

# **Rajarshi Janak University**

**Janakpur, Dhanusha**

**Nepal**

**Revised Curriculum of Bachelor in Computer Application**

**(Semester – I)**

**Under**

**Humanities and Law**



**Submitted by**

**Faculty of Humanities and Law  
RAJARSHIJANAK UNIVERSITY  
Janakpurdham, Nepal**

**2082**

## About Course

The 4-year Bachelor in Computer Application (BCA) program at Rajarshi Janak University (RJU) is a forward-looking, labor market-driven curriculum designed to equip students with the technical knowledge, analytical abilities, and practical skills required to excel in the rapidly evolving field of information technology. Tailored to meet the growing demand for skilled IT professionals, the program integrates modern computing concepts, software development practices, and real-world applications. It ensures that graduates are fully prepared to thrive in diverse sectors such as software development, database management, networking, cloud computing, and digital innovation contributing effectively to the digital transformation and socio-economic development of Nepal and beyond.

## Objectives

Some of the prime objectives of the curriculum designed for the Bachelor in Computer Application (BCA) are:

1. To build strong foundations in computing
2. To develop practical software development skills
3. To strengthen analytical and problem solving abilities
4. To Prepare students for advanced technologies
5. To foster interdisciplinary and business understanding
6. To promote practical learning through projects and internship

## Purpose of the course

The purpose of the revised curriculum of BCA at RJU is to provide a structured, industry aligned and future focused academic framework that prepares students to become competent IT professional capable of contributing meaningfully to Nepal's growing digital ecosystem and the global ICT sector.

Curriculum is designed to:

1. Align education with labor market needs
2. Develop competent and ethical ICT professionals
3. Integrate theoretical knowledge with practical application
4. Foster innovation and technological adaptability
5. Support national digital transformation
6. Strengthen research, creativity and project execution skills
7. Prepare students for higher education and professional growth

## Eligibility

The student who has passed +2, Intermediate, PCL or equivalent level with minimum of 40% marks or C grade (at least 1.61 GPA) from any stream shall be eligible to join the BCA course on the basis of marks obtained in entrance.

  
Vice-Chancellor

**Bachelor in Computer Application (BCA)  
(Humanities and Law)**

Courses	Credit hours
Computer Application	78(26 * 3)
Language courses	6(2*3)
Mathematics and statistics course	9(3 *3)
Social Sciences & Management courses	9(3*3)
Elective course	12(4*3)
Projects	12(2*3+6)
Internships	3
<b>Total Credit Hours</b>	<b>129</b>


Curriculum Structure

**First Year/First semester**

Sr. No.	Course Code	Course Title	Credit hours
1	BCACS 101	Computer Fundamental and Office Automation	3
2	BCAEN102	English for Computing and information Technology	3
3	BCAMT103	Mathematics	3
4	BCACS104	Programming Technique using C	3
5	BCACS105	Logic and Computer Design	3
		Total	15

**First Year/Second semester**

Sr.No	Course Code	Course Title	Credit hours
1	BCASO111	The art of social engagement and Technology	3
2	BCACS112	UI/UX using UML	3
3	BCAMT113	Mathematics – II	3
4	BCACS114	OOP in Java	3
5	BCACS115	Microprocessor and Computer Architecture	3
		Total	15

  
 Vice-Chancellor

### Second Year /Third semester

Sr.No	Course Code	Course Title	Credit hours
1	BCAMG201	Financial Accounting	3
2	BCACS202	Data Structures & Algorithm	3
3	BCACS203	Operating System	3
4	BCACS204	Discrete Structure	3
5	BCACS205	Web Technology - I	3
		Total	15

### Second Year/ Fourth semester

Sr.No	Course Code	Course Title	Credit hours
1	BCACS211	Advance Java programming	3
2	BCACS212	Database Management System	3
3	BCACS213	Numerical Method	3
4	BCACS214	System Analysis and Design	3
5	BCACS215	Web Technology - II	3
6	BCAPJ216	project - I	3
		Total	17

### Third year / Fifth semester

Sr.No	Course Code	Course Title	Credit hours
1	BCACS301	Data communication and Computer Networking	3
2	BCAST302	Probability and Statistics	3
3	BCACS303	Artificial Intelligence	3
4	BCACS204	Software Engineering	3
5	BCACS205	Programming in Python	3
		Total	15

