Tribhuvan University

Faculty of Humanities and Social Sciences

Office of the Dean

Balkhu, Kathmandu.

Curriculum of Bachelor of Computer Applications (BCA)

2025

Title

The title of the program is Bachelor of Computer Applications (BCA).

Objectives

The objective of the Bachelor of Computer Applications (BCA) program is to prepare human to resource for productive careers in IT industry, software development, IT Entrepreneur, and related fields, as well as to pursue higher education.

Credit and Duration of Program

Bachelor of Computer Applications (BCA) is 130 credit program. The duration of the program is four academic years. Each year is divided into two semesters. Semester duration is of 16 weeks of working days for teaching /learning excluding examinations and other activities.

Medium of Instruction and Examination

The medium of instruction and examination in the Bachelor of Computer Applications (BCA) program shall be English.

Entry Requirement

The entry requirement for students applying to the Bachelor of Computer Applications (BCA) program is the completion of an intermediate level or higher secondary level (10+2), or twelve years of schooling (or an equivalent qualification) in any discipline from a recognized institution, with at least second division marks or a minimum of a D grade in each subject and a CGPA of 1.8 or higher (as per HSEB or equivalent standards). Beside academic requirement, an entrance will be conducted for all applicants by the Dean's Office, Faculty of Humanities and Social Sciences (FOHSS).

Admission Procedure

An Entrance exam shall be conducted once a year by Dean's Office, FOHSS. Eligible successful candidates shall be admitted on merit basis.

Academic Schedule

The academic session of the program consists of two semesters per year. The fall semester begins in September and the spring semester begins in February. Dean's Office, FOHSS will publish academic calendar

Hours of Instructions

Working Days: 96 days in a semester

Class Hours:

- 3 credit hour courses with theory and lab is equivalent to 3 hours theory and 3 hours lab = 6 working hours per week.
- 3 credit hours theory-only course is equivalent 3 hours theory and 2 hours tutorial = 5 Working hours per week.

Student Evaluation

Theory course should have internal weightage of 20% and external weightage of 80%. For the course having lab work, the internal weightage is 20%, lab work weightage is 20% and external weightage is 60%. A student should secure minimum of 40% in each category to pass a course. The final score in each course will be the sum of overall weightage of in all categories. There will be a separate practical examination for the 20% weightage of lab work conducted by concerned college in the presence of an external examiner. The project work and internship are evaluated by different evaluators. To pass project work and internship, students should secure at least 40% marks in the evaluation of each evaluator and final score will be the sum of all the evaluations. For the evaluation of practical and final defence of projects and internship, an external examiner will be assigned from Dean's Office, Faculty of Humanities and Social Sciences (FOHSS).

Final Examination format (Subjects with Practical)

SN	Group	Types of Question	No. of Questions	Marks per Question	Total Marks
1	Group – A	Brief Answer Question	10	1	10 x 1 = 10
2	Group – B	Short Answer Questions	6	5	6 x 5 = 30
3	Group – C	Long Answer Questions	2	10	2 x 10 = 20

Final Examination format (Subjects without Practical)

SN	Group	Types of Question	No. of Questions	Marks per Question	Total Marks
1	Group – A	Brief Answer Question	10	1	10 x 1 = 10
2	Group – B	Short Answer Questions	8	5	8 x 5 = 40
3	Group – C	Long Answer Questions	3	10	3 x 10 = 30

To pass in a subject, a student must pass in internal and external theory and practical examinations separately with securing a minimum of 40% in that subject in internal, practical examinations and C+ grade in the final examination

Grading System

Total marks obtained in internal, practical and end semester exams shall be graded on absolute basis. The performance of a student shall be measured on a four-point scale ranging from 0 to 4 grades. A student must secure a minimum Grade Point Average (GPA) of 2.3 or Grade C plus (C+) in each course.

Grading Scale is as follows for BCA program:

Letter Grade	CGPA	Marks in Percentage	Grade Point Description
A	4.0	90 and above	Distinction
A-	3.7	80 – 89.9	Very Good
B+	3.3	70 – 79.9	First Division
B	3.0	60 – 69.9	Second Division
B-	2.7	50 – 59.9	Third Division
C+	2.3	40 – 49.9	Pass in individual Subject
F	0	Below 40	Fail

Performance of a student in a semester is evaluated in terms of the **Semester Grade Point Average (SGPA)** = Total honor points earned in a Semester / Total number of credits registered in a semester.

Performance of a student in a whole program is evaluated in terms of the **Cumulative Grade Point Average (CGPA)** = Total honor point earned / Total number of credits completed. The workload of the lecture, tutorial and practical will follow the following standard:

Lecture= hundred per cent workload (one hour lecture =one hour workload)

Tutorial= 80 per cent of the workload (one hour tutorial= 0.80 hour workload)

Practical = 50 per cent of the workload (one hour practical = 0.50 hour workload)

Attendance Requirement

Students are required to attend regularly all theory, tutorial and practical classes and should maintain minimum 80% attendance in each course separately. Those who will not able to fulfill attendance requirement are not eligible to attend in final examination of that course.

Maximum duration to complete course

Maximum time to complete the BCA course is 8 years. If any student will unable to complete the program during this maximum period of time but still he/she wants to complete the program, he/she will have to re-join from beginning following the complete admission procedure.

Award of Degree

Tribhuvan University Awards Bachelor of Computer Applications (BCA) degree after completion of all the requirement of this program as prescribed in this curriculum.

BCA Course Structure

Year 1st / Semester I						
SN	Course Code	Course Name	Credit (Cr)	Lecture (L)	Tutorial (T)	Practical (P)
1	BCA 101	Computer Fundamentals and Applications	3	3	-	3
2	BCA 102	Programming in C	3	3	-	3
3	BCA 103	Digital Logic	3	3	-	3
4	BCA 104	Mathematics-I	3	3	1	1
5	BCA 105	Professional Communication and Ethics	3	3	2	-
6	BCA 106	Hardware Workshop	1	-	-	2
Total			16	15	3	12

Year 1st / Semester II						
SN	Course Code	Course Name	Credit (Cr)	Lecture (L)	Tutorial (T)	Practical (P)
1	BCA 151	Discrete Structure	3	3	-	3
2	BCA 152	Microprocessor and Computer Architecture	3	3	-	3
3	BCA 153	OOP in Java	3	3	-	3
4	BCA 154	Mathematics-II	3	3	1	1
5	BCA 155	UX/UI Design	3	3	-	3
6	BCA156	Principles of Management	1	1		
Total			16	16	1	13

Year 2nd / Semester III						
SN	Course Code	Course Name	Credit (Cr)	Lecture (L)	Tutorial (T)	Practical (P)
1	BCA 201	Data Structure and Algorithms	3	3	-	3
2	BCA 202	Database Management System	3	3	-	3
3	BCA 203	Web Technology-I	3	3	-	3
4	BCA 204	System Analysis and Design	3	3	-	3
5	BCA 205	Probability and Statistics	3	3	-	3
6	BCA 206	Applied Economics	2	2	-	-
		Total	17	17	-	15

Year 2nd / Semester IV						
SN	Course Code	Course Name	Credit (Cr)	Lecture (L)	Tutorial (T)	Practical (P)
1	BCA 251	Operating Systems	3	3	-	3
2	BCA 252	Software Engineering	3	3	-	3
3	BCA 253	Numerical Methods	3	3	-	3
4	BCA 254	Python Programming	3	3	-	3
5	BCA 255	Web Technology-II	3	3	-	3
6	BCA 256	Project-I	2	-	-	4
		Total	17	15	-	19

Year 3rd / Semester V						
SN	Course Code	Course Name	Credit (Cr)	Lecture (L)	Tutorial (T)	Practical (P)
1	BCA 301	Computer Network	3	3	-	3
2	BCA 302	Artificial Intelligence	3	3	-	3
3	BCA 303	Advance Java Programming	3	3	-	3
4	BCA 304	MIS and e-Business	3	3	-	3
5	BCA 305	Society and Technology	3	3	2	-

6	BCA 306	Project-II	3	-	-	6
Total			18	15	2	18

Year 3rd / Semester VI

SN	Course Code	Course Name	Credit (Cr)	Lecture (L)	Tutorial (T)	Practical (P)
1	BCA 351	Computer Graphics and animation	3	3	-	3
2	BCA 352	Mobile Programming	3	3	-	3
3	BCA 353	Cryptography and Network Security	3	3	-	3
4	BCA 354	Technical Writing	2	2	2	-
5	BCA 355	Distributed System	3	3	-	3
6	BCA 356	Project-III	3	-	-	6
Total			17	15	2	18

Year 4th / Semester VII

SN	Course Code	Course Name	Credit (Cr)	Lecture (L)	Tutorial (T)	Practical (P)
1	BCA 401	Cyber Security and Ethical Hacking	3	3	-	3
2	BCA 402	Software Project Management	3	3	-	3
3	BCA403	Financial Accounting	2	2		-
4	BCA 404	Project-IV	3	-	-	6
5	BCA 405	Elective-I	3	3	-	3
6	BCA 406	Elective-II	3	3	-	3
Total			18	15	-	18

Elective I

1	BCA 404 – I	Machine Learning	3	3	-	3
2	BCA 404 – II	E-Commerce	3	3	-	3
3	BCA 404 – III	Database Administration	3	3	-	3
4	BCA 404 – IV	Linux	3	3	-	3

Elective II						
1	BCA 405 – I	Dotnet Technology	3	3	-	3
2	BCA 405 – II	Business Intelligence	3	3	-	3
3	BCA 405 – III	Software Testing and Quality Assurance	3	3	-	3
4	BCA 405 – IV	Data Visualization	3	3	-	3
Year 4th / Semester VIII						
SN	**Course Code**	**Course Name**	**Credit (Cr)**	**Lecture (L)**	**Tutorial (T)**	**Practical (P)**
1	BCA 451	Cloud Computing	3	3	-	3
2	BCA 452	Internship	3	-	-	-
3	BCA 453	Elective-III	3	3	-	3
4	BCA 454	Elective-IV	3	3	-	3
		Total	12	9	-	9
Elective III						
1	BCA 453 - I	Network Administration	3	3	-	3
2	BCA 453 – II	E-governance	3	3	-	3
3	BCA 453 – III	Database Programming	3	3	-	3
4	BCA 453 - IV	Geographical Information System	3	3	-	3
Elective IV						
1	BCA 454 - I	Digital Marketing and SEO	3	3	-	3
2	BCA 454 – II	Image Processing	3	3	-	3
3	BCA 454 – III	Internet of Things	3	3	-	3
4	BCA 454 - IV	Data Mining and Data warehouse	3	3	-	3

First Semester

COMPUTER FUNDAMENTALS AND APPLICATIONS

COURSE CODE: BCA 101

YEAR/SEMESTER: I/I **CREDIT HOURS: 3**

WORKLOAD: 6 Hrs./WEEK (THEORY: 3 Hrs., PRACTICAL: 3Hrs)

Course description:

This course covers the concepts of basic computer fundamental knowledge and its application to solve real life problems. This course including basic introduction, its types and Application various types in various fields, computer software and hardware, operating system, database management system and computer networks and recent trends technology developed and used in computer and ICT. It also aims at helping students convert theoretical concept into practical skill through the use of different application packages including word processor, spreadsheet package, presentation package and photo editing graphical package and others tools and application use in computer.

