Scheme for 4-Year Bachelor of Computer Applications Program

[Leading to BCA and BCA(Honours)/BCA(Honours) with Research) Degrees]

w.e.f.

Academic Session 2023-24

Semester	Discipline-Specific Courses – Major	Credits	Minor/Minor Vocational (VOC)	Credits	Multidisciplinary Courses (MDC)	Ability Enhancement Courses (AEC)	Skill Enhancement Courses/Internship/ Dissertation	Value Added Courses	Total Credits
I	**23BCA401DS01** Mathematical Foundations of Computer Science	**4:0:0**	**23BCA401MI01** Computing Fundamentals & PC Software	**3:0:1**	**One Course @3 Credits** (To be chosen outof the common pool excluding the Multidisciplinary course prepared by the Department)	**One Course @2 Credits** (To be chosen out of common pool)	**23BCA401SE01** Internet & Web Design @ 3 Credits **1:0:2**	**VAC1 @ 2 Credits** (To be chosen out of the common pool)	**22**
	23BCA401DS02 Computer Fundamentals & Problem Solving using C	**3:0:1**							
II	**23BCA402DS01** Digital Logic Design	**4:0:0**	**23BCA402MI01** Programming in C & Data Structure	**3:0:1**	**One Course @3 Credits** (To be chosen out of the common pool excluding the Multidisciplinary course prepared by the Department)	**One course @ 2 Credits** (To be chosen out of common pool)	**23BCA402SE01** Python Programming @ 3 Credits **1:0:2**	**VAC2@ 2 Credits** (To be chosen out of the common pool)	**22**
	23BCA402DS02 Data and File Structures	**3:0:1**							
III	**24BCA403DS01** Operating Systems	**4:0:0**	**24BCA403MI01** Database Management System & SQL	**3:0:1**	**One Course @3 Credits** (To be chosen out of the common pool excluding the Multidisciplinary course prepared by the Department)	**One Course @ 2 Credits** (To be chosen out of common pool)	**24BCA403SE03** Android Programming @ 3 Credits **1:0:2**	-----	**24**
	24BCA403DS02 Object Oriented Programming using C++	**3:0:1**							
	24BCA403DS03 Database Management System	**3:0:1**							
IV	**24BCA404DS01** Computer System Architecture	**4:0:0**	**24BCA404MV01** Object Oriented Programming using	**1:0:3**	-----	**One Course @ 2 Credits**		**VAC3 @ 2 Credits** (To be chosen out of the common pool)	**24**

	24BCA404DS02 DataCommunication & Computer Networks	**4:0:0**	C++			(To be chosen out of common pool)			
	24BCA404DS03 Java Programming	**2:0:2**							
	24BCA404DS04 Computer Graphics	**3:0:1**							
V	**25BCA405DS01** Software Engineering	**4:0:0**	**25BCA405MV01** Web Development	**1:0:3**	-----	-----	**25BCA404IN01** Internship @ 4 Credits 0:0:4		**24**
	25BCA405DS02 Artificial Intelligence& Expert Systems	**4:0:0**							
	25BCA405DS03 Cloud Computing	**3:0:1**							
	25BCA405DS04 Web Application Development	**2:0:2**							
VI	**25BCA406DS01** Internet of Things (IoT)	**3:0:1**	**25BCA406MV01** Object Technology	**1:0:3**	-----	-----		---	**20**
	25BCA406DS02 Computer Security	**3:0:1**							
	25BCA406DS03 Data Warehousing & Mining	**3:0:1**							
	25BCA406DS04 Advanced Database Systems	**2:0:2**							
VII	**26BCA407DS01** Advance Software Engineering	**3:0:1**	**26BCA407MI01** Operating Systems & Shell Programming	**2:0:2**	-----	-----	-----	-----	**24**
	26BCA407DS02 Computational Intelligence	**3:0:1**							