### Third Year / Sixth semester

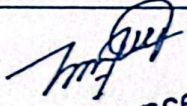
Sr.No	Course Code	Course Title	Credit hours
1	BCAMG311	Economics for Computer Application	3
2	BCACS312	Cloud Computing	3
3	BCACS313	Linux	3
4	BCACS314	Mobile Application Development	3
5	BCACS315	Computer Graphics and Multimedia	3
6	BCAPJ316	Project- II	3
		Total	17

### Fourth Year/Seven semester

Sr.No	Course Code	Course Title	Credit hours
1	BCACS401	Cyber Security	3
2	BCACS402	Software Project Management	3
3	BCANP403	व्यवहारिक लेखन तथा सम्पादन	3
4	BCAIN405	Internships	3
5		Elective -I	3
6		Elective -II	3
		Total	18

Elective subjects:

Sr.No	Course Code	Course Title
		Elective I
1	BCACS406	Information Audit
2	BCAMG407	Enterprise Resource Planning
3	BCACS408	Data Analysis & Visualization
4	BCACS409	Geographical Information System
5	BCACS4010	Data mining
		Elective II
6	BCACS4011	MIS and E-commerce
7	BCACS4012	Machine Learning
8	BCACS4013	Software Testing and Quality Assurance
9	BCACS4014	Business Intelligence
10	BCACS4015	Wireless Networking

  
Vice-Chancellor

### Fourth year/Eight semester

Sr.No	Course Code	Course Title	Credit hours
1	BCACS411	E-governance	3
2	BCAPJ412	Project - III	6
3		Elective-III	3
4		Elective -IV	3
	Total		15

### Elective Subjects:

Sr.No	Course Code	Course Title
Elective - III		
11	BCACS413	Database Administration
12	BCACS414	Network management and Administration
13	BCACS415	Advanced Database Management System
14	BCACS416	Distributed System
15	BCACS417	.Net Technology
Elective - IV		
16	BCACS418	Big data
17	BCACS419	Search Engine Optimization
18	BCAMG419	Digital Marketing
19	BCACS4110	Image Processing
20	BCACS4111	Generative AI

**Course Title:** Computer Fundamental and Office Automation  
**Course No:** BCACS101  
**Year/Semester:** I/I  
**Course Duration:** 48 Hours (Theory + Practical)

**Credit Hours:** 3  
**Full Marks:** 60+20+20  
**Pass marks:** 24+8+8

## Course Description

This course provides a foundational and comprehensive understanding of the core concepts, components, and applications of computer systems and Information Technology (IT). It blends essential theoretical knowledge with hands-on proficiency in desktop productivity software and an introduction to modern technological trends, preparing students for advanced studies in BCA.

## Course Objectives

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

- To understand the fundamental hardware and software architecture of a computer system.
- To demonstrate proficiency in using essential office automation tools for professional documentation, data analysis, and presentations.
- To grasp the principles of data representation, networking, and the Internet.
- To identify and understand recent trends and emerging technologies like Cloud Computing and Artificial Intelligence.

## UNIT I – Overview of Computers

[9 hrs]

- 1.1 Introduction to computer
- 1.2. Characteristics and applications
- 1.3. Evolution of AI & quantum computing
- 1.4. Generations of computers
- 1.5. Microprocessors
- 1.6. AI chips
- 1.7. Cloud systems
- 1.8. Block diagram and Modern architecture:
  - 1.8.1 CPU, ALU, Control Unit
  - 1.8.2 Memory Hierarchy
  - 1.8.3 I/O subsystem
  - 1.8.4 Embedded systems and digital devices (smartphones, IoT)
  - 1.8.5 Capabilities and limitations in the AI era.

### Lab work

- Identify components of modern computers.
- Explore the architecture and functionality of embedded devices.

## UNIT 2 Storage Technologies

[9 hrs]

### 2.1 Introduction to Storage and its types:

2.1.1 Primary, Secondary, Tertiary,

### 2.2 Cloud storage,

2.3 Primary memory: Registers, Cache memory, RAM, ROM, cache, flash, virtual memory,

2.4. Secondary storage: Magnetic Tapes, Magnetic Disks, hard disks, SSD, Floppy disk, Optical Disks: CD, DVD, pen drives.