Course objectives:

The main aims of this course are to provide fundamental concepts of information and communication technology and to make students capable of using different application packages in their personal as well as professional life.

The general course objectives of this course are outlined as:

- To familiarize students with fundamental knowledge about computer system
- To make students understand software, hardware and their working procedure.
- To enhance student's knowledge about various software and its types.
- To provides the knowledge about database management system and nature of data use in computer.
- To provide knowledge about computer networking knowledge and internet use.
- To enhance students' knowledge about computer security and computer threat.
- To provide the knowledge about the recent trends and technologies use in ICT.

Course contents

Unit 1: Introduction to computer **5 hrs.**

1.1 Definition, characteristics of computer.

1.2 Anatomy of computer

1.3 Types of computers (size, principle, brand and purpose).

1.4 History of computer and generation

1.5 Application of computer

Unit 2. Computer Hardware **10 hrs.**

2.1. Basic computer Organization and Architecture

2.2. Component of computer (hardware, software, user, data and procedure).

2.3. Component of CPU (ALU, CU and RA).

2.4. Computer memory, Memory Hierarchy, Primary and Secondary memory.

2.5. Motherboard and its parts, slots, ports, interface, processor, memory chips.

2.6. BIOS, SMPS, CMOS, and Microprocessor chips.

Unit 3 Computer software **10 hrs.**

3.1. Introduction to Software, program

3.2 Types of Software (System and Application)

3.3. Operating System (Function and types).

3.4. Utility Software, Virus and Antivirus Software.

3.5. Programming language and Types of language Translator.

Unit 4. Database management System. **5 hrs.**

4.1. Introductions to data, database and DBMS

4.2. Database system Architecture.

4.3. Database Model, database Application.

4.2. SQL and No SQL concepts

4.4. Introduction to data warehousing

4.5. Data mining and concept of big data.

Unit 5: Computer Network and Internet. **5 hrs**

5.1 Introductions to Network. Intranet, Internet

5.2. Types of networks, LAN Topologies

5.3. Transmission media, Network devices

5.4. Data Communication, Transmission Mode

Humanities and Social Science
Dean's Office
T.U. Kirtipur

5.5. OSI reference model, Network Protocol.

5.6. Concept of web, and www, URL, DNS, client server

Unit 6: Computer Security **8 Hrs**

6.1. Introduction: security Threat and security attacks

6.2. Malicious Software and types of viruses.

6.3. Security Mechanisms (Cryptography, Digital Signature)

6.4. Firewall, users Authentication, intrusion Detection System

6.5. Security Awareness and Security Policy.

Unit 7: Contemporary Technology **5 hrs**

7.1. Introduction to AI, AI and Its applications.

7.2. Machine Learning, Neural network (basic concepts).

7.3. Blockchain Technology and bitcoin

7.4. IoT, cloud computing and its use

7.5. Virtual and Augmented Reality

Laboratory works **48 hrs.**

a. Office automation

b. Word-processor

- Basics options of word-processing for typing, editing, formatting margin setting viewing, designing, printing a document.
- Crating, inserting, formatting table and working with large documents in word-processing

c. Spreadsheet

- Basic sheet concepts and its shell address and other features.
- Preparing sheet for data processing like arithmetic, logical and other types of functional operation, prepared bills and invoices.
- Prepared data table for calculation, analysis and creating various charts for presentation and using different formulae for calculation and logical test.

d. Presentation

- Create the various types of slides with master slides for presentation.
- Setting slide into the required format.

1. Basic DOS commands

- Comparison of DOS and window, Switching between DOS and window
- Various Internal and external command

2. Basics of window and user interface
    - Various features of GUI base Operating System
    - Explore different files and folders
    - Control panel setting
3. Computer commination and Internet
    - Basics of computer network, WWW and websites
    - Web browsing, net surfing and search engine
    - Use of various AI tool with their purposed

Required readings:

Goel, A. (2010). *Computer fundamental,* Pearson Education India.

Leon, A. & Leon (2010). *Fundamental of information technology.* Leon Techworld

Norton, P. (2017). *Introduction to computer*, 7th Edition, McGraw Hill Education.

Sinha, P.K. (2003). *Computer Fundamentals*, BPB Publication

C PROGRAMMING

COURSE CODE: BCA-102

YEAR/SEMESTER: I/I **CREDIT: 3**

WORKLOAD: 6 Hrs/ WEEK (THEORY: 3 Hrs, PRACTICAL: 3 Hrs)

Course description:

This course provides a comprehensive introduction to the C programming language, a foundational tool in computer science and software development. Emphasizing structured and procedural programming techniques, the course covers the core concepts of C including variables, data types, operators, control structures, arrays, functions, pointers, structures, unions, and file handling. Students will gain practical skills in writing efficient, modular, and error-free code, reinforced through hands-on lab sessions. The course is designed to develop algorithmic thinking, problem-solving ability, and a strong understanding of how memory and low-level operations work in modern computing environments. This course provides the essential groundwork for more advanced in programming for software development, algorithms, and systems programming.

Course objectives:

The main objective of this course to provide students both the theoretical foundation and practical knowledge of programming using C programming language. After the completion of the course, students will be able to:

- Understand the basic structure and syntax of the C programming language, including data types, operators, and control statements.
- Develop algorithmic thinking and structured programming techniques to solve computational problems.
- Apply modular programming concepts using functions to enhance code reusability and clarity.
- Manipulate arrays, strings, and pointers effectively for data processing and memory management.
- Use structures and unions to model and organize complex data in C.
- Perform basic file input and output operations to read, write, and manage data in text and binary files.

- Write, compile, debug, and test C programs using standard development tools and environments.
- Establish a strong foundation for learning advanced programming topics, such as data structures, operating systems, and embedded systems.

Course contents

1 Unit-1 Introduction to C programming language and basics **5 Hrs.**

1.1 Evolution of programming languages

1.2 History, characteristics and applications of C.

1.3 Structure of a C program.

1.4 Compilation and execution process of a C program.

1.5 C Tokens: Keywords, identifiers, constants, string literals, operators.

1.6 Variables: Declaration, initialization, scope, and lifetime.

1.7 Data Types: Basic data types (int, char, float, double, void), derived data types, user-defined data types.

1.8 Type Casting: Implicit and explicit type conversion.

1.9 Operators and Expressions: Arithmetic, relational, logical, bitwise, assignment, increment/decrement, conditional (ternary), special operators (sizeof, comma, address-of, dereference). Operator precedence and associativity.

Unit 2: Input/Output and Control Structures **11 Hrs**

2.1 Standard Input and Output Functions:

2.1.1 Formatted I/O: printf() and scanf().

2.1.2 Unformatted I/O Functions: Character I/O (getchar(), putchar(), getch(), getche(), putch()),

2.2 Control Structures:

2.3 Decision Making Statements:

2.3.1 if statement

2.3.2 if-else statement

2.3.3 Nested if-else statement

2.3.4 else-if ladder

2.3.5 switch statement

2.3.6 Conditional operator (? :)

2.4 Looping Statements:

2.4.1 while loop

2.4.2 do-while loop

2.4.3 for loop

2.4.4 Nested loops

2.5 Jump Statements:

2.5.1 break statement

2.5.2 continue statement

2.5.3 goto statement

Unit 3: Functions, Arrays and String **11 Hrs**

3.1 Functions:

3.1.1 Introduction to functions: Advantages, function definition, function declaration/prototype,

3.1.2 function call

3.1.3 Types of functions: Library functions vs. User-defined functions

3.1.4 Function arguments: Call by value, Call by reference

3.1.5 Recursion: Concepts and examples

3.2 Arrays:

3.2.1 Introduction to arrays: Declaration, initialization, accessing elements

3.2.2 One-dimensional arrays: Processing and examples

3.2.3 Two-dimensional arrays: Declaration, initialization, accessing elements (matrix operations).

3.2.4 Multi-dimensional arrays

3.2.5 Arrays and functions

3.3 Strings:

3.3.1 Introduction to strings: Declaration, initialization

3.3.2 String input/output(gets,puts)

3.3. 3 String handling functions(strlen,strrev,strcpy,strupr,strlwr,strcmp,strcat)

3.3.4 Array of strings.

Unit 4: Structures, Unions, and Enumerations **5 hrs**

4.1 Structures and Unions:

Defining and declaring structures.

Accessing members, including nested structures, arrays of structures.

Passing structures to functions.

Defining and declaring unions; comparison with structures.

4.2 Enumerations (Enums) and typedef implementation .

Unit 5: Pointers and Memory Management **5 hrs**

5.1 Pointer declaration, initialization, and dereferencing

5.1.2 Pointer arithmetic

5.1.3 Pointers with arrays, string, function, structure.

5.2 Dynamic memory allocation: malloc, calloc, realloc, free

5.3 DMA with structure

5.4 Dangling pointers and memory leaks

Unit 6: File Handling, Command-Line Arguments, and Graphics **11 Hrs**

6.1 File Handling:

6.1.2 Concepts of files (text vs. binary) and file operations (opening, closing, reading, writing).

6.1.3 File modes and standard I/O functions (fgetc, fputc, fgets, fputs, fprintf, fscanf, fread, fwrite).

6.1.4 Random access (fseek, ftell, rewind) and error handling.

6.2 Command-Line Arguments:

6.2.1 Introduction to main() function parameters (argc, argv).

6.2.2 Passing and accessing arguments from the command line.

6.3 Graphics in C:

6.3.1 Introduction to graphics programming (basic concepts, graphics modes).

6.3.2 Drawing primitives: Points, lines, and common geometric shapes (rectangles, circles, ellipses, arcs, polygons).

6.3.3 Managing colors and fill styles for graphical output.

6.3.4 Displaying and formatting text in graphics mode.

6.3.5 Fundamentals of simple animation (clearing the screen, redrawing elements, using delays).

Laboratory work **[48 hrs.]**

The practical component will reinforce theoretical concepts through hands-on programming exercises. A minimum of 48 hours (3 hours/week for 16 weeks) will be dedicated to practical sessions. Students are expected to complete assignments on a weekly basis.

General Instructions for Practical Sessions:

- Familiarization with a C IDE (e.g., Code::Blocks, Visual Studio Code with GCC, Eclipse CDT, Dev-C++, Turbo C++).
- Understanding the compilation and execution process.
- Debugging techniques.
- Emphasis on practical application for complex topics, especially Pointers and File Handling, to ensure thorough understanding.