	26BCA407DS03 Advance Java Programming	**2:0:2**							
	26BCA407DS04 Machine Learning using Python	**2:0:2**							
	26BCA407DS05 Theory of Computation	**4:0:0**							
VIII [4 Year BCA(Hons.)]	**26BCA408DS01** BlockchainTechnology and Applications	**3:0:1**	**26BCA408MI01** Computer Networks & Programming	**2:0:2**	-----	----- -----		-----	**24**
	26BCA408DS02 Artificial Neural Network & Deep Learning	**3:0:1**							
	26BCA408DS03 Data Analytics using R	**2:0:2**							
	26BCA408DS04 Mobile Application Development	**2:0:2**							
	26BCA408DS05 Augmented and Virtual Reality	**3:0:1**							
VIII [4 Year BCA(Hons.) with Research]	**26BCA408DS06** Research Methodology	**4:0:0**	**26BCA408MI01** Research Ethics	**4:0:0**			**26BCA408PD01** Research Project/ Dissertation @ 12 credits 0:0:12		**24**
	26BCA408DS07 Web and Social Media Analytics	**3:0:1**							

Syllabus for
4-Year Bachelor of Computer Applications Program
[Leading to BCA and BCA(Honours)/BCA(Honours) with Research Degrees]
w.e.f.
Academic Session 2023-24
First Semester

Name of the Program	**4-Year Bachelor of Computer Applications**	**Program Code**	**-----**
Name of the Course	**Mathematical Foundations of Computer Science**	**Course Code**	**23BCA401DS01**
Hours/Week	**4**	**Credits (L:T:P)**	**4:0:0**
Max. Marks.	**Theory: 100 (70+30)**	**Time of end term examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of 7 parts (short-answer type questions) covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory.

Course Objectives:

The objective of this course is to inculcate in students the fundamental mathematical background in Computer Science and to provide reasoning, learning and understanding skills. The students get exposure of various concepts of mathematics such as sets, relations, and functions, trigonometry, limit, continuity, derivatives, matrix and determinants.

Course Outcomes:

By the end of the course the students will be able to:
CO1: Understand and solve the problems on set, relation and functions.
CO2 Understand the concepts of trigonometry.
CO3 Solve the problems on limit and continuity.
CO4 Understand the concepts of derivative and solve the problems on derivative.
CO5 Understand the concept of Matrix and Determinants.

Unit – I

Sets: Sets, Subsets, Equal Sets Universal Sets, Finite and Infinite Sets, Operation on Sets, Union, Intersection and Complements of Sets, Cartesian Product, Cardinality of Set, Practical applications of set theory.

Relations And Functions: Properties of Relations, Equivalence Relation, Partial Order Relation. Function: Domain and Range, Onto, Into and One to One Functions, Composite and Inverse Functions.

Unit – II

Trigonometry: Introduction, Measurement of angles, trigonometric functions, relation between trigonometric functions, signs of trigonometric functions, trigonometric functions of standard angles. Basic of inverse trigonometry.

Limits & Continuity: Limit at a Point, properties of limit, computation of limits of various types of functions, Continuity of a function at a point, Continuity over an interval.

Unit – III

Differentiation: Derivative of a function, Derivatives of sum, differences, product & quotient of functions, Derivatives of polynomial, trigonometric, exponential, logarithmic, inverse trigonometric and implicit functions, Logarithmic Differentiation, Chain rule and differentiation by substitution.

Unit – IV

Matrices: Definition, Types of Matrices, Addition, Subtraction, Scalar Multiplication and Multiplication of Matrices.

Determinants: Definition, Minors, Cofactors, Properties of Determinants, Applications of determinants in finding area of triangle, Adjoint of matrix, Inverse of matrix, solving a system of linear equations using matrix method.

Suggested Readings:

1. C.L.Liu: Elements of Discrete Mathematics, McGraw Hill.
2. Lipschutz, Seymour: Discrete Mathematics, Schaum's Series
3. Babu Ram: Discrete Mathematics, Vinayek Publishers, New Delhi.
4. Trembley, J.P & R. Manohar: Discrete Mathematical Structure with Application to Computer Science, TMH.