### 2.5. Introduction to virtual Memory

2.5 Cloud storage: Google Drive, OneDrive, AWS

2.6 System Bus: Introduction to bus and its type

2.7 Storage interfaces: NVMe, SATA, USB-C, Green storage and sustainable practices.

### Lab work

- Hands-on with storage devices and cloud storage services.
- Comparison of storage performance (SSD vs HDD, cloud vs local).

## UNIT 3 – Input/Output & Emerging Devices

[9 hrs]

### 3.1 Role of I/O in computing

### 3.2 Introduction to Input Devices

3.2.1 Keyboard

3.2.2 Pointing Devices

3.2.3 Scanners and its types,

3.2.4 Voice Recognition Systems,

3.2.5 Vision Input System,

3.2.6 Touch Screen

3.2.7 Biometric

3.2.8 Gesture Recognition,

3.2.9 IoT sensors

### 3.3 Introduction to Output Devices

3.3.1 Monitors and its types

3.3.2 Introduction to Printers

3.3.2.1 Impact Printers

3.3.2.2 Non-Impact Printers

3.3.2.3 3D Printers

3.3.3 Introduction to Plotters

3.3.4 Sound cards

3.3.5 Speakers,

3.3.6 AR/VR

3.3.7 Smart projectors

3.3.8 Assistive & accessibility technologies

3.3.9 Multimedia peripherals: webcams, microphones, speakers,

3.3.10 Connectivity trends: Bluetooth 5, Wi-Fi 6, NFC, USB 4.

#### Lab Work

- Hands-on practice with I/O devices and multimedia peripherals.
- Explore connectivity setups using modern interfaces.

#### UNIT 4 – Software and E-Waste

[6 hrs]

4.1 Introduction to Software and its needs

4.2 Types of Software

4.2.1 System Software

4.2.2 Operating System

4.2.3 Device driver

4.2.4 Firmware

4.2.5 Utility Programs

4.3 Language processor

4.3.1 Assembler

4.3.2 Interpreter

4.3.3 Compiler

4.4 Application Software and its types, Open source software, Role and function of operating system, Types of Operating Systems, Structure of Operating System, Definition, types, sources, and composition of E-waste, Environmental and health impacts, E-waste management: collection, segregation, dismantling, recycling, disposal, Roles of users, organizations, and recyclers, Case studies of successful e-waste management.

#### Lab Work

-Research and report on sustainable technology practices.

RJU  
Vice-Chancellor

## **UNIT V – Networking, Internet, Security & Office Automation [9 hrs]**

Digital communication systems, transmission media, data speeds, Network types: LAN, WAN, MAN, PAN, VPN; network topologies; devices (hub, switch, router, bridge), Internet fundamentals: web, email, browsers, search engines, Computer security: viruses, spyware, hacking; firewalls, access control

Office automation: definition, importance, components, Productivity software overview: word processing, spreadsheets, presentation, open-source vs proprietary.

### **Lab Work**

-Practice office tasks in MS Word, Excel, and PowerPoint.

-Practical tasks related to Excel(at least )

1. Create a student marksheet workbook: calculate totals, averages, and assign grades using IF function.
2. Create a sales report with charts and PivotTables for monthly analysis.
3. Prepare a salary statement workbook using formulas, formatting, and conditional formatting.
4. Apply data validation and sorting/filtering on a product inventory dataset.
5. Automate invoice generation using VLOOKUP or a simple macro.

## **UNIT VI – Number Systems & Logic Gates [6 hrs]**

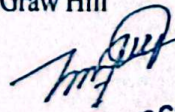
Number systems: binary, octal, decimal, hexadecimal; conversions, Signed number representation: 1's complement, 2's complement, Arithmetic operations in digital computers  
Boolean algebra, Boolean operators, logic gates: AND, OR, NOT, NAND, NOR, XOR

### **Lab Work**

-Simulate basic logic circuits using software or hardware kits.