List of Practical Concepts to implement (with enhanced emphasis):

1. Fundamental Programming Constructs:
   - Implement programs demonstrating basic input/output, arithmetic operations, and data type conversions.
   - Apply various operators (arithmetic, relational, logical, bitwise, conditional) in expressions.
2. Control Flow Mechanisms (Practical Emphasis):
   - Gain practical proficiency in utilizing conditional statements (if, if-else, switch) to control program flow based on various conditions.
   - Implement and analyze iterative solutions using while, do-while, and for loops for repetitive tasks.
   - Demonstrate the effective use of jump statements (break, continue, goto) for specific control transfers.
3. Functions and Modularity (Practical Emphasis):
   - Design and implement modular programs using user-defined functions, focusing on effective argument passing (call by value, call by reference) and return types.
   - Explore practical applications of recursive problem-solving approaches.
4. Array and String Manipulation (Practical Emphasis):
   - Gain hands-on experience performing various operations on one-dimensional, two-dimensional, and multi-dimensional arrays, including searching, sorting, and matrix operations.
   - Implement practical solutions involving string input/output and proficiently utilize standard string handling functions.
   - Work effectively with arrays of strings for real-world data representation.
5. Pointers and Dynamic Memory (Strong Practical Emphasis):

- Develop robust programs demonstrating core pointer concepts, including declaration, initialization, and dereferencing.
- Master pointer arithmetic and deeply understand the relationship between pointers and arrays/strings for efficient memory access.
- Gain comprehensive practical experience in dynamic memory management using malloc(), calloc(), realloc(), and free() for flexible data structures.

6. Structured Data Types:
   - Define and manipulate structures, including nested structures, arrays of structures, and pointers to structures.
   - Utilize unions and understand their memory implications.
   - Implement programs using enumerated types.
7. File Operations (Practical Emphasis):
   - Achieve practical competence in performing fundamental file operations (create, open, read, write) on both text and binary files.
   - Implement formatted and block I/O operations for efficient data handling with files.
   - Navigate and manipulate file pointers for random access within files.
   - Effectively incorporate error handling mechanisms for reliable file operations.
8. Command-Line Interaction:
   - Develop programs that accept and process command-line arguments.
9. Basic Graphics Programming:
   - Initialize graphics mode and draw fundamental shapes like points, lines, rectangles, circles, and ellipses.
   - Experiment with colors and fill patterns.
   - Display and format text within graphics output.
   - Implement simple animation sequences.

Required readings

Balagurusamy, E. (n.d.). *Programming in ANSI C* (Latest ed.). Tata McGraw-Hill Education.

Deitel, P. J., & Deitel, H. M. (n.d.). *C how to program* (Latest ed.). Pearson Education.

Kernighan, B. W., & Ritchie, D. M. (n.d.). *The C programming language* (Latest ed.). Prentice Hall.

Prata, S. (n.d.). *C primer plus* (Latest ed.). Pearson Education.

DIGITAL LOGIC

COURSE CODE: BCA 103

YEAR/SEMESTER: I/I **CREDIT HOURS: 3**

WORKLOAD: 6 HOURS A WEEK LECTURE: 3 Hrs. **PRACTICAL: 3 Hrs.**

Course description

This course familiarizes students with fundamental concept of digital logic system, Number System, principles and properties of Boolean algebra and its application in simplification, gate implementation, Understand and Design Functions of Combinational Logic, Sequential Logic, Programmable Logic Devices. It also covers the use of flip flops in the design of synchronous and asynchronous sequential logic circuits.

Course objectives

- To provide the concepts used in the design and analysis of digital systems
- To enable student to design combinational logic circuit
- To enable student to design sequential logic circuit and programmable memory.

Course contents

1	**Digital Design Fundamentals and Number System**	**8 hrs**
1.1	Analog and digital Signal	
1.2	Analog and digital System	
1.3	Number System Representation	
1.4	Number System and their Conversion	
1.4.1	Binary Number System	
1.4.2	Octal Number System	
1.4.3	Decimal Number System	
1.4.4	Hexadecimal Number System	
1.5	Representation of signed numbers, Floating point number	
1.6	Complement of Number Systems	
1.6.1	r's complement	
1.6.2	r-1's complement (with r as 2 or 10)	
1.7	Binary Arithmetic	
1.8	Representation of BCD, ASCII, Excess 3, Gray Code, Error Detection Codes	

2 Boolean Algebra and its simplification **8 hrs**

2.1 Basic Logic Gates: AND, OR and NOT

2.2 Universal Logic Gates: NAND and NOR

2.3 Extended/Derived Logic Gates: Ex-OR and Ex-NOR

2.4 Boolean Algebra

2.4.1 Postulates and Theorems

2.4.2 Canonical Forms(SOP, POS)

2.4.3 Simplification of Boolean Functions using laws

2.5 Simplification of Logic Function using Karnaugh Map

2.5.1 Analysis of SOP and POS expressions

2.6 Simplification of up to 5 variable Boolean expression using Quine-McCluskey Minimization Technique (Tabular Method)

3 Combinational Logic Design **10 hrs**

3.1 Implementation of Combinational Logic Function

3.1.1 Half Adder and Full Adder

3.1.2 Half Subtractor and Full Subtractor

3.1.3. Encoders and Decoders

3.2 Implementation of data processing circuits

3.2.1 Multiplexers and Demultiplexers

3.2.2 Parallel Binary adder

3.2.3 Magnitude comparator (2bit and 4 bit)

3.2.4 Code Converters

3.2.5 Parity Generator and checker

3.3 Basic Concepts of Programmable Logic

3.3.1 ROM

3.3.2 PAL

3.3.3 PLA

4 Sequential Logic Design **10 hrs**

4.1 Concept of Sate and State Diagram

4.2 State Reduction technique

4.3. Triggering and its types

4.4 Latches and Flip-Flops (RS,D,T,JK, Master-Slave)

5 **Counters and Registers** **12 hrs**

5.1 Asynchronous and Synchronous Counter

5.1.1 Ripple counter

5.1.2 Ring counter

5.1.3 Modulus 10 Counter

5.1.4 Modulus counter (5,7,11)

5.1.5 Synchronous Design of above counters

5.2 Registers

5.2.1 Serial in Parallel out register

5.2.2 Serial in Serial out register

5.2.3 Parallel in Parallel out register

5.2.3 Parallel in Serial out register

5.2.4 Bidirectional Shift Register

Laboratory Works [48 hrs]

Tools: Digital Logic Trainer Kit, Bread Board and Simulator

1. Familiarization with Logic Gates
2. De-Morgan's Law and its familiarization with NAND and NOR gates
3. Realization of Half Adder and full Adder
4. Realization of Half subtractor and full subtractor
5. Encoder, Decoder, Multiplexer and Demultiplexer
6. Implementation of RS and D flip flop
7. Implementation of T and JK flip flop
8. Implementation of synchronous and asynchronous counter

Required readings:

Floyd, T. L. (2011). *Digital Fundamentals.* Pearson

Mano, M. (2018). *Digital Design.* Pearson

Tocci, R.J. (2001). *Digital Systems-Principles & Application.* Pearson

MATHEMATICS I

COURSE CODE: BCA 104

NATURE OF COURSE: THEORY & LAB **CREDIT HOURS** 3

YEAR/SEMESTER: I/I

Course Description:

The course covers the concepts of real numbers, functions and their graphs, sequences and series, matrices and determinants, analytical geometry, vector space, and permutations and combinations. The course is a foundation for computer applications and programming. Teachers and facilitators are suggested to connect mathematical concepts with the programming and make them more applicable to real-world problems during their theoretical and practical teaching.

Course objectives:

The objectives of this course are to make students able to

- Understand the basic mathematical concepts required to understand computer applications courses,
- Understand the nature of real numbers, their property, and compactness, functions and their types, and the use of functions in graphs.
- Solve different types of sequences and their connections to real-world problems,
- Solve the matrix-related problems and their uses in computer programming.
- Understand the standard equations and parts of the conic sections and the relation between the Cartesian and polar equations.
- Notations and meanings of vectors and their operations, and the concept of linear dependency and independence.
- Meaning and problems related to permutations and combinations.

Course content

Unit 1: Logic, Relations, Functions, and Graphs **10 Hrs.**

1.1 Elementary logic,

1.2 Real number system,

1.3 Field and ordered axioms of real numbers,

1.4 Intervals, rational and irrational numbers,

1.5 Absolute value, and its properties, complex numbers and their properties,

1.6 Ordered pairs, Cartesian product, relation, equivalence relation,

1.7 Functions, composite functions, domain and range, inverse function, types of functions (Identity, constant, algebraic, trigonometric, exponential and logarithmic), combination of functions,

1.8 Graphs of different types of functions,

Unit 2: Sequence and Series **7 Hrs.**

2.1 Sequence and Series (Arithmetic, Geometric and Harmonic) and their properties,

2.2 Means (AM, GM, and HM), and theorems to show the relation among them,

2.3 nth term and sum of arithmetic series, & finite and infinite geometric series,

2.4 Sum of first n-natural numbers, their squares and cubes,

2.5 Arithmatico-geometric series.

Unit 3: Matrices and Determinants **10 Hrs.**

3.1 Definitions and types of matrices, algebra of matrices,

3.2 Determinants, transpose, minors, and cofactors of matrices,

3.3 Properties of determinants (without proof), singular, non-singular, adjoint, and inverse of a matrix,

3.4 Rank of a matrix,

3.5 Linear and orthogonal transformation, composite transformation, and its applications to computer graphics,

3.6 Characteristic equations, Eigenvalues and Eigenvectors.

Unit 4: Analytical geometry **7 Hrs.**

4.1 Defining terms of conic sections,

4.2 Standard equations of circle, parabola, ellipse, hyperbola and their graphs.

4.3 Conic sections in terms of eccentricity,

4.4 Polar equations of the circle, ellipse, parabola and hyperbola.

Unit 5: Vector and Vector Space **7 Hrs.**

5.1 Definition of vector and scalar, magnitude and distance and unit vector,

5.2 Operations on vectors (addition, subtraction, scalar multiplication)

5.3 Scalar product and vector product of two and three vectors with their geometrical interpretations,

5.4 Vector space, subspace,

5.5 Linear combination, linear dependence and independence,

5.6 Scalar product, norm and orthogonality.

Unit 6: Permutations and Combinations **7 Hrs.**

6.1 Basic counting principle

6.2 Deduction method for the formulas for permutations and combinations

6.3 Relation between permutations and combinations.

6.4 Permutation of n objects (all different and not all different, but taking all at a time when the objects are not all different), circular permutations,

6.5 Combination of different objects and their properties.

Laboratory works

Students are expected to implement Python, MATLAB, and Mathematica to solve the numerical problems and compare the solution with that of pen and paper.

Examination Scheme				
Internal Assessment		External Assessment		
Theory	Practical	Theory	Practical	
20	20 (3hrs)	60 (3 Hrs)	---	100

Required readings

Bajracharya, B. C., (2082). *Basic Mathematics*, Sukunda Publication.

Boice, W.E., Diprima, R.C., & Mead, D. B. (2017). *Elementary Differential Equations and Boundary Value Problems*, John Wiley & Sons.

Budnick, F. S., (2019). *Applied Mathematics for Business Economics and Social Sciences*, McGraw-Hill,

Chand, K. B., Sapkota (2079), B. P., *Principles of Mathematical Analysis*.

Lay, D. C. (2003). *Linear algebra and its applications*. Pearson Education India.