5. Kenneth H. Rosen: Discrete Mathematics and its applications, TMH.
6. Doerr Alan &Levasseur Kenneth: Applied Discrete Structures for Computer Science, Galgotia Pub. Pvt. Ltd.
7. Any other book covering the contents of the subject.

Note: Latest and additional good books may be suggested and added from time to time.

Name of the Program	**4-Year Bachelor of Computer Applications**	**Program Code**	**-----**
Name of the Course	**Computer Fundamentals & Problem Solving using C**	**Course Code**	**23BCA401DS02**
Hours/Week	**5 (3+2)**	**Credits (L:T:P)**	**3:0:1**
Max. Marks.	**Theory: 75 (50+25)** **Practical: 25 (20+5)**	**Time of end term examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of 5 parts (short-answer type questions) covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory.

Course Objectives:

This is first course in programming. The objective of this course is to inculcate knowledge about fundamental concepts of computer and logical thinking amongst the young minds and to teach the Programming Language C. However, the process of learning a computer language will also be emphasized. Emphasis is also on semantics and problem solving. Students will be able to develop logics which will help them to create programs, applications in C. By learning the basic programming constructs, they can easily switch over to any other language in future.

Course Outcomes:

By the end of the course the students will be able to:
CO1: Understand the concepts of computer and its applications in various foelds.
CO2: Understand the fundamental concepts of programming in C language.
CO3: Demonstrate an understanding of data types, control structures, functions, arrays, and pointers
CO4: Develop basic programming solutions using C language.
CO5: Apply basic programming concepts to solve practical problems.

Unit – I

Computer Fundamentals: Generations of Computers, Block Diagram along with its components, classification of computers, Applications of computers in various fields.Input/Output Devices, Memory: Concept of primary & secondary memory, Cache Memory, Secondary storage devices.

Overview of Networking & Operating System: Introduction to computer networking, Network types, Network topologies, Internet and its applications; Operating system and its functions.

Unit – II

Planning the Computer Program: Problem definition, Program design, Debugging, Types of errors in programming, Techniques of Problem Solving- Flowcharting, Algorithms

Overview of C: History of C, Importance of C, Elements of C: C character set, identifiers and keywords, Data types, Constants and Variables, Assignment statement, Symbolic constant, Structure of a C Program,printf(), scanf() Functions, Operators & Expression, type casting and conversion, operator hierarchy & associativity.

.

Unit – III

Decision making & Branching: Decision making with IF statement, IF-ELSE statement, Nested IF statement, ELSE-IF ladder, switch statement, goto statement.

Decision making & Looping: while, do-while and for loop, jumps in loops, break, continue statement, Nested loops

Unit – IV

Functions: Standard Mathematical functions, Input/output: Unformatted & formatted I/O function in C, Input functions output functions, string manipulation functions.User defined functions**:** Introduction/Definition, function prototype, Local and global variables, passing parameters, recursion.

Arrays & Pointers: Definition, types, initialization, processing an array, passing arrays to functions, Declaration and initialization of string, Input/output of string data, Introduction to pointers.

Suggested Readings:

1. Gottfried, Byron S.: Programming with C, Tata McGraw Hill
2. Gill Nasib Singh: Computing Fundamentals and Programming in C, Khanna Book Publishing Company(Private) Limited, NewDelhi.
3. Balagurusamy, E.: Programming in ANSI C, Tata McGraw-Hill
4. Jeri R. Hanly & Elliot P. Koffman: Problem Solving and Program Design in C, Addison Wesley.
5. Yashwant Kanetker: Let us C, BPB.
6. Rajaraman, V.: Computer Programming in C, PHI.
7. Yashwant Kanetker: Working with C, BPB.
8. Any other book covering the contents of the subject.

Note: Latest and additional good books may be suggested and added from time to time.