### **References/Textbooks**

- P. K. Sinha & Priti Sinha, Computer Fundamentals, BPB Publications
- Peter Norton, Introduction to Computers, McGraw Hill
- R.K. Taxali, PC Software for Windows 10 Office 2019, Tata McGraw Hill
- Satish Jain, Office Automation, BPB Publications
- S. R. Pandey & R. K. Gupta, E-Waste Management in India: Challenges and Opportunities, Springer
- E. Balagurusamy, Fundamentals of Computers, Tata McGraw Hill

  
Vice-Chancellor

**Course Title:** English for Computing and Information Technology

**Course no:** BCAEN102

**Year / Semester:** I/I

**Course Duration:** [48 hrs.] Theory

**Credit hours:** 3

**Full Marks:** 60+40

**Pass Marks:** 24+16

**Objective:**

This course aims to enhance students' communicative competence in English within the context of computing and information technology. It focuses on developing reading, writing, listening, and speaking skills using computer-related topics. Students will learn to use technical vocabulary, comprehend IT texts, and communicate effectively in academic and professional settings related to computing. Basically,

- Understand and explain key concepts in computer science and information technology.
- Apply appropriate grammatical and lexical structures in discussing technical topics.
- Read and interpret technical documents, manuals, and articles.
- Write coherent reports, comparisons, and summaries about IT-related issues.
- Use spoken English effectively in presentations, discussions, and teamwork tasks within a computing context.

**Course Contents:**

<b>Unit 1 Personal computing</b> 4Hrs. 1. Overview of personal computers (PC) and their components <b>Language focus A</b> Key Vocabulary and Terminology Communication Skills: Introduction and description	<b>Unit 7 Computers in the Workplace</b> 4Hrs Office Information Systems Role of Computers in the Office <b>Language focus</b> Passive Voice Indirect narratives
<b>Unit 2 Online Services and Data Communication</b> 5Hrs. <b>Language focus</b> Vocabulary Tense use Functional English for online communication	<b>Unit 8 Computers in E-learning</b> 4 hrs • E-Learning and Digital Education • Technology for Teaching and Collaboration <b>Language focus</b> Connectives Signals
<b>Unit 3 Programming and Computer Languages</b> 5Hrs. Online services Data transmission <b>Language focus</b> Past tense Conditional Comparatives	<b>Unit 9 Robotics and Robot Characteristics</b> 5 hours Intro / components and characteristics <b>Language focus</b> WH Questions World formation prefixes Compound Nouns

<p><b>Unit 4 Software Applications and Packages</b> 5Hrs.          Programming and languages  <b>Language focus</b>          Future Tense          Comparative          Contrastive structures</p>	<p><b>Unit 10 Online Employment</b> 4 hours          Job Platforms and Tools          Skills for Digital Workplaces  <b>Language focus</b>          Cause and effects    <i>Explanations and definitions</i></p>
<p><b>Unit 5 Networking and Connectivity</b> 4Hrs.    <i>Fundamentals of Computer Networks</i>    <b>Language Focus</b>    <i>Expressions for explaining purposes and functions</i>            Linking steps in a process</p>	<p><b>Unit 11: Artificial Intelligence (AI), Machine Translation (MT) and MOOC</b> 4 hours    <b>Language Focus</b>    <i>Common expressions for predictions: Predicting future trends in technology</i></p>
<p><b>Unit 6 Computer viruses and Computer security</b> 4 hours    <i>Security Measures</i>  <i>Cyber security Awareness</i>  <b>Language focus</b>          Conjunctions          Punctuation</p>	

**Teaching Methodology**

Lectures, discussions, task-based learning, multimedia use, and practical assignments will be applied to integrate language skills with technical contexts.

*[Signature]*  
 Vice-Chancellor

## Texts Books

Boeckner, K., P.C. (1993). *Oxford English for Computing*. Spain : Oxford University Press.

## Reference Books

1. Glendinning, E. H., & McEwan, J. (2003). *Oxford English for Computing*. Oxford University Press.
2. Hutchinson, T., & Waters, A. (1987). *English for Specific Purposes: A Learning-Centered Approach*. Cambridge University Press.
3. Nelson, M. (2000). *A Corpus-Based Study of Business English and ESP*. Cambridge University Press.
4. Shelly, G. B., & Vermaat, M. E. (2017). *Discovering Computers*. Cengage Learning.
5. Flowerdew, J., & Peacock, M. (2001). *Research Perspectives on English for Academic Purposes*. Cambridge University Press.
6. Online resources: BBC Learning English, Coursera (English for IT Professionals)
7. *Oxford English for Information Technology*, 2nd Edition (by Eric H. Glendinning & John McEwan)
8. *English for Information Technology: Course Book* (by Maja Olejniczak)