Thomas G. B., Finney, R. L. (1995). *Calculus and Analytical Geometry*, Narosa Publishing House. (Text Book)

PROFESSIONAL COMMUNICATION AND ETHICS

COURSE CODE: BCA105
CREDIT HOURS: 3
SEMESTER/YEAR I/I
TEACHING HOURS: 48
COURSE TYPE: THEORETICAL

Course Description

This course is designed to develop students' professional communication competencies and ethical understanding required in the modern digital and organizational landscape. It emphasizes communication strategies, formats, and tools used in professional settings, with a strong foundation in workplace ethics and social responsibility. The course bridges communication theory with practical business, technical, and interpersonal communication skills, integrating ethical decision-making frameworks relevant to IT and business contexts in Nepal and beyond.

Course objectives

By the end of this course, students will be able to:

- Understand the principles and practices of effective professional communication in oral written, and digital forms.
- Develop effective professional communication skills.
- Understand ethical dilemmas in business communication.
- Apply persuasion, negotiation, and conflict resolution ethically.
- Analyze ethical issues in professional and technological contexts.
- Collaborate effectively in team-based communication projects.

Learning Outcomes

Upon successful completion, students will be able to:

- communicate clearly and persuasively in professional settings.
- compose professional documents such as résumés, cover letters, emails, and reports.
- deliver effective oral presentations with appropriate visual and technological support.
- identify and resolve ethical dilemmas in organizational and IT-related contexts.
- work collaboratively and ethically within teams and multicultural environments.
- evaluate the impact of communication and ethics in professional growth and social change.

Course Contents

Unit 1:Foundation of Professional Communication **8 LH**

1.1 Fundamentals of Language

1.2 A Brief History of Professional Communication

1.3 Principles of Professional Communication

1.4 The Communication Process

1.5 Nonverbal Communication in the Workplace

1.6 Barriers to Effective Communication

1.7 Vocabulary and Grammar

1.7.1 Commonly Confusing Words

1.7.2 Use of Tenses

1.8 Readings:

1.8.1 "Mother Tongue" by Amy Tan

1.8.2 "The Letter" by Dhumketu

Unit 2: Oral Communication **8 LH**

2.1 Telephone Conversation

2.2 Public Speaking and Presentation Skills

2.3 Meeting, agendas and minutes

2.4 Elevator Pitches

2.5 Interviewing and Professional Dialogue

2.6 Vocabulary and Grammar

2.6.1 Professional Idioms

2.6.2 Reported Speech

2.7 Readings:

2.7.1 "Death by PowerPoint" by Angela R. Garber

2.7.2 "Our world on fire needs you" by Maria Ressa

Unit 3: Writing Professionally **8 LH**

3.1 Rules of Professional Writing

3.2 Text Messages, Emails, and Memos

3.3 Notice Writing

3.4 Informal and Formal Letters

3.5 Résumés and Cover Letters

3.6 Vocabulary and Grammar

3.6.1 Business Vocabulary and Its Uses

3.6.2 Active and Passive Voice

3.7 Readings:

3.7.1 "Gateman's Gift" by R. K. Narayan

3.7.2 "My School" by Rabindranath Tagore

Unit 4: Interpersonal and Group Communication **LH 8**

4.1 Interpersonal Communication in the Workplace

4.2 Listening Environment in the Workplace

4.3 Intercultural Communication

4.3 Effective Communication in Teams

4.4 Leadership and Communication

4.5 Adapting Communication Styles for Diverse Audiences

4.6 Vocabulary and Grammar

4.6.1 Polite Words in Communication

4.6.2 Making Requests and Offers

4.6.2 Conditional Sentences

4.7 Readings:

4.7.1 "Computer and the Pursuit of Happiness" by David Gelempter

4.7.2 "The Collapse of the Family and the Community" by Yuval Noah Harari

Unit 5: Digital Communication **8 LH**

5.1 Virtual Etiquette/Netiquette

5.2 Online Professionalism

5.3 Communicating via Social Media and Collaborative Tools

5.4 Plagiarism and Online Plagiarism

5.5 Visuals in Communication (Maps, Tables, Charts, Infographics, Icon, Photograph, Diagrams)

5.6 Vocabulary and Grammar

5.6.1 Social Media Vocabulary and Its Uses

5.6.2 Preposition of Time, Place and Direction

5.7 Readings

5.7.1 "Cat Pictures Please" by Naomi Kritzer

5.7.2 "ChatGPT May Be Eroding Critical Thinking Skills" *by* Andrew R. Chow

Unit 6: Professional Ethics **8 LH**

6.1 Introduction to Professional Ethics in Computing

6.2 Professional Codes of Ethics

6.3 Responsibilities of IT Professionals

6.4 Ethical Decision-Making in IT

6.5 Whistle-Blowing and Professional Integrity

6.6 Workplace Ethical Challenges in IT

6.7 Vocabulary and Grammar

6.7.1 Ethical Vocabulary

6.7.2 Concord (Subject Verb Agreement)

6.8 Readings

6.8. 1"The Necklace" by Guy de Maupassant

6.8.2 "The Digital Citizen" by Lugi Ceccarini

Teaching Methodologies and Evaluation Scheme

- Interactive Lectures and Discussions
- Role Plays and Simulations
- Group Presentations and Peer Reviews
- Written Assignments and Reflective Journals

Assessment

- Class Participation and Attendance: 5%
- Assignments and Written Tasks: 10%
- Mid-term Exam: 15%
- Group Presentation/Project: 10%
- Final Examination: 60%

Required readings

Adhikari, B. & Subedi, S. (2024). *New modern grammar, writing, comprehension, word formation and pronunciation*. Heritage Books.

Adler, R. B., & Elmhorst, J. M. (2019). *Communicating at Work: Principles and Practices for Business and the Professions* (11th ed.). McGraw-Hill.

Bovee, C. L., & Thill, J. V. (2021). *Business Communication Today* (15th ed.). Pearson.

Ghillyer, A. W. (2017). *Business Ethics: Now* (5th ed.). McGraw-Hill Education.

Guffey, M. E., Loewy, D., & Almonte, R. (2020). *Essentials of Business Communication* (11th ed.). Cengage Learning.

Quinn, M. J. (2014). *Ethics for the Information Age* (6th ed.). Pearson.

Spinello, Richard A. (2020). *CyberEthics: Morality and Law in Cyberspace.* Jones & Bartlett

HARDWARE WORKSHOP

COURSE CODE: BCA106
YEAR/SEMESTER: I/I
TOTAL CREDIT: 1

PRACTICAL HOURS: 2 HRS. PER WEEK

Course Description:

This course provides a hands-on foundation in computer hardware, electronics, and networking through practical exploration of system components and real-world troubleshooting. Students will learn to identify and test electronic components, assemble and disassemble desktops and laptops, and configure both wired and wireless networks. The training emphasizes diagnostic methods using tools like Multimeters, POST cards, and cable testers to resolve hardware and connectivity issues. Additionally, learners will gain experience in preventive maintenance and system upgrades, including RAM, storage, PSU, and BIOS configurations. The course concludes with a comprehensive practical assessment, equipping students with the skills needed for IT support and hardware technician roles.

Course objectives:

- To understand basic components of computer hardware and networking.
- To learn to assemble, disassemble, and troubleshoot both hardware and network setups.
- To gain hands-on experience in configuring and testing basic networks.
- To develop diagnostic and maintenance skills for common IT issues.
- To build foundational skills required for careers in IT support,
- To gain system administration role and network troubleshooting.

Course Outcomes:

1. Identify and describe the functions of key hardware and network components.
2. Assemble and configure basic PCs and networks.
3. Diagnose and troubleshoot common hardware and network problems.
4. Perform maintenance and apply preventive measures for IT systems.

Syllabus Breakdown:

1. **Introduction to Hardware (Electronic Components) -** **5hr**

   1.1.Introduction to Electronics & Safety

   - Lab safety demo, basic electronics kit introduction

   1.2.Passive Components: Resistors, Capacitors, Inductors

- Identify by color code & markings, using multimeter

1.3.Active Components: Diodes, LEDs, Transistors, ICs

- Orientation, pinouts, forward/reverse bias testing
- MOSFET (NPN & PNP logic gate)

1.4.Connectors, Switches, Relays, Buzzers, Crystals

- Identification, test switching/buzzing circuits

1.5.Reading Datasheets and Packaging Types

- DIP, SMD, TO-92, TO-220, etc.

1.6.Testing and Troubleshooting Components

- Faulty vs working component tests, use of ESR meter
- Mini Project or Component Identification Test

Tools & Equipment Needed

a) Digital Multimeter (DMM)
b) Analog/Digital Component Tester
c) Breadboards and jumper wires
d) Assorted electronic component kits
e) Datasheets and charts

2. Introduction of Networking components **2hrs**

2.1.Overview of Network Devices

- Identify modem, NIC, router, switch, hub, access point

2.2.Cabling & Connectors

- Crimp RJ45, identify CAT5/6, Coaxial, Fiber-optic, Rollover, Ethernet crossover, Straight Through cable

2.3.Testing and troubleshooting Common physical Issues

- Simulate network faults, replace cables/devices
- Cable tester

Tools & Equipment Needed

a) Wireless Router, Ethernet Switch, Hub
b) RJ45 cables (CAT5/6), Crimping Tools
c) LAN Tester, Multimeter
d) Access Points, Extenders

e) PCs or Laptops for testing

f) Console cables (for Cisco/basic CLI config)

3. Desktop and Laptop Hardware Identification and Handling 6hrs

3.1.Introduction to PC Hardware

- Identify components: Case, motherboard, CPU, RAM, ROM, PSU
- Demonstration different types of Case, motherboard, RAM (SD, DDR, DDR1234), ROM (PROM, EPROM, EEPROM), PSU.
- Identify components front-panel and back-panel

3.2. Preparing the Workbench & Safety

- ESD safety, tool handling, workspace preparation

3.3.System Unit Components

- Motherboard, CPU, southbridge, Northbridge, RAM, Power Supply, BIOS/UEFI-Internal Buses and Slots
- Identify components in a system- Label and describe functions

3.4.Storage Devices

- HDD, SSD, Optical Drives- Comparison and Interfaces (PATA, SATA, SSD, NVMe)
- Remove & reinstall a drive- Show real vs fake SSD

3.5.Input/Output Devices

- Keyboard, Mouse, Monitor, Printer, Scanner
- Connect and test devices- Discuss port types (USB, HDMI, etc.)