List of Programs

1. Write a C program to demonstrate the use of `printf()` and `scanf()` functions. Prompt the user to enter their name and age, then display the entered information.
2. Write a C program to define and use symbolic constants. Define symbolic constants for PI and the radius of a circle, then calculate and display the area of the circle.
3. Write a C program to demonstrate the use of various operators and expressions. Perform addition, subtraction, multiplication, and division on two numbers entered by the user and display the results.
4. Write a C program to implement decision making using if-else statements. Check if a number entered by the user is positive, negative, or zero and display the result.
5. Write a C program to implement a nested if statement.
6. Write a C program to demonstrate the use of switch statement. Simulate a simple calculator to perform addition, subtraction, multiplication, and division based on user input.
7. Write a C program to implement looping using a while loop.
8. Write a C program to implement looping using a do-while loop.
9. Write a C program to implement looping using for loop
10. Write a C program to demonstrate the use of break and continue statements. Print numbers from 1 to 10, but skip the number 5 using continue and stop the loop when it reaches 8 using break.
11. Write a C program to define a user defined function to calculate the factorial of a number, entered by the user and display the result.
12. Write a C program to demonstrate the use of arrays. Read 10 integers from the user, store them in an array to find the largest number.
13. Write a C program toread a string from the user and then display its length, and convert it to uppercase and lowercase.
14. Write a C program to demonstrate the use of pointers. Declare an integer variable, assign a value to it, and then use a pointer to display its value and address.
15. Write a C program to pass arrays to a function. Read 5 integers into an array, pass the array to a function, and return sum of the elements of the array.

Any other programs assigned by the teachers.

Name of the Program	**4-Year Bachelor of Computer Applications**	**Program Code**	**-----**
Name of the Course	**Computing Fundamentals and PC Software**	**Course Code**	**23BCA401MI01**
Hours/Week	**5 (3+2)**	**Credits (L:T:P)**	**3:0:1**
Max. Marks.	**Theory: 75 (50+25)** **Practical: 25 (20+5)**	**Time of end term examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of 5 parts (short-answer type questions) covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory.

Course Objectives:

The main objective is to introduce computer to all undergraduate students, regardless of their specialization. It will help them to pursue specialized programmes in the area of computer. The focus of the course is to introduce basic concepts of computer, computer memory, software, ICT and MS-office tools.

Course Outcomes:
By the End of course the students will be able to:
CO1: Understand the fundamental concepts of Computers & its applications.
CO2: Understand the basic concepts of memory, storage devices and operating system.
CO3.Understand the concept of ICT.
CO4: Get exposure of data processing tool.
CO5: Get exposure of data analysis and Presentation tool.

Unit – I

Introduction to Computers: Introduction, Definition, Characteristics of Computer, Evolution of Computer, Generations of Computer, Block diagram of computer systems, Classification of Computers, Applications of Computer, Capabilities, and Limitations of Computer, Input and Output devices, Computer Virus and Antivirus.

Unit – II

Memory: Concept of primary & secondary memory, RAM, ROM, types of RAM and ROM, Cache Memory.**Secondary storage devices:** Sequential & direct access devices viz. magnetic tape, magnetic disk, optical disks, Virtual memory.
Software & Operating System: Software & its types, Operating System & its functions.
Computer Network: Concept of Networking, Types of Networks, Network topology

Unit – III

ICT: Introduction, ICT tools and terminology. Basics of Internet, Intranet, E-mail, Audio and Video-conferencing. Digital initiatives- SWAYAM, National digital library.

Documentation Using MS-Word: Introduction to word processing, Toolbars, Creating & Editing Document, Formatting Document and printing document, Finding and replacing text, Format painter, Header and footer, Spelling and Grammar Tool, Page Formatting, Bookmark, Mail Merge, Macros, Tables and File Management.

Unit – IV

Electronic Spreadsheet using MS-Excel: Introduction to MS-Excel, Feature of MS-Excel, Creating & Editing Worksheet, Formatting and Essential Operations, Formulas and Functions, Charts, Cell referencing, Pivot table & Pivot Chart, Linking, Sorting, Filtering.

Presentation using MS-PowerPoint: Starting MS–Power Point, working with power point, Creating, Saving andPrinting a presentation, Working with Animation, adding a slide to presentation, navigating through a presentation, Slide-sorter, Slide-show, Editing slides, Working with Graphics and Multimedia in PowerPoint (Inserting Photo, Video & Sound).