**Course Title: Mathematics – I**

**Course no: BCAMT103**

**Year/Semester: I/I**

**Duration of course: 48 hrs. (Theory + Practical)**

**Credit hours: 3**

**Full Marks: 60+20+20**

**Pass Marks: 24+08+08**

## Course Objectives:

The objective of this course is to impart fundamental mathematical background of real and complex number, relations and functions, theory of matrices and determinants, progressions, permutations and combinations, and coordinate geometry develop skills in students for pursuing computer application courses.

## Course Content

### UNIT 1: REAL AND COMPLEX NUMBER SYSTEMS

[9 HRS.]

#### 1.1 Set Theory

- 1.1.1 Definition of set
- 1.1.2 Element
- 1.1.3 Representation of set
- 1.1.4 Relation between two set
- 1.1.5 Operation on two set
  - 1.1.5.1 Union
  - 1.1.5.2 Intersection
  - 1.1.5.3 Difference
  - 1.1.5.4 Complement

#### 1.2 Concept of Real Numbers

- 1.2.1 Review of different real number
  - 1.2.1.1 Natural Numbers
  - 1.2.1.2 Whole Numbers,
  - 1.2.1.3 Integers
  - 1.2.1.4 Rational and Irrational numbers
- 1.2.2 Inequality order
- 1.2.3 Intervals
- 1.2.4 Absolute value

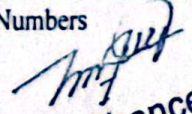
#### 1.3 Complex Numbers

- 1.3.1 Imaginary unit
- 1.3.2 Notations and geometrical representation
- 1.3.3 Algebra of complex numbers
- 1.3.4 Conjugate of complex number and properties

#### Practical

LAB 1: Check Type of Number (Natural, Whole, Integer, Rational, Irrational)

LAB 2: Addition, Subtraction, Multiplication of Complex Numbers

  
Vice-Chancellor

## Unit 2: RELATIONS AND FUNCTIONS

[6 Hrs.]

### 2.1 Relations

- 2.1.1 Introduction
- 2.1.2 Ordered pairs
- 2.1.3 Cartesian product
- 2.1.4 Relations and its domain and range
- 2.1.5 Properties of Relations
  - 2.1.5.1 Reflexivity
  - 2.1.5.2 Symmetry
  - 2.1.5.3 Anti-Symmetry
  - 2.1.5.4 Transitivity
  - 2.1.5.5 Equivalence relations

### 2.2 partial order, relation and Lattice

### 2.3 Introduction of Function

### 2.4 Function, its domain and range

- 2.4.1 Injective
- 2.4.2 surjective
- 2.4.3 Bijective Functions
- 2.4.4 Composite and Inverse Functions
- 2.4.5 Algebraic function
- 2.4.6 Trigonometric function
- 2.4.7 Exponential function
- 2.4.8 Logarithmic function

### 2.5 Modulus

- 2.5.1 Floor and ceiling Functions and their graphs

#### Practical

LAB 3: Generate Ordered Pairs and Cartesian product of Two Sets

LAB 4: Check Whether Relation is a Partial Order & Lattice

## Unit 3: Quadratic Equations

Est'd: 2017

[7 hrs.]

- 3.1 Introduction of Polynomial equation
- 3.2 Roots of polynomial equations
- 3.3 Quadratic Equation
- 3.4 Nature of the roots of Quadratic Equation
  - 3.4.1 Relation between roots and Coefficients
  - 3.4.2 Formulation of Quadratic equations,
- 3.5 conditions for common roots

#### Practical

LAB 5: Evaluate a Polynomial  $f(x) = a_n x^n + \dots + a_1 x + a_0$  (Using Horner's Rule)

LAB 6: Check Conditions for Common Roots of Two Quadratic Equations

## Unit 4: Matrix determinant and system of linear equation

[8 hrs.]