3.6.Understanding, role and Testing the Power Supply Unit (PSU)

3.6.1 PSU Overview (ATX, Modular, SMPS)

- Show different PSU models and internal parts

3.6.2 PSU Connectors and Their Purposes

- Hands-on with 24-pin, 4+4 CPU, SATA, PCIe

3.6.3 Voltage Standards (3.3V, 5V, 12V rails)

- Explain color-coded wires and voltages

3.6.4 Testing PSU with Multimeter

- Measure voltages at different outputs (live demo)

3.6.5 Common PSU Failures

- Show burnt connectors, fan failure, no power scenario

3.6.6 Safety Tips for Handling PSUs

- Do's and Don'ts (e.g., discharge capacitors)

4. PC and Laptop Assembly **6hrs**

- Safety precautions- Step-by-step assembly of a PC
- Group hands-on assembly task- Boot to BIOS

4.1.Installing Motherboard, CPU, and RAM

- Mount motherboard, RAM, Install CPU, apply thermal paste

4.2.Connecting PSU, Storage, and Cables

- Connect SATA, power cables, front panel ports

4.3.BIOS/UEFI Configuration & POST Checks

- Access BIOS, set boot order, set password, run POST tests

4.4.Troubleshooting Basics

- POST, Beep Codes- Common errors and solutions
- Use diagnostic tools (multimeter, POST card)- Simulate hardware errors

4.5.Installing Operating System (Optional)

- Boot from USB/DVD, install Windows/Linux
- BIOS updates, driver updates, disk cleanup, antivirus scan
- Virtual/real printer installation
- Smart and Useful commands of OS
- Cross-Platform Useful Commands

4.6.Troubleshooting & Diagnostics

- Simulate no boot, no power, display error, beep codes
- Simulate boot failure, no power, overheating, connectivity issues

4.7.Final Practical Assessment

- Complete PC build, POST, and viva questions

Tools & Equipment Required:

a) ATX-compatible PC components (CPU, RAM, PSU, Motherboard, Storage, Case)
b) ESD wrist straps, screwdrivers, thermal paste
c) Monitor, keyboard, mouse, USB OS boot media
d) Diagnostic cards, multimeter, BIOS reset jumpers

e) ATX Power Supply (working and faulty units)

f) Desktop and Laptop Motherboard (for demo)

g) PSU Tester and Multimeter

h) Dummy Load or Old PC for testing

i) Screwdrivers and safety gear

5. **Basic Network Device and Deployment** **2hrs**

5.1. Identification of Different Network devices. (Modem, NIC, Hub, Switch, Router, AP)

- Configure topology
- Configure crossover/ straight through cable with devices

5.2.Sharing

- Printer Technologies
- Install, Configure, and Maintain Printers
- File/Folder

6. **Basic Networking Configuration** **2hrs**

6.1.IP Addressing, Subnetting

- IP class, Subnet Mask, Gateway, DNS

6.2.Router & Switch Configuration

- Configure IP address, test connectivity, console setup

6.3.Wireless Access Points (AP) & Extenders

- Setup SSID, password, MAC filtering, signal test

6.4.Network Design and Topologies

- Build and test star, bus topologies using real devices

6.5.Testing & Diagnostics

- Use ping, tracert, ipconfig, netstat, arp, nslookup, route, netsh etc.
- cable testers

6.6.Troubleshooting Common Issues

- Simulate network faults, replace cables/devices
- Mini LAN/WLAN setup task with demo and viva

7. **Problem Solving Techniques** **2hrs**

7.1. Start with visual inspection: loose cables, damaged ports.

7.2. Use elimination method: test one part at a time.

7.3. Check for BIOS/UEFI messages and beep codes.

7.4. Use ping, ipconfig/ifconfig, and network tester tools for network issues.

7.5. Maintain a troubleshooting log for repeated issues.

7.6. Follow a systematic approach: Identify → Test → Isolate → Fix → Verify.

8. **Maintenance & Upgrade** **5hrs**

8.1.Hardware Upgrade Tips

8.1.1 Upgrade Planning & Compatibility Checks

- Check motherboard compatibility for CPU, RAM, and GPUs.
- Verify laptop upgrade limitations (RAM max, NVMe slots, etc.).
- Use manufacturer's documentation & tools like(CPU-Z, Speccy, Crucial System Scanner)

8.1.2 Memory (RAM) Upgrades

- Match speed, voltage, and form factor (SO-DIMM for laptops, DIMM for desktops).
- Dual-channel RAM installation for performance boost.
- BIOS/UEFI settings for recognizing upgraded RAM.

8.1.3 Storage Upgrades

- Compare SSD vs HDD vs NVMe (speeds, durability, boot time).
- Use cloning tools for OS/data migration.
- Align partitions properly for SSDs.

8.1.4 Graphics Card (GPU) Upgrade (Desktop)

- Check PSU wattage and physical space inside the case.
- Verify PCIe slots and cooling requirements.
- Install latest drivers after upgrade.

8.1.5 CPU Upgrade Tips (Desktop)

- Check socket type, chipset support, and BIOS compatibility.
- Apply thermal paste correctly and test with stress tools (Prime95, AIDA64).

8.2.Preventive Maintenance Tips

8.2.1 Cleaning Internal Components

- Use compressed air, soft brushes, and isopropyl alcohol.
- Clean fans, heat sinks, and power supplies every 6 months.
- For laptops, carefully clean vents and fans without damaging small connectors.

8.2.2 Thermal Paste Replacement (Advanced Task)

- Remove old thermal paste from CPU/GPU using alcohol.
- Apply a small pea-sized drop of thermal paste before mounting cooler.
- Monitor temperatures post-application using HWMonitor or CoreTemp.

8.2.3 Battery Care for Laptops

- Avoid overcharging or deep discharges frequently.
- Calibrate battery every 3-6 months (full charge and discharge cycle).
- Replace swollen or degraded batteries immediately.

8.2.4 BIOS/UEFI & Firmware Updates

- Update only if needed (security patch, hardware support).
- Follow manufacturer instructions strictly to avoid "bricking" the system.
- Backup settings before flashing BIOS.

8.2.5 Cable Management & Airflow Optimization

- Route cables away from fans and components to improve airflow.
- Use cable ties, Velcro straps, or case grommets.
- Keep air intakes and exhausts free from obstructions.

8.2.6 Health Monitoring & Diagnostics

- Regularly check SMART status of drives.
- Run memory diagnostics every few months (MemTest86+).
- Monitor CPU/GPU temperatures under normal and load conditions

9. Final Practical Assessment **2hrs**

9.1. Hardware Identification and Handling: Label parts, explain function, use anti-static measures.

9.2. PC Assembly/Disassembly: Step-by-step guided practice using demo PCs.

9.3. Network Setup: Configure wired LAN, assign IPs, test with ping.

9.4. Troubleshooting Simulations: Use POST card, interpret beep codes, fix connectivity issues.

9.5. Maintenance Tasks: Clean internal components, apply thermal paste, verify power supply.

Internal Assessment Breakdown for (40 marks)

- Attendance & Participation – 10%
- Daily Lab Performance – 20%
- Practical Test – 50%
- Viva/Oral Exam – 20%

Methodologies for Delivery:

- The course must be conducted regularly and intensively to effectively accomplish the learning objectives of the subject matter.
- Demonstration-Based Learning: Live setup of hardware and networking devices.
- Hands-On Practice: Students carry out assembly, configuration, and troubleshooting tasks.
- Interactive Discussions: Q&A sessions to support conceptual clarity.
- Group Work: Pair-based assembly and networking tasks.
- Case-Based Problem Solving: Realistic scenarios for diagnosis and resolution.
- Simulation Tools: Optional use of Packet Tracer or VirtualBox.
- Checklists and Worksheets: To support step-by-step execution.

Examination Scheme

This Course doesn't need final Examination. The evaluation will be based on:

Internal Assessment	External Assessment	Total Marks
Theory + Practical	Viva/Oral	100 marks
20+60 marks	20 marks	

Required readings:

Gookin, D. (2017). *Troubleshooting & maintaining your PC all-in-one for dummies®* (3rd ed.). John Wiley & Sons, Inc.

Halsey, M. (2022). *Troubleshooting and supporting Windows.* Apress.

Mueller, S. (2005). *Upgrading and repairing laptops* (2nd ed.). Que Publishing.

Roberts, R. M. (2010). *Networking fundamentals* (2nd ed.). Goodheart-Willcox Publisher.

Singh, V. P. (2025). *Laptop repairing course* (with Windows 8). Computech Publications.

Yusufu, A. M. (2014). *A guide to electronic maintenance and repairs*. Partridge Singapore.

Zacker, C., & Rourke, J. (2001). *PC Hardware: The Complete Reference* (Illustrated ed.). Osborne/McGraw-Hill.

Second Semester

DISCRETE STRUCTURE

COURSE CODE: BCA 151
YEAR/SEMESTER: I/II
WORKLOAD: 6 HOURS A WEEK

CREDIT HOURS: 3
TEACHING HOURS: 48
LECTURE: 3 Hrs.
PRACTICAL: 3 Hrs.

Course description:

This course is designed to build a strong foundation in discrete mathematics—the kind of math that powers the world of computer science. It sharpens logical thinking, introduces techniques for writing solid mathematical proofs, and strengthens problem-solving skills needed in programming, algorithm design, and system development. Students will dive into key topics like logic, sets, functions, relations, combinatorics, graphs, trees, and Boolean algebra, learning not just the theory but also how these concepts apply in real tech scenarios like databases, circuits, and code structure. Through hands-on practice with tools like Python, Jupyter Notebooks, NetworkX, and Graphviz, students will bring these abstract ideas to life by building models, writing logic-based programs, and exploring how mathematics directly supports computing.

Course objectives:

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

- develop the ability to think logically and construct valid mathematical arguments.
- apply set theory concepts such as set operations, Venn diagrams, Cartesian products, and power sets to solve problems in databases, programming, and decision-making structures.
- analyze relations and functions using matrix and graph representations and understand applications in modeling data and structures.
- use combinatorial techniques to solve real-life counting and arrangement problems.
- explore graph and tree structures, implement traversal algorithms (DFS, BFS, etc.), and apply these concepts in computer science domains such as networking, compiler design, and operating systems.

Course contents

1 Set Theory **6 hrs.**

1.1. Basic Concepts: Sets, elements, roster and set-builder notation, cardinality.

1.2. Set Relationships:

1.2.1. Subsets

1.2.2. Proper subsets

1.2.3. Universal set

1.2.4. Complement

1.2.5. Disjoint sets.

1.3. Set Operations

1.3.1. Union

1.3.2. Intersection

1.3.3. Difference

1.3.4. Complement

1.3.5. Symmetric difference.

1.4. Venn Diagrams: Visual representation of set relationships and operations.

1.5. Set Identities: Proof of identities using algebraic and Venn diagram methods.

1.6. Cartesian Products: Ordered pairs, cross product of two or more sets.

1.7. Power Sets: Definition and computation of power sets.

1.8. Applications: Use of sets in databases, computer programming, and decision structures.

2. Logic and Propositional Calculus **8 hrs.**

2.1. Propositions and Logical Operators: Definition of propositions, types (simple, compound), logical connectives: AND, OR, NOT, IMPLICATION, BICONDITIONAL.

2.2. Truth Tables: Constructing truth tables for expressions involving logical operators.

2.3. Tautologies, Contradictions, and Contingencies: Identifying always true/false/logical expressions.

2.4. Logical Equivalence and Implications: Laws of logic (De Morgan's, distributive, associative, etc.), verifying equivalences.