Suggested Readings:

1. Gill Nasib Singh: Handbook of Computer Fundamentals, Khanna Book Publishing Company(Pvt.) Limited, New Delhi.
2. Balagurusamy E: Computing Fundamentals and C Programming, Tata McGraw Hill.
3. Norton, Peter: Introduction to Computer, McGraw-Hill
4. Leon, Alexis & Leon, Mathews: Introduction to Computers, Leon Tech World
5. Rajaraman, V.: Fundamentals of Computers, PHI
6. Gill, Nasib Singh: Essentials of Computer and Network Technology, Khanna Books Publishing Co.(P), New Delhi.
7. Russell A. Stultz: Learn Microsoft Office — BPB Publication
8. Any other book covering the contents of the subject.

Note: Latest and additional good books may be suggested and added from time to time.

List of Programs

1. Create a document in MS-Word with a cover page, table of contents, and three sections explaining the introduction to word processing, toolbars, and document creation & editing. Apply various formatting styles to each section.
2. Write a step-by-step guide on how to format a document in MS-Word, including changing font styles, paragraph alignment, adding headers and footers, and printing the document.
3. Create a letter in MS-Word using the Mail Merge feature to send personalized invitations to a list of recipients stored in an Excel sheet.
4. Prepare a tutorial document in MS-Word demonstrating the use of the Spelling and Grammar tool, Format Painter, and Bookmark features.
5. Create a table in MS-Word to manage a list of books. Include columns for title, author, and publication year. Apply table formatting options to enhance readability.
6. Create a budget spreadsheet in MS-Excel, including rows for income sources and expenses. Use formulas to calculate the total income, total expenses, and net balance.
7. Prepare an Excel sheet to record student marks for five subjects. Use cell referencing and formulas to calculate the total marks, percentage, and grade for each student.
8. Create a chart in MS-Excel to visualize the monthly sales data for a year. Include at least two different types of charts.
9. Develop a pivot table and pivot chart in MS-Excel to summarize sales data by region and product category. Demonstrate how to sort and filter data within the pivot table.
10. Demonstrate how data is linked between two Excel sheet by creating and managing links, and demonstrate the impact of changing data in the source sheet on the linked sheet.
11. Prepare a PowerPoint presentation on the basics of creating, saving, and printing a presentation. Include screenshots and step-by-step instructions.
12. Create a PowerPoint presentation to introduce a new product with slides depicting product features, benefits, pricing, and a conclusion. Add animations and transitions to make the presentation engaging.
13. Prepare a PowerPoint presentation on the basics of creating, saving, and printing a presentation. Include screenshots and step-by-step instructions.
14. Design a PowerPoint presentation to explain how to add and navigate through slides, including using slide-sorter and slide-show views.
15. Create a PowerPoint presentation demonstrating the use of graphics and multimedia. Insert photos, videos, and sounds to enhance the presentation content.
16. Develop a PowerPoint presentation on a topic of your choice, using all features learned, including animations, transitions, graphics, and multimedia. Ensure it is professional and visually appealing.

Any other programs assigned by the teachers.

Multidisciplinary Course (MDC) 1st Semester

[To be chosen by the students other than that of Bachelor Program in Computer Applications]

Name of the Program	**4-Year Bachelor of Computer Applications**	**Program Code**	**-----**
Name of the Course	**Foundations of Information Technology**	**Course Code**	**23BCAX01MD01**
Hours/Week	**3**	**Credits (L:T:P)**	**3:0:0**
Max. Marks.	**Theory: 75 (50+25)**	**Time of end term examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of 5 short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory.

Course Objectives:

The main objective is to introduce IT in a simple language to all undergraduate students, regardless of their specialization. It will help them to pursue specialized programs leading to technical and professional careers and certifications in the IT industry. The focus of the subject is on introducing skills relating to IT basics, computer applications, programming, interactive medias, Internet basics etc.

Course Outcomes:

By the end of the course the students will be able to:
CO1: Understand the fundamental concepts of Computers its applications & Understand various input and output devices.
CO2: Understand the concept of Memory and operating System.
CO3. Understand the concept of data communication, networking and internet.
CO4: Get exposure of memory and operating system.
CO5: Know about E-Mail and the concepts related to Business data processing.

