating secondary data 4.4. Concept of sampling techniques and types 4.5. Introduction to data organization, processing, and analysis
Unit 5: Research Ethics, Proposal, and Report Writing	**10 Hours**
• Understand and apply key ethical principles in research, including integrity, objectivity, confidentiality, and avoiding plagiarism. • Enable students to structure and write comprehensive research proposals, reports and presentations, covering all essential sections and ensuring accurate referencing.	5.1. Ethical principles in research: Integrity, objectivity, confidentiality, and informed consent. 5.2. Plagiarism and academic integrity: plagiarism, consequences, avoidance. 5.3. Ethical issues in data collection and reporting 5.4. Structure of a research proposal: Introduction, objectives, rationale of study, review of literature, research gap, research design and methods, expected outcomes, and timeline, discussion. 5.5. Report writing: abstract, introduction, literature review, methodology, result and discussion, conclusion and future work. 5.6. Referencing and citation: Understanding citation styles (APA, MLA, and IEEE) and the importance of accurate referencing.

	5.7. Preparing research presentations: effective visual and oral presentations of research findings. 5.8. Publication process: journal selection, manuscript preparation, and peer review.

5. Practical Works

5.1	**Laboratory work - Presentation**
	1. Development of questionnaire
	2. Data entry in any computer application
	3. Apply data filtration and validation rules
	4. Apply any statistical application tool for analysis and interpretation
5.2	**Literature Review and Analysis**
	5. Conducting a literature review
	6. Identifying research gaps, and formulating problem statement
	7. Compare different data organization, processing, and analysis technique

6. 6. Evaluation System and Students' Responsibilities:

6.1 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.

Internal Evaluation	Weight	Marks	External Evaluation	Marks
Theory		**30**	**Semester End examination**	**50**
Attendance / Class Participation	10%			
Assignments	20%			
Project Work/Presentations	10%			
Term Exam	60%			
Practical		**20**		
Attendance and Lab Participation	10%			
Lab Report	20%			
Lab Examination	40%			
Viva Examination	30%			
Total Internal Marks		**50**		
Full marks=50+50 =100				

6.2 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 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.

7. Prescribed Books and References:

Text Books

1. Kothari, C.R. (2004) Research Methodology: Methods and Techniques (2nd Edition), New Age International Publishers, New Delhi.

2. Creswell, J. W. and Creswell, J.D (2017). Research Design: Qualitative, Quantitative, and Mixed Methods Approaches (5th ed) Sage.

References

1. Leedy, P. D. and Ormrod, J. E. (2019). Practical Research: Planning and Design (12th Edition) Pearson.
2. Kumar, R (2023) Research Methodology: A Step by Step Guide for Beginners (4th Edition), SAGE Publications Pvt. Ltd; Fourth edition (13 August 2023)

Pokhara University
Faculty of Science and Technology

Course Code.: MTH 330 (3 Credit)	Full Marks: 100
Course Title: Mathematical Foundation and Computer Application (3-0-0)	Pass Marks: 45
Nature of the Course: Theory	Total Lectures: 48 hours
Level: Bachelor | Year: III / Semester: V	Program: Bachelor of Computer Application

1. Course Description:

This course is designed to provide a framework for solving complex problems mathematically. It supports for exploring abstract concept as the foundation of countless scientific and technological advancements. At its core, it deals with countable, distinct elements rather than continuous quantities. It incites mathematical reasoning, algorithms thinking, combinatorial analysis, discrete structures, application and modelling through mathematical logic, proof and induction; networks and graph theory; advanced counting techniques; and modelling computation. This course also introduces the emerging new trended concept and essence for explore its diverse applications across different domains such as encryption, networks and gaming theory, optimization, circuit analysis, structural analysis, quantum mechanics, etc. After completion of this course, students can select, implement and apply the best mathematical tools for solving real-world problems during ICT processes

2. General Objectives:

The general objectives of this course are:

1. To acquaint the students thinking solutions mathematically.
2. To develop the skills in students to select and measures the optimal techniques for real-life problem solution.
3. To acquaint the students with mathematical foundations for many computer science courses including data structures, algorithms, database theory, automata theory, formal languages, compiler theory, computer security, and operating systems, etc.
4. To acquaint the student's success in developing the problem-solving skills needed in subsequent courses and professional work.