2.5. Predicate Logic and Quantifiers: Introduction to predicates, universal and existential quantifiers.

2.6. Rules of Inference: Modus ponens, modus tollens, hypothetical syllogism, and others.

2.7. Proof Methods: Direct, indirect, contradiction, contrapositive, and proof by cases.

3. Relations and Functions **8 hrs.**

3.1. Relations: Definition, Binary Relation, Representation, Domain, Range, Universal Relation, Void Relation, Union, Intersection, and Complement Operations on Relations

3.2. Properties of Binary Relations in a Set : Reflexive, Symmetric, Transitive, Anti-symmetric Relations

3.3. Relation Matrix and Graph of a Relation; Partition and Covering of a Set, Equivalence Relation, Equivalence Classes, Compatibility Relation, Maximum Compatibility Block, Composite Relation, Converse of a Relation,

3.4. Transitive Closure of a Relation R in Set X, examples from real-world scenarios.

3.5. Representation of Relations: Using matrices and directed graphs (digraphs).

3.6. Equivalence and Partial Order Relations: Properties and examples, Simple or Linear Ordering, Totally Ordered Set (Chain), Frequently Used Partially Ordered Relations, Representation of Partially Ordered Sets, Hesse Diagrams, Least & Greatest Members, Minimal & Maximal Members, Least Upper Bound (Supremum), Greatest Lower Bound (infimum), Well-ordered Partially Ordered Sets (Posets). Lattice as Posets, complete, distributive modular and complemented lattices Boolean and pseudo Boolean lattices. (Definitions and simple examples only)

3.7. Closures and Composition of Relations: Reflexive, symmetric, transitive closures.

3.8. Functions: Definition, domain, co-domain, range, examples.

3.9. Types of Functions: Injective (one-to-one), surjective (onto), bijective (one-to-one correspondence).

3.10. Inverse and Composition of Functions: Definitions and computations.

3.11. Applications: Use in programming, data mapping, and relational databases.

4. Mathematical Reasoning and Proof Techniques **6 hrs.**

4.1. Mathematical Reasoning: Basic structure of arguments, logical flow.

4.2. Mathematical Induction: Principle of induction, proof by induction, applications in series and recursive definitions.

4.3. Strong Induction: Differences from regular induction, applications.

4.4. Recursive Definitions: Defining sequences and structures recursively.

4.5. Structural Induction: Proofs involving recursively defined structures like trees and lists.

4.6. Applications: Problem-solving and validation of algorithms.

5. Combinatorics and Counting Principles **5 hrs**

5.1. Basic Counting Principles: Introduction to counting, rule of sum and rule of product with real-world examples (e.g., menu combinations, clothing combinations).

5.2. Permutations and Combinations: Concepts of ordered and unordered selections, factorial notation, formulae for permutations (nPr) and combinations (nCr), applications in password generation and team selection.

5.3. Pigeonhole Principle: Understanding the concept, simple and strong pigeonhole principle, applications such as birthday paradox, drawer problems, and error checking.

5.4. Inclusion-Exclusion Principle: Set-based approach to solving overlapping sets, solving problems involving counting elements in unions of sets (up to three sets), and its application in probability and combinatorics.

6. Graph Theory and Trees **12 hrs.**

6.1. Graphs: Introduction, definition, examples; Nodes, edges, adjacent nodes, directed and undirected edge, Directed graph, undirected graph, examples; Initiating and terminating nodes, Loop (sling), Distinct edges, Parallel edges, Multi-graph, simple graph, weighted graphs, examples, Isolated nodes, Null graph; Isomorphic graphs, examples; Degree, Indegree, out-degree, total degree of a node, examples

6.2. Subgraphs: definition, examples; Converse (reversal or directional dual) of a digraph, examples;

6.3. Path: Definition, Paths of a given graph, length of path, examples; Simple path (edge simple), elementary path (node simple), examples; Cycle (circuit), elementary cycle, examples;

6.4. Reachability: Definition, geodesic, distance, examples; Properties of reachability, the triangle inequality; Reachable set of a given node, examples, Node base, examples;

6.5. Connectedness: Definition, weakly connected, strongly connected, unilaterally connected, examples; Strong, weak, and unilateral components of a graph, examples, Applications to represent Resource allocation status of an operating system, and detection and correction of deadlocks;

6.6. Matrix representation of graph: Definition, Adjacency matrix, boolean (or bit) matrix, examples; Determine number of paths of length n through Adjacency matrix, examples; Path (Reachability) matrix of a graph, examples; Warshall's algorithm to produce Path matrix, Flowchart

6.7. Types of Graphs: Simple, multigraph, weighted, directed/undirected, complete, bipartite.

6.8. Graph Traversal: Breadth-First Search (BFS), Depth-First Search (DFS).

6.9. Trees: Trees: Definition, branch nodes, leaf (terminal) nodes, root, examples;

6.10. Different representations of a tree, examples; Binary tree, m-ary tree, Full (or complete) binary tree, examples;

6.11. Converting any m-ary tree to a binary tree, examples;

6.12. Representation of a binary tree: Linked-list; algorithms; Applications of List structures and graphs

6.13. Tree Traversals: Inorder, preorder, postorder traversal techniques.

6.14. Applications: Networking, pathfinding algorithms, compiler syntax trees, file systems.

7. Algebraic Structures **3 hrs.**

7.1. Binary Operations: Definition and examples of binary operations on sets.

7.2. Algebraic Systems: Semigroups, monoids, and groups - axioms and properties.

7.3. Group Theory Basics: Identity element, inverse, associativity, examples with integers and matrices.

7.4. Boolean Algebra: Basic postulates and theorems, duality, Boolean functions.

7.5. Logic Circuits: Simplification of logic circuits using Boolean expressions.

7.6. Applications: Automata theory, logic design, cryptography

Practical **(48 Hours)**

Instructor should encourage the use of open-source tools like Python, Jupyter Notebooks, NetworkX, and Graphviz for practical sessions.

Practical Report Contents: Theory, Source code, Output, Conclusion

1. Truth tables, logical operators, tautologies using Python or logic tools
2. Set operations implementation and Venn diagram visualization
3. Relation properties and matrix/digraph representation in code
4. Function definitions and mapping types with coding examples
5. Recursion-based programs (factorial, Fibonacci), validating inductive proofs
6. Permutations and combinations via iteration and recursion
7. Graph representation: adjacency list and matrix (using NetworkX or code)
8. Graph traversal: DFS and BFS implementations
9. Tree structures and traversals (inorder, preorder, postorder)
10. Boolean algebra simplification and expression evaluation via code
11. Mini project: Logical puzzle solver or graph simulation

Required readings:

Grimaldi, R. P. (3rd ed.). Discrete and Combinatorial Mathematics: An Applied Introduction. Pearson

Kolman, B., Busby, R. C., & Ross, S. (2000). *Discrete mathematical structures* (3rd ed.). Prentice Hall.

Liu, C. L. (1985). *Elements of discrete mathematics* (2nd ed.). McGraw-Hill.

Rosen, K. H. (2011). *Discrete mathematics and its applications* (8th ed.). McGraw-Hill.

Stein, C., & Drysdale, R. L. (2010). Discrete Mathematics for Computer Scientists. Pearson Education.

Required Journals/ Articles:

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to algorithms* (3rd ed.). MIT Press.

Hopcroft, J. E., & Tarjan, R. E. (1973). Algorithm for finding the strongly connected components of a directed graph. Communications of the ACM, 16(6), 372–378. https://doi.org/10.1145/362248.362272

Knuth, D. E. (1968). The art of computer programming: Volume 1: Fundamental algorithms. Addison-Wesley.

Pippenger, N. (1978). Theories of computation. Journal of Computer and System Sciences, 16(1), 99–115. https://doi.org/10.1016/0022-0000(78)90030-6

MICROPROCESSOR AND COMPUTER ARCHITECTURE

COURSE CODE: BCA 152

NATURE OF COURSE THEORY + LAB **CREDIT HRS: 3**

YEAR/SEMESTER: II /I **WORK LOAD/THEORY/LAB: 3/3 Hrs**

Course description:

This course is designed to familiarized the fundamental knowledge about computer architecture, instruction cycle, components of microprocessor, Intel 8085 and assembly programming.

Course objectives:

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

- Understand the basic components of a microprocessor
- Explain the block diagram of Intel 8085
- Demonstrate assembly language programming using Intel 8085
- Interpret timing diagrams, instruction cycles, and machine cycles
- Explain the role of the control unit and central processing unit (CPU)
- Differentiate between RISC and CISC architectures
- Describe the concept of Direct Memory Access (DMA)
- Explain memory organization and operations
- Understand the concept of pipelining in processors
- Describe microprogramming and microinstructions
- Perform computer arithmetic operations such as multiplication and division

Course contents:

Unit 1	**Introduction to Microprocessor**	**3 Hours**
1.1	Definition of Microprocessor Components: Registers, ALU, Control and Timing, System Buses (Address, Data, Control), Microprocessor System with Bus Organization, Application of MP	
Unit 2	**8085 Microprocessor**	**12 Hours**

2.1 Functional Block Diagram, Pin Configuration, Description of each Block: Registers, Flag (Description of each Flag), Multiplex Address data bus (Ad0-Ad7), Timing and Control Unit, Interrupts (Introduction Only), Addressing Modes, Instruction cycle, Machine cycle (Opcode, fetch, memory read, memory write) and T states, Timing diagram of MOV,LDA, STA, MVI

2.2 8085 Instruction Set
Data Transfer: - MOV, MVI, STA, LDA, LXI, LDAX, STAX, XCHG
Arithmetic: - ADD, ADI, ADC, SUB, SUI, SBB, INR, DCR, INX, DCX,
Logic: - ANA, ANI, ORA, ORI, XRA, XRI, CMA, CMP
Branching: - JMP, JNZ, JZ, JNC, JC

2.3 Basic Assembly Language Programming using 8085 Instruction Sets Addition (8 and 16 bit), Subtraction (8 and 16 bit), Multiplication (8 bit) and Division (8 bit), Simple Sequence Program, Array Searching using branching and looping

Unit 3 8086 Microprocessor **4 Hours**

3.1 Logical block diagram and components, Bus interface unit and Execution Unit, flag, pipeline concept, Memory Segmentation, Segmentation register

Unit 4 Basic Computer Architecture and Design **6 Hours**

4.1 Stored Program Organization, Computer Registers, Common bus system, Instruction set, Timing and Control-Instruction Cycle

4.2 Micro-Operation, Arithmetic Micro Operations: Addition, Subtraction, Increment, Decrement, Logic Micro Operations: AND, OR, NOT, NAND, NOR, XOR, Shift Micro Operations: Logical, Circular and Arithmetic

Unit 5 Microprogrammed Control Unit **5 hrs.**

5.1 Hardwired vs micro program CU, Control Memory, Address Sequencing, Micro-operation, Micro instruction, Micro Instruction Format, Micro-program: Symbolic and Binary Micro-program (FETCH)

Unit 6: Central Processing Unit **6 Hours**

6.1 Introduction, General Register Organization, Stack Organization,

Instruction Formats:3,2,1,0 address Instruction,

6.2 RISC and CISC architecture

Unit 7: Computer Arithmetic **3 hrs.**

7.1 Addition and Subtraction with signed magnitude data, Addition and Subtraction with signed 2's complement data, Booth Multiplication

Unit 8 Input and Output Organization and Memory Organization **5 Hours**

8.1 Introduction to Peripheral Devices, I/O interface-I/O bus and Interface Modules, Isolated versus Memory Mapped I/O, Interrupt

8.2 Direct Memory Access (DMA): Introduction, Basic DMA Procedures (DMA controller only)

8.3 Hierarchy of Memory System

8.4 Primary Memory: RAM and ROM, Memory Address Map with examples of Address Decoding.
Secondary Memory: Structure of Magnetic Disk, Cache Memory

Unit 9 Pipelining **4 Hours**

9.1 Concept of Pipelining and Flynn's Classification, Pipelining Example with Speed Up Ratio

9.2 Arithmetic Pipeline, Pipeline for Floating-point Addition and Subtraction

9.3 Instruction Pipeline: Four Segment Instruction Pipeline

9.4 Data Dependency, Handling of Branch Instruction, pipeline hazard and its solution

Laboratory Works: **48 Hours**

- Write assembly language programs using both the 8085-microprocessor trainer kit and a software-based 8085 simulators.
- Demonstrate the use of all types of instructions and various addressing modes available in the 8085-instruction set.
- Develop programs that include fundamental arithmetic operations (8-bit addition and subtraction,16-bit addition and subtraction,8-bit multiplication and division), logical operations, loops, bitwise manipulation, and branching techniques.
- Implement algorithms for computer arithmetic using high level language

Required readings

Gaonkar, R. S. (1998). *Microprocessor architecture, programming, and applications with the 8085*. Prentice-Hall, Inc.