Unit – I

Introduction: Historical evolution of computers,Classification of computers, Block Diagram along its components and characteristics, Usefulness of Computers. Human being Vs. Computer, Applications of computers in various fields.

Input/Output Devices: Keyboards, mouse, joysticks, trackballs, digitizer, voice-recognition, optical-recognition, scanners, terminals, point-of-sale terminals, machine-vision systems, Printer & its types

Unit – II

Memory & Mass Storage Devices: Characteristics of memory systems, types of memory, RAM, ROM, magnetic disks-floppy disk, hard-disk; optical disks; Magnetic tapes; Concepts of Virtual and Cache memory

Operating System: Functions, Measuring System Performance, Assemblers, Compilers and Interpreters. Batch Processing, Multiprogramming, Multi Tasking, Multiprocessing, Time Sharing, DOS, Windows, Unix/Linux.

Unit – III

Data Communication: Communication Process, Data Transmission speed, Communication Types (modes), Data Transmission Medias, Modem and its working, characteristics, Types of Networks, LAN topologies, Computer Protocols, Concepts relating to networking.

Internet: Introduction to Internet, WWW and Web Browsers; Applications of Internet; connecting to internet; What is ISP?; Search Engines; Understanding URL; Domain name; IP Address; Web page, Website and home page.

Unit – IV

Electronic Mail: Introduction, advantages and disadvantages, User Ids, Passwords, e-mail addresses, message components, message composition, Web Browsers and search engines.

Business Data Processing: Introduction, data storage hierarchy, Method of organizing data, File Types, File Organization, File Utilities.

Suggested Readings:

1. Gill Nasib Singh: Handbook of Computer Fundamentals, Khanna Book Publishing Company(Pvt.) Limited, New Delhi.
2. Donald Sanders: Computers Today, McGraw-Hill Publishers.
3. Davis: Introduction to Computers, McGraw-Hill Publishers.

4. V.Rajaraman: Fundamental of Computers, Prentice-Hall India Ltd.,NewDelhi.
5. R Bangia: Learning MS-Office2000, Khanna Book Pub.
6. Sanders: Teach yourself MS-Office, BPB Publications.
7. Bott: MS-Office, PHI.
8. Any other book covering the contents of the subject.

Note: Latest and additional good books may be suggested and added from time to time.

Name of the Program	**4-Year Bachelor of Computer Applications**	**Program Code**	**-----**
Name of the Course	**Internet & Web Design**	**Course Code**	**23BCA401SE01**
Hours/Week	**5 (1+4)**	**Credits (L:T:P)**	**1:0:2**
Max. Marks.	**Theory: 25 (20+5)** **Practical: 50 (35+15)**	**Time of end term examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of 4 short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory.

Course Objective:

The objective of this course is to acquire knowledge and Skills for creation of Web Sites. Also to acquire the knowledge regarding creation of Web applications using tools and techniques used in industry and how to design a basic web site using HTML & CSS to demonstrate responsive web design.

Course Outcomes:

By the end of the course the students will be able to:
CO1: Understand the fundamental concept of internet and search engine.
CO2: Understand the concept of Web-casting techniques.
CO3.Understand the concept of website planning.
CO4: Get exposure of HTML and handful exposure of HTMLtags.
CO5: Learn about CSS and its characteristics.

Unit – I

Introduction to Internet and World Wide Web:A brief Introduction to the Internet, Evolution of World Wide Web; Basic features; Web Browsers; Web Servers; Hypertext Transfer Protocol, URLs; Searching and Web-Casting Techniques; Search Engines and Search Tools, Domain Name System, Home Page, Web page and website.

Unit – II

Web Publishing: Hosting your Site; Internet Service Provider; Phases of Planning and designing your Web Site; Steps for developing your Site; Choosing the contents;

Web Development: Introduction to HTML; Hypertext and HTML; HTML Document Features; HTML command Tags; Headers; Text styles; Text Structuring; Text colors and Background; Formatting text.