- 4.1 Introduction
- 4.2 System of Linear Equations
- 4.3 Solution of system of linear equation by
  - 4.3.1 Inverse Matrix method,
  - 4.3.2 Cramer's Rule,

  
Vice-Chancellor

**Practical**

LAB 7: Introduction – Input and Display a Matrix

LAB 8: Solve n×n System by Gaussian Elimination (General Program)

**Unit 5: Sequence and Series**

[9 hrs.]

5.1 Concept of sequences

5.1.1 Classical and Modern definitions of sequence with examples

5.1.2 Arithmetic

5.1.3 Geometric

5.1.4 Harmonic Progressions and their properties

5.1.4.1 Arithmetic, Geometric, Harmonic Means and relations among them

5.2 Infinite Series

5.2.1 Partial Sums and Remainders

5.2.2 Conditions for an infinite series to have sum

5.2.3 Sum of an infinite geometric series

**Practical**

LAB 9: Generate a sequence using classical and modern definition

LAB 10: Find remainder term of an infinite series

**UNIT 6: ANALYTICAL GEOMETRY**

[9 hrs.]

6.1 Conic Sections

6.1.1 Parts straight line

6.1.2 Circle

6.1.3 Ellipse

6.1.4 Parabola and Hyperbola as locus of points

6.1.5 Different conics

6.1.5.1 Plane section of right circular cone

6.2 Different forms of Equations of conics and their graphs

6.3 Determination of parts of conics from their equations and vice versa

6.4 Introduction of Coordinates in Space

6.4.1 Distance and section formula

6.4.2 Direction cosines and ratios

6.5 Angle between two lines

**Practical**

LAB 11: Check if a point lies inside, outside or on a circle

Circle:  $(x-h)^2+(y-k)^2=r^2$   $(x-h)^2+(y-k)^2=r^2$   $(x-h)^2+(y-k)^2=r^2$

LAB 12: Section formula (Internal division in ratio m:n)

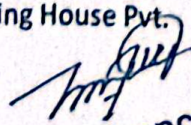
[Note: Instructor can use MATLAB to conduct the lab session]

**Reference / Textbook:**

Thomas, G.B. & Finney, R.L., "Calculus with Analytic Geometry", Addison-Wesley, 9<sup>th</sup> Edition.

Bajracharya, S.R., Shrestha, R.M. & Set al., "Basic Mathematics," Sukunda Pustak Bhawan, Nepal. Budnick, F.S., "Applied Mathematics for Business Economics and the Social Sciences", McGraw-Hill Ryerson Limited.

Mongga, G.S., "Mathematics for Management and Economics", Vikas Publishing House Pvt. Ltd., New Delh

  
Chancellor

Course Title: Programming Technique using C  
Course no: BCACS104  
Year / Semester: I/I  
Course Duration: Theory + Lab  
24+08+08

Credit hours: 3  
Full Marks: 60+20+20  
Pass Marks:

### Course Description

This course is designed to introduce students to the fundamental concepts of computer programming using the C language. It emphasizes problem-solving through structured programming, guiding students from algorithm design and flowchart creation to the development of complete C programs. The syllabus covers essential topics such as variables, data types, operators, control structures, functions, arrays, pointers, strings, structures, unions, and file handling, along with an introduction to graphics programming. Through a blend of theoretical instruction and hands-on lab exercises, students gain practical experience in writing, testing, and debugging C programs.

### Course Objectives:

1. Develop the ability in students to analyze real-life problems and design algorithmic solutions using structured programming concepts.
2. Enable student's ability to translate algorithms into efficient and correct C programs.
3. Introduce students to the syntax, semantics, and structure of the C programming language.
4. Make Familiarization to students with control structures, functions, arrays, pointers, and file handling in C.
5. Encourage the student to problem-solving skills through practical programming exercises and projects.
6. Prepare students to build simple applications that include input, processing, output, and data storage using C.

### Contents

Estd: 2017

#### Unit 1: Introduction to Programming Concept

[4 Hrs.]

- 1.1 Introduction of Programming Language
- 1.2 Assembler, Compiler and Interpreter
- 1.3 Syntax and Semantics
- 1.4 Programming Design Tools
  - 1.4.1 Algorithm
  - 1.4.2 Flow chart
  - 1.4.3 Pseudo codes
- 1.5 Features of good program


#### Lab Works

- Designing algorithm and draw flow chart for sequence, decision making and repetition concept of general programming using any CASE tool.