Hall, D. V. (1986). *Microprocessors and interfacing: programming and hardware*. McGraw-Hill, Inc.

Mano, M. M. (1993). *Computer system architecture*. Prentice-Hall, Inc.

OOP IN JAVA

COURSE CODE: BCA 153
YEAR/SEMESTER: I/II
WORKLOAD: 6 HOURS A WEEK

CREDIT HOURS: 3
LECTURE: 3 Hrs.
PRACTICAL: 3 Hrs.

Course description:

This course provides an in-depth introduction to object-oriented programming (OOP) principles and their implementation using the Java programming language. This course covers the different concepts of object-oriented programming such as classes, objects, inheritance, polymorphism, abstraction, encapsulation, exception handling, multithreading collections, generics, file handling and advanced concept of OOP. The course emphasizes both theoretical understanding and hands-on programming skills.

Course objectives:

The main objective of this course to provide students both the theoretical foundation and practical knowledge of programming using Java. After the completion of the course, students will be able to:

- understand OOP principles and implement them in Java.
- install java compiler and IDE ,compilation and run of java program
- design and develop Java applications using classes, objects, and interfaces.
- use exception handling, collections, and generics and file I/O in real-world programs.
- apply Advanced OOP concept (Design pattern, Lambda, Streams API) for efficient coding.
- build a small-scale Java project using OOP best practices.

Prerequisite:

Conceptual understating on programming logic and technique.

Course Contents

Unit-1 Introduction to Java and OOP concept **4Hrs.**

1.1 History, Feature or Buzzwords of Java

1.2 Java Architecture: JVM, JDK and JRE

1.3 Procedural-oriented Vs. Object-oriented Programming

1.4 Setting up java environment and IDE in Local machine

1.5 Sample java programs

1.6 Compiling and running java program

1.7 Command-line arguments

1.8 Scanner class for input

1.9 Handling common errors

Unit-2 Basic of Java Programming **8Hrs.**

2.1 Writing comments and its type

2.2 Java token: keywords, identifier, literal, operators and separators

2.3 Data types: primitive and user-defined data type.

2.4 Variable declaration and assignment, expression

2.5 Control statements: selection statements, looping statement and jump statements.

2.6 Arrays: single dimension array, multi-dimensional array (Rectangular and Jagged)

2.7 Type conversion and casting

2.8 Garbage Collection

2.9 String: creation, concatenation, comparison, modification, changing case and searching

2.10 String Buffer Class

Unit-3 Class and Objects in Java **8 Hrs.**

3.1 Defining class, adding method to class, creating object and calling function/method

3.2 Abstraction and Encapsulation

3.3 Constrictors and its type (Default, Parameterized and copy)

3.4 'this' keyword

3.5 Static fields and methods

3.6 More onmethod: passing by value, by reference

3.7 Recursion

3.8 Nested and inner class

3.9 Variable length arguments

3.10 Package: Defining and importing package

Unit-4 Inheritance and Polymorphism **6 Hrs.**

4.1 Inheritance basics

4.2 Inheritance Type(Single-level, Multi-level, Multiple and Hierarchical)

4.3 'super' keyword

4.4 Polymorphism: Method overloading and method overriding

4.5 Object class

4.6 'final' keyword

4.7 Abstract class and methods

4.8 Access control (private, protected, default and public)

4.9 Interface: Defining, implementing and applying interface

Unit-5 Exception Handling and Multithreading **6 Hrs.**

5.1 Basic exceptions, proper use of exceptions

5.2 Exception hierarchy

5.3 Exception handling keywords: try, catch, throw, throws and finally

5.4 Java's built-in exceptions

5.5 User-defined exceptions

5.6 Multithreading basics

5.7 Thread class and Runnable interface

5.8 Thread priorities

5.9 Thread synchronization and inter-thread communication

Unit-6 File Handling in Java **6 Hrs.**

6.1 Console and File I/O

6.2 Reading and writing file using byte stream

6.3 Reading and writing file using character stream

6.4 Serialization and deserialization

6.5 RandomAccessFile class

Unit-7 Collections and Generics **6 Hrs.**

7.1 Wrapper class and associate methods

7.2 Java collection framework

7.2.1 List, Set, Map interface

7.2.2 ArrayList, LinkedList, HashSet, HashMap and TreeSet Class

7.3 Accessing collections: Iterator/comparator

7.4 Defining generic class and methods

7.5 Using wildcard arguments

7.6 Generic interface and generic hierarchy

7.7 Some generic restrictions

Unit-8 Advanced OOP concepts in Java **4 Hrs.**

8.1 Design pattern: singleton, factory, observer pattern

8.2 Lambda expression

8.3 Stream API: Introduction

8.4 Optional class

8.5 Method references

Laboratory Work **48 Hrs.**

Practical report contains: Theory, source code and output.

- To install Java SE and IDE (Notepad++, NetBeans or Eclispe) on local Machine.
- To run and compile java program and implement concept of command-line argument and scanner class.
- To implement java basic concepts: Token, control statements, arrays, type casting.
- To implement method provided by String and StringBuffer class.
- To define class, object and implement other object-oriented feature like Encapsulation, Inheritance, polymorphism and Abstraction.
- To handle built-in and user-defined exception and implement multithreading concepts.
- To create, write and read file using byte stream, character stream class and use concept of serialization, deserialization and RandomAccessFile.
- To implement java collection Framework and generic concepts.
- To implement advanced OOP concepts in java
- After end of course student must submit programming project on object-oriented concept of Java.

Required readings

Balagurusamy, E. (2024). *Programming with Java* (7th ed.). McGraw Hill.

Deitel, P. J., & Deitel, H. M. (2024). *Java how to program: An objects-natural approach* (12th ed.). Pearson.

Horstmann, C. S. (2024). *Core Java, Volume I: Fundamentals* (13th ed.). Addison-Wesley Professional.

Schildt, H., & Coward, D. (2024). *Java: The complete reference* (13th ed.). McGraw Hill.

MATHEMATICS II

COURSE CODE: BCA 154

NATURE OF COURSE: THEORY/ LAB **CREDIT HOURS** 3

YEAR/SEMESTER: I/I

Course Description:

The course covers the concepts of Limit and Continuity, Derivatives of Algebraic, Trigonometric, Exponential, and Logarithmic functions, and their applications. It also covers Integration and its Applications, Volume and Surface Integral of some selected functions, and Numerical Integration using Trapezoidal and Simpson's Rule. Ordinary and Partial Differential Equations, their examples and uses are also discussed in this course. Other contents of the course include Optimization Problems like Simplex Method for two variables, and Gauss-Seidel, Gauss Elimination, Matrix Inversion, Bisection, and Newton-Raphson Method for solving the linear equations.

Course Objectives:

The objective of this course is to enable students to

- understand the concept of limit and continuity of functions and their connection to the derivative,
- differentiate different types of functions, geometrical meaning of the derivative, and its applications to real-world problems,
- integrate functions, understand their meaning and applications, including the surface and volume integrals,
- solve ordinary and partial differential equation and their connections to real-world problems,
- calculate the optimization problems, describe and interpret the graphical and numerical solutions.

Course contents:

Unit 1: Limit and Continuity **7 Hrs.**

1.1 Definition of limit including epsilon-delta condition, right and left hand limit, and its interpretation

1.2 Algebraic properties of limit

1.3 Definition and conditions of continuity and discontinuity

1.4 Continuity of algebraic, Trigonometric, and exponential functions, examples, and counterexamples.

Unit 2: Derivatives **7 Hrs.**

2.1 Definition and geometrical meaning of derivatives,

2.2 First principle (or by definition) method to differentiate algebraic, trigonometric, exponential, and logarithmic functions,

2.3 Rules of derivatives (sum, product, power, chain, and quotient rule),

2.4 Derivatives of inverse circular, hyperbolic functions and implicit functions,

2.5 Higher order derivatives,

2.6 Relation between derivative and continuity

2.7 Definition and examples of partial derivatives

Unit 3: Applications of Derivatives **8 Hrs.**

3.1 Increasing and decreasing functions,

3.2 Equation of tangents and normals using first derivatives,

3.3 L' Hospital's rule,

3.4 Angle between two lines,

3.5 Maxima and minima, absolute maxima and minima, concavity, stationary points and points of inflection,

3.6 Statement and geometrical interpretation of Rolle's theorem, Cauchy Mean-value theorem and Generalized Mean-value theorem,

3.7 Taylor's theorem, Maclaurin theorem (without proof) and its use in expansion of some simple functions,

3.8 Applications of derivatives in Economics,

3.9 Rate measures

Unit 4: Anti-derivative and its Applications **8 Hrs.**

4.1 Definition and geometrical meaning of integration,

4.2 Basic integration formulas for algebraic, trigonometric, exponential, and logarithmic functions,

4.3 Trigonometric substitution for basic functions, Integration by parts (Product rule for integration),

4.4 Partial fractions,

4.5 Improper integral,

4.6 Definite integral in terms of Riemann sum, and fundamental theorem of integral calculus

4.7 Applications of definite integral (Area under curve, area between curves, Quadrature & rectification)

4.8 Surface and volume integrals,

4.9 Trapezoidal and Simpson's Rule for numerical integration

Unit 5: Differential Equations **8 Hrs.**

5.1 Definition, order, and degree of differential equations,

5.2 Differential equations of first order and first degree,

5.3 Variables separable, homogeneous, exact, and linear differential equations,

5.4 Reducible to linear form,

5.5 Partial differential equations with some basic examples

Unit 6: Computational Methods **10 Hrs.**

6.1 Linear programming problems,

6.2 Linear inequalities in two variables and their graphical solutions,

6.3 Simplex Method (up to 3 variables), Duality problems

6.4 Matrix inversion method,

6.5 Gauss Elimination, Gauss-Seidel method,

6.6 Bisection method and Newton-Raphson Method for non-linear equations

Laboratory Works

Students are expected to use Python, MATLAB, or Mathematica to solve the numerical problems of each units and compare the solution with that of pen and paper.