Unit – III

List: Ordered and Unordered lists, Table Creation and Layouts. Images; Inserting Graphics; Frame Creation and Layouts; Creating Links; Working with Forms and Menus; Working with Radio Buttons and Check Boxes; Text Boxes; Page layouts.

Unit – IV

Cascading Style Sheets (CSS): Basic Concepts, Properties, Creation of Style Sheets. Common Tasks with CSS: Text, Fonts, Margins, Links, Tables, Colors. Marquee. Mouse Overs. Filters and Transitions. Adding Links. Adding Tables. Adding Forms. Adding Image and Sound. Use of CSS in HTML Documents, Linking and Embedding of CSS in HTML.

Suggested Readings:

1. Raj Kamal: Internet and Web Technologies, Tata McGraw-Hill.
2. Ramesh Bangia:Multimedia and Web Technology, Firewall Media.
3. Thomas A. Powell : Web Design: The Complete Reference , 4/e, Tata McGraw-Hill
4. Wendy Willard : HTML Beginners Guide, Tata McGraw-Hill.
5. Deitel and Goldberg : Internet and World Wide Web, How to Program, PHI.
6. Any other book covering the contents of the subject.

Note: Latest and additional good books may be suggested and added from time to time.

List of Programs

1. Write HTML code to display your education details in a tabular format.
2. Write HTML code to display your CV on a web page.
3. Write HTML code to create a Home page having three links: About Us, Our Services and Contact Us. Create separate web pages for the three links.
4. Write HTML code to create a login form. On submitting the form, the user should get navigated to a profile

page.

5. Write HTML code to create a Registration Form. On submitting the form, the user should be asked to login with new credentials.
6. Write HTML code to create your Institute website, Department Website and Tutorial website for specific subject.
7. Write HTML code to illustrate the usage by creating the following:
   - Ordered List • Unordered List • Definition List
8. Write HTML code to create a frameset having header, navigation and content sections.
9. Write HTML code to demonstrate the usage of inline CSS.
10. Write HTML code to demonstrate the usage of internal CSS.
11. Write HTML code to demonstrate the usage of external CSS.
12. Write HTML program to create a webpage to show different art forms of India, with appropriate title on the title bar. Use different heading tags for the headings, and list them using ordered list.
13. Write HTML program to create sections in the document using appropriate tags and apply different color as background to them. Use internal hyperlinks to move to different points within the page.
14. Write HTML program to insert a picture on the webpage, giving description for the picture in a paragraph. Use properties of height, width, hspace, vspace and align, with different values.
15. Write HTML Program, to create a profile of 2 pages, the First page containing the applicant's picture with personal details using unordered lists, and the second containing Educational details using tables. Use hyperlinks to move to the next page.

Any other programs assigned by the teachers.

Second Semester

Name of the Program	**4-Year Bachelor of Computer Applications**	**Program Code**	**-----**
Name of the Course	**Digital Logic Design**	**Course Code**	**23BCA402DS01**
Hours/Week	**4**	**Credits (L:T:P)**	**4:0:0**
Max. Marks.	**Theory: 100 (70+30)**	**Time of end term examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of 7 parts (short-answer type questions) covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory.

Course Objectives:

To acquire the basic knowledge of digital logic levels and application to understand digital electronics circuits. This course also prepares students to perform the analysis and design of various digital electronic circuits, design and analyze sequential and combinational logic circuits.

Course Outcomes:

By the end of the course the students will be able to:
CO1: Understand the concept of logic gates.
CO2: Understand and use of number system and their conversion.
CO3: Learn the concept of combinational circuit and sequential circuits.
CO4: Understand the concept of Computer Organization and instruction sets.
CO5: Explore concepts related to Memory Organization and Input Output Organization.

Unit – I

Digital Systems and Binary Numbers: Digital Systems: Digital Signals, Digital Waveforms, Digital Computers and Digital Integrated Circuits. **Number Systems:** Binary Number Systems, Octal and Hexadecimal Number System. Number Base Conversions. Complements, Signed Binary Numbers and Binary Codes, Error Detection and Correction codes.