3. Methods of Instructions:

Lecture
Tutorial/Discussion/Readings
Practice works

4. Course Contents:

Specific Objectives	**Contents**
Unit 1: The Foundations: Logic and Induction	**6 hours**
• Familiarize with basic concepts of discrete mathematics. • Explain use of logic and conclude the different statements towards a goal. • Apply different proof techniques and verify formulas using induction.	1.1 Propositional Logic and its Applications 1.2 Propositional Equivalences 1.3 Predicates and Quantifiers 1.4 Nested Quantifiers 1.5 Expressing statement in the language
Unit 2: Mathematical Reasoning	**7 hours**

• Learning the different relation available with their types and properties • Find the closure of relation.	2.1 Mathematical Induction 2.2 Strong Induction and Well-Ordering 2.3 Recursive Definitions and Structural Induction 2.4 Recursive Algorithms 2.5 Rules of Inference and proofs 2.6 Direct proof and Indirect proof 2.7 Formal and Informal proof
Unit 3: Recurrence Relations	**7 hours**
• Conceptualize the role and working procedure for sequence • Apply the concept of recursion used in both linear and nonlinear recurrence relation.	3.1 Recursive Definition of Sequences. 3.2 Differencing and Summation, 3.3 Solution of Linear Recursive Relation, 3.4 Solution of Non-linear Recurrence Relation. 3.5 Methods of Solving Recurrence Relations- *Substitution, Recurrence Tree and Master Method*
Unit 4: Graph Theory and Its Application	**14 hours**
• Understand the role of graph theory • Familiarize with basic terminologies and properties of graph, and how to simulate the real-world problem through graph	4.1 Graph Terminology– definition, direct and indirect graphs, walk, path, circuits, connected components. 4.2 Graph Models and Special Types of Graphs 4.3 Representing Graphs and Graph Isomorphism 4.4 Connectivity 4.5 Euler and Hamilton Paths 4.6 Shortest-Path Problems 4.7 Planar Graphs 4.8 Graph Coloring
Unit 5: Modelling Computation	**14 hours**
• Visualize the computing model using language, grammars • Explain the actual computation of the computer and how language and expression are perceived by the machine.	5.1 Languages and Grammars- Introduction, Phrase-structure grammars, Types of phrase-structure grammar, Derivation trees 5.2 Finite-State Machines with output and with no output- Definition, State diagram 5.3 Language Recognition- Regular sets, regular grammars, pushdown automata, Moore and Mealy machines 5.4 Turing Machines- Introduction, definition, representation and language of a Turing Machine

5. Evaluation System and Students' Responsibilities:

5.1 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.

Internal Evaluation	Weight	Marks	External Evaluation	Marks
Theory		50	Semester End	50

Attendance & Class Participation	10%			
Assignments	20%			
Presentations/Quizzes	10%			
Internal Assessment	60%			
Total Internal		**50**		
Full Marks: 50 + 50 = 100				

5.2 Students' Responsibilities:

To be eligible for the Semester End Examinations, 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.

6. Prescribed Books and References:

Test Book

1. Rosen Kenneth H., *Discrete Mathematics and Its Applications,* Seventh revised edition, 2015, Tata McGraw-Hill Publishing Company Limited, ISBN 0–07–338309–0

References

1. Sarkar, Swapan Kumar, *A textbook of Discrete Mathematics*, Eighth Revised Edition, 2013, S. Chand & Company Ltd., ISBN 81-219-2232-1
2. Richard, Johnsonbaugh, *Discrete Mathematics*, Fifth Edition, Addison Wesley, Pearson Education Asia (LPE), ISBN: 81-780-82799, 2000
3. Trus, J., *Discrete Mathematics for Computer Scientists*, Second Edition, Addison Wesley ISBN: 0-201-36061,1999