#### Unit 2: Introduction to C

[6 Hrs.]

- 2.1 History of C Program
- 2.2 Basic Structure of C Program
- 2.3 Character set, Token and Comments
- 2.4 Variables and Constants
- 2.5 Data Types
- 2.6 Type Conversion
- 2.7 Input and Output
- 2.8 Operators

  
Vice-Chancellor

- 2.8.1 Arithmetic
- 2.8.2 Relational
- 2.8.3 Logical
- 2.8.4 Increment/Decrement
- 2.8.5 Assignment
- 2.8.6 Bitwise
- 2.8.7 Ternary/ Conditional (:?)
- 2.8.8 Comma

**Lab Works**

- Developing basic structure of C program
- Declaring and assigning variables and constants.
- Applying input and output build in function
- Using arithmetic operators
- Giving demo of type conversion

**Unit 3: Control Structure**

[7 Hrs.]

- 1.1 Selective Structure
  - 1.1.1 If statement
  - 1.1.2 If-else statement
  - 1.1.3 Nested if-else statement
  - 1.1.4 Switch statement
  - 1.1.5 Conditional operator (:?)

1.2 Looping structure:

- 1.2.1 While Loop
- 1.2.2 Do-while loop,
- 1.2.3 For loop
- 1.2.4 Nested Loops

3.2 Loop interrupts (jump, break, exit)

**Practical Works**

- User if, if-else and switch statement
- Use while, do-while, for and nested loop concept.

**Unit 4: Function**

[6 Hrs.]

- 4.1 Function Concept
- 4.2 Function prototype, call and definition
- 4.3 Different ways of using function
- 4.4 Call by value, call by reference
- 4.5 Recursion

**Practical Works**

- Prototype, call and define function
- Pass the different parameter methods
- Use call by reference methods using function.
- Create a recursion function.


**Unit 5: Array, Pointer and String**

[7 Hrs.]

- 5.1 Concept of array
- 5.2 Array declare, access and initialization.
- 5.3 Multi-dimensional array
- 5.4 Concept of Pointer
- 5.5 Pointer address, dereference, declaration, assignment, initialization
- 5.6 Pointer Arithmetic
- 5.7 Array and Pointer
- 5.8 String
- 5.9 String functions in C
- 5.10 Pointer and String

**Practical Works**

- Array declares, define, initialize.

  
18 Vice-Chancellor

- Creating a single or multi-dimensional array.
- Using pointer and demo for arithmetic function.
- Using different string function in program.

**Unit 6: Structure and Union**

[7 Hrs.]

- 6.1 Concept of Structure
- 6.2 Initializing, accessing member of structure
- 6.3 Array of structure
- 6.4 Pointer to structure
- 6.5 Union
- 6.6 Different between union and structure

**Practical Works**

- Creating structure data types with application of loop.
- Creating union data types.

**Unit 7: Memory Management**

[6 Hrs.]

- 7.1 Concept of Memory Management
- 7.2 Dynamic Memory Allocation (DMA)
- 7.3 Common Memory Management Issues
- 7.4 Pointer and Memory Relationship

**Practical works**

- Creating dynamic arrays using pointers
- Allocating memory for strings and structures dynamically
- Implementing simple memory management examples

**Unit 8: Input output and File Handling**

[5 Hrs.]

- 7.1 Concept of File handling
- 7.2 File Access methods
- 7.3 Functions of file handling: fopen(), fclose(), fflush(), freopen()
- 7.4 Formatted input out
- 7.5 Character input output
- 7.6 Direct input output
- 7.7 Random file access
- 7.8 Error handling
- 7.9 File operation

**Practical Works**

- Creating file handling application for open, read, write and appends.
- Handling the random access files.
- Applying the text formatting function.

**Micro Project**

Preparation of simple application using C with the feature of input, process, output and store in external file.

**TEXT BOOKS:**

- Kanetkar, Y. P. (2008). *Let us C* 8<sup>th</sup> Ed, New Delhi, BPB Publication (Unit 1 -8)
- Balagurusamy, E. (2007). *Programming in ANSI C*. New Delhi, India: Tata McGraw-Hill.

**References materials:**

- Raman, R. (1984). *Computer programming in C*, New Delhi, PHI. India
- Carlo Ghezzi, Mehdi Jazayeri, "Programming Language Concepts", John Wiley and Sons
- B.S. Gottfried(2001), *Schaum's Outline Series for Programming with C*, Second Edition, Tata McGraw Hill Publishing Company, New Delhi

**Course Title:** Logic and Computer Design  
**Course Code:** Credit Hours: 3 (Theory + Lab)  
**Course Duration:** 48hrs. [Theory + Lab]

**Credit hours:** 3  
**Full Marks:** 60+20+20  
**Pass Marks:** 24+8+8

### Course Description

This module introduces students to the core concepts of digital systems and guides through Boolean algebra, logic gates, and the design of combinational and sequential circuits. Study of this subject enable students about digital circuits operation, how they are built and how different components work together to solve specific problems. The module connects theory with real applications found in computing, communication and embedded devices. Students will also work on laboratory tasks that strengthen their design skills and help them apply the concepts in practical digital system design.

### Course Objectives

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

- To explain the principles of number systems, Boolean algebra, and digital circuit design.
- To design, analyze, and simplify combinational and sequential logic circuits.
- To implement and test digital systems using simulation tools.

### Course Contents:

#### Unit I: Introduction to Digital Systems

[6 hrs.]

- Digital signals and logic levels
- Number systems and base conversions
- Binary arithmetic
- Complements
- R's and (R-1)'s complements
- Subtractions using complements
- Binary codes (BCD, Gray, Excess-3, ASCII)
- BCD addition
- Error detection and correction codes, Parity

#### Practical:

- Binary arithmetic exercises using simulation tools
- Design and test code converters using Logisim

#### Unit II: Boolean algebra and Logic Gates

[6 hrs.]

- Introduction to Boolean algebra
- Fundamental Laws and theorems of Boolean algebra
- De-Morgan's Theorems,
- Basic Logic Gates, Universal Gates and Arithmetic Gates and their truth tables
- Logic Operation
- Realization of Boolean functions using basic gates and universal gates.

**Practical:**

- Logic gate verification on breadboard/kit and simulator
- Proof of De-Morgans Theorem

**Unit III: Simplification and Logic Optimization**

[7 hrs.]

- Canonical forms (SOP, POS)
- Karnaugh map simplifications (upto 4 variables)
- Don't-care conditions
- Analysis of SOP and POS Expression
- Conversion SOP to POS and Vice-versa
- NAND and NOR implementation.

**Practical:**

- Simplify Boolean expressions using K-map
- Implement simplified circuits on simulator/Kit

**Unit IV: Combinational Logic Design**

[12 hrs]

- Design methodology (truth table to circuit)
- Half Adders, Full Adder, Half subtractors, Full Subtractors, Magnitude Comparators, Multiplexers, De-Multiplexers, Encoders, Decoders
- Binary multipliers and dividers
- Binary Parallel adder-Subtractors
- Programmable Logic Devices (ROM, PLA, PAL, FPGA overview)

**Practical:**

- Implement adder-subtractor
- Implement combinational circuits using Simulators

**Unit V**

**Sequential Logic Design**

[10 hrs]

- Latches and flip-flops (SR, JK, D, T)
- Excitation and characteristic tables
- Clocking, setup and hold time
- Interconversion of Flip-Flops
- State diagrams and state transition tables
- Synchronous and asynchronous sequential circuits
- FSM design and analysis
- Mealy and Moore model

**Practical:**

- Implement flip-flops using Kit/ ICs

**Unit VI**

**Registers, Counters, and Memory**

[7 hrs]

- Registers and shift registers
- Synchronous and asynchronous counters
- Johnson and ring counters
- Introduction to memory organization (RAM, ROM, EEPROM)

### Practical:

- Simulate Registers and synchronous counter

### Text and Reference Books

#### Text Book:

- M. Morris Mano & M.D. Ciletti – *Digital Design*, 5th Edition, Pearson
- Thomas L. Floyd – *Digital Fundamentals*, 9th Edition, Pearson

#### Additional References:

- Charles Roth & Larry Kinney – *Fundamentals of Logic Design*, 7th Edition

#### Online Resources:

- Logisim Evolution (Free Simulator)
- Xilinx Vivado Tutorials (FPGA learning)
- TinkerCAD Circuits (for remote simulation)