Boolean Algebra and Logic Gates: Boolean Algebra: Axiomatic Definition, Theorems and Properties. Boolean Functions, Canonical Standard forms: SOP and POS forms. **Digital Logic Gates:** NOT, OR, AND, NOR, NAND, XOR and XNOR. Universal Gates and their implementation

Unit – II

Gate Level Minimization: Karnaugh Map (K-map) Method: Simplification: Algebra postulates and Canonical forms. Prime Implicants: Types, Determination and Selection of Prime implicants.

Don’t Care Conditions, NAND and NOR implementation.

Unit – III

Combinational Circuits: Introduction, Characteristics and Designing principles of Combinational circuits. Binary Adder: Half-Adder & Full-Adder, Subtractor: Half-Subtractor & Full-Subtractor, Parallel binary Adder/Subtractor, Binary Multiplier, Comparators, Multiplexers, De-multiplexers, Encoders and Decoders.

Unit – IV

Sequential Circuits: Characteristics of Sequential Circuits, Latches, **Flip-Flops:** Introduction, S-R Flip flop, J-K Flip Flop, D Flip flop, T Flip flop and Master Slave Flip flop.

Registers: Shift Registers, Applications of Registers. **Counters**: Asynchronous & Synchronous Counters. Modulo-N Counters and Up-Down Counters.

Suggested Readings:

1. Mano, M.M. : Digital Logic and Computer Design, Prentice- Hall of India.
2. Gill, Nasib Singh and Dixit J.B.: Digital Design and Computer Organisation, University Science Press (Laxmi Publications), New Delhi.
3. Stallings, William: Computer Organisation & Architecture.
4. Mano, M.M. : Digital Design, Prentice-Hall of India.
5. Anand Kumar : Fundamentals of Digital Circuits, PHI.
6. Tokheim : Digital Electronics, TMH.
7. S. Rangnekar: Digital Electronics, ISTE/ EXCEL
8. Any other book covering the contents of the subject.

Note: Latest and additional good books may be suggested and added from time to time.

Name of the Program	**4-Year UG program in Computer Applications**	**Program Code**	**-----**
Name of the Course	**Data and File Structures**	**Course Code**	**23BCA402DS02**
Hours/Week	**5 (3+2)**	**Credits (L:T:P)**	**3:0:1**
Max. Marks.	**Theory: 75 (50+25)** **Practical: 25 (20+5)**	**Time of end term examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of 5 parts (short-answer type questions) covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory.

Course Objectives:

The course aims to empower students with a comprehensive skill set in data and file structures, fostering both theoretical understanding and practical application, preparing them for analyzing and applying algorithms, design and apply efficient algorithms using data structures, understand the significance of efficient data and file organization, develop coding proficiency in data structure applications.

Course Outcomes:

By the end of the course the students will be able to:
CO1: Understand the fundamental concepts of data structures.
CO2: Design and implement various data structures to solve computational problems.
CO3: Apply data structures for efficient storage and retrieval of information.
CO4: Develop algorithms for searching and sorting data.
CO5: Implement file handling operations in a programming environment.

Unit – I

Introduction: Elementary data organization, Data Structure definition, Data type vs. data structure, Categories of data structures, Data structure operations, Applications of data structures.

Arrays: Introduction, Linear arrays, Representation of linear array in memory, address calculations, Traversal, Insertions, Deletion in an array, Multidimensional arrays, Parallel arrays, Sparse arrays.

Searching: Introduction, Sequential search, Binary search, Prerequisite for binary search, Comparison in terms of efficiency.

Unit – II

Sorting: Bubble sort, Selection sort, Insertion sort, Quick sort, Merge sort, Comparison in terms of their efficiency
Stack: Introduction, Array and linked representation of stacks, Operations on stacks, Applications of stacks: Polish notation, Recursion.

Queues: Introduction, Array and linked representation of queues, Operations on queues, Deques, Priority Queues, Applications of queues.

Unit – III

Linked List: Introduction, Representation of