Examination Scheme				
Internal Assessment		External Assessment		
Theory	Practical	Theory	Practical	
20	20 (3hrs)	60 (3 Hrs)	---	100

Required reading:

Bajracharya, B. C. (2025). *Basic Mathematic,* Sukunda Pustak Bhawan,

Budnick, F. S. (2017). *Applied Mathematics for Business Economics and Social Sciences.* McGraw-Hill,

Chand, K. B., Sapkota, B. P. (2022). *Principles of Mathematical Analysis.* Pinnacle Publication.

Lay, D. C. (2003). *Linear algebra and its applications.* Pearson Education India.

Stewart, J., Clegg, D., Watson, S. (2020). *Calculus.* Cengage|webassign.

Thomas G. B., Finney, R. L. (1995). *Calculus and Analytical Geometry.* Narosa Publishing House. (Textbook)

UX/UI DESIGN

COURSE CODE: BCA 155
YEAR/SEMESTER: I/II **CREDIT HOURS: 3**
WORKLOAD: 6 Hrs A WEEK (THEORY: 3 Hrs, PRACTICAL: 3 Hrs)

Course description:

This course covers different design principles, concepts and techniques of UX and UI designs focusing on making products with user centric user interface which is easy to use, efficient and user-friendly. Students are introduced with the basic concepts, design techniques and evaluation of UX and UI. It also includes latest concepts used by the designers to design the UX/UI and also trends in the UX/UI design.

Course objectives:

The primary objective of this course is to introduce different principles, techniques and aspects of UX/UI design. Students should be able to:

- Develop a deep understanding of concepts of UX and UI designs
- Conduct user research and analysis for user interaction designs
- Understand the principles and techniques of interaction design
- Implement user friendly interaction styles
- Create UI designs using proper user interface components and controls
- Create wireframes, prototypes, and high-fidelity designs using industry-standard software
- Perform design evaluations on the visual designs and interactions
- Implement advanced techniques for interface design including voice user interface and NLP based interface

Course contents

Unit 1: Introduction **4 Hrs**

1.1 Fundamentals of UX and UI
1.2 UX vs UI
1.3 Tasks of UX designer and UI designer
1.4 UX principles: Usability, Accessibility, Simplicity
1.5 Core discipline of UX: User research, content strategy, Information Architecture, Interaction design, Visual Design, usability evaluation
1.6 User interfaces: CLI, GUI, VUI, Menu-driven, NLP based
1.7 Properties of good UX/UI design
1.8 UX/UI tools: Figma, Adobe XD, Sketch

Unit 2: User interaction design **4 Hrs**

2.1 UX design process and user center design, Mindmap
2.2 UX research: Conducting user research: Interviews, Surveys, Competitive analysis, creating user personas, user journey mapping

2.3 Ideation techniques: using Mood boards, Brainstorming and sketching

Unit 3: User Interface design **6 Hrs**

2.1 Graphical and web user interfaces
2.2 Interaction styles: Command line, Menu selection, Form fill in, Direct manipulation, Anthropomorphic
2.3 Principles of UI design
2.4 Graphical user interface
2.5 UI design process
2.6 Human considerations in interface and screen design
2.7 Technological considerations in interface design

Unit 4: UI components **12 Hrs**

3.1 System menus and functions of menus
3.2 Formatting of menus: Consistency, display, presentation, organization, complexity, item arrangement, ordering, grouping
3.3 Types of menus: Menu bar, pull down menu, cascading menu, popup menu, tear off menu, iconic menu
3.4 Selection of Windows and its components, window presentation styles: Tiled windows, overlapping windows, cascading windows
3.5 Types of windows: Primary, secondary windows, dialog boxes
3.6 Screen based controls: Operable controls (Buttons, toolbars), Text entry/Read-only controls (single and multiple line textboxes,) selection controls (radio buttons, checkboxes, palettes, list boxes, list view controls, drop down/popup list boxes
3.7 Other operable controls: slider, tabs, date picker, tree view, scroll bars
3.8 Selecting the proper controls
3.9 Creating meaningful graphics, icons and images

Unit 5: UI Design considerations **6 Hrs**

5.1 Page layout, Color scheme and font selection, typography, screen size and responsive designs, interactive element
5.2 Visual hierarchy principles: Alignment, Color, Contrast, Proximity, Size, Texture, Time
5.3 Navigation: Global navigation, utility navigation, Associative and Inline Navigation
5.4 Navigational models: Hub and spoke, fully connected, multilevel or tree, stepwise navigation, Pyramid navigation, flat navigation

Unit 6: Wireframing and prototyping **6 Hrs**

6.1 Wireframes and mock-ups
6.2 Prototyping: Low fidelity and high fidelity prototyping, interactive prototyping
6.3 UX storyboarding, mockups
6.4 Software prototyping
6.5 Transition and animation to prototypes
6.6 Creating a simple clickable prototype

Unit 7: Design evaluations **6 Hrs.**

7.1 Formative and summative evaluation

7.2 Usability testing: Moderated vs Unmoderated
7.3 Analyzing test results and gathering insights
7.4 Evaluation through expert analysis and user participation, iterative evaluation and evaluation paradigms
7.5 DECIDE evaluation framework, heuristic evaluation
7.6 Task analysis and performance metrics

Unit 8: Advanced techniques: VUI and NLP based UI **4 Hrs.**

8.1 Command and control vs Conversational UI
8.2 Personas, Avatars, Actors and Video games
8.3 Speech recognition technology and Dialog management
8.4 Designing for Wearable Devices

Laboratory Works: **48 Hrs.**

Laboratory work should be carried out by using some UX/UI tools such as Figma, Adobe XD or Sketch and need to cover the following tasks:

- Use the basic tools and features for visual designs
- Use proper page layout, font selection, screen size and responsive designs
- Implement color theory and typography in interface designs
- Create designs with proper grids and layouts
- Integrating interactive elements to the designs
- Create design with different types of windows and operable controls
- Use different navigational styles in the designs
- Construct wireframe and prototypes for the designs
- Create the storyboards for interaction designs
- Apply different effects, styles and animation for designs

At the end of the semester, students need to submit a project that should include the basic interface and interaction designs for a chosen topic individually and perform the evaluations on the designs.

Required readings:

Galitz, W. O. (2007). *The essential guide to user interface design* (3rd ed.). Wiley Publishing.

Gothelf, J., & Seiden, J. (2013). *Learn UX: Applying lean principles to improve user experience* (1st ed.). O'Reilly Media.

Green, T., & Brandon, K. (2024). *UX design with Figma: User-centered interface design and prototyping with Figma*. Apress.

Tidwell, J., Brewer, C., & Valencia, A. (2020). *Designing interfaces: Patterns for effective interaction design* (3rd ed.). O'Reilly Media.

Wood, D. (2014). *Interface design: An introduction to visual communication in UI design*. Bloomsbury Publishing.

PRINCIPLE OF MANAGEMENT

COURSE CODE: BCA 156
YEAR SEMESTER: I/II
WORKLOAD: 1 Hrs. /WEEK (THEORY: 1 Hrs. **CREDIT HOURS: 1**

Course description:
This course introduces the fundamental principles of management by integrating classical concepts with modern practices and emerging issues in management. It introduces students to the core functions of management - planning, organizing, decision-making and leading - while highlighting social dimensions of organizational behavior. Practical emphasis is placed on understanding the uses of IT in management.

Course objectives:
The main objective of this course is to provide students with a conceptual foundation and practical understanding of management principles, along with their basic applications. After the completion of this course, students will be able to:

- Understand the concepts, evolution, and functions of management,
- Apply management processes including planning, organizing and leading in organizational settings,
- Evaluate managerial decision-making using both traditional approaches and IT-supported methods,
- Examine the ethical and social responsibilities of organizations in modern contexts and
- Appreciate the growing role of IT in enhancing management efficiency.

Course content

Unit – 1 Introduction to Management **5 Hrs.**
1.1 Concept and meaning of management
1.2 Forms of business
1.3 Management process
1.4 Types of managers
1.5 Basic managerial roles
1.6 Managerial skills
1.7 Integrated management framework
1.8 Managing ethics and diversity
1.9 Social responsibilities and organizations
1.10 Role of IT in management

Unit - 2 Planning and Decision making **5 Hrs.**
2.1 Concept of planning
2.2 Planning process
2.3 Types of plan
2.4 Organizational goals
2.5 Organizational planning
2.6 SWOT analysis
2.7 Nature and process of decision-making
2.8 Use of IT in planning and decision-making

Unit – 3 Organizing **3 Hrs.**

3.1 Elements of organizing
3.2 Job design, job description and job specification
3.3 Authority distribution
3.4 Forms of organizational design

Unit – 4 Leading **3 Hrs.**

4.1 Nature of leadership
4.2 Generic approaches to leadership
4.3 Situational approaches to leadership
4.4 Emerging approaches to leadership
4.5 Managing team in the time of crisis
4.6 Leadership challenges in IT based organization

Required readings

Griffin, R. W. (2024). *Management (12th ed.)*. Cengage Learning.
Jones, R. G., & George, M. J. (2018). *Essentials of Contemporary Management*. McGraw-Hill Higher Education.
Mark, W. H., & Koontz, H. C. (2019). *Essentials of management*. McGraw-Hill Higher Education.

Teaching approach

- The course will be delivered in a seminar format where students take an active role.
- Each student (individually or in groups) will present assigned topics from the syllabus in class.
- Sessions will focus on discussion, critical thinking, and peer learning rather than traditional lectures.
- The instructor will act as a facilitator and moderator, guiding discussion and ensuring key concepts are well covered.

2. Student Responsibilities

- **Presentations:** Students must prepare and present course topics, integrating theory, case studies, and practical applications.
- **Active Participation:** Students are expected to engage in discussions, raise questions, and contribute insights during every class.
- **Term Paper:** Each student will submit a written term paper on a management topic (approved by the instructor) by the end of the semester.
- **Reflection Note:** A short reflective write-up summarizing personal learning and insights from the seminar experience is required.
- **Quizzes/Short Assignments:** Short quizzes may be conducted to reinforce learning.

3. Evaluation Scheme

The course does not include a final examination. Assessment will be continuous and based on:

- **Class Presentation – 30%**
- **Participation & Discussion – 20%**
- **Attendance – 10%**
- **Quizzes/Short Assignments – 10%**
- **Term Paper & Reflection Note – 30%**

4. Seminar Conduct Rules

- Presenters must share presentation slides or handouts in advance.
- All students should complete basic reading before each session to participate effectively.
- Respectful listening, constructive feedback, and academic integrity are expected at all times.
- Reflection notes should highlight key learning, challenges, and takeaways from the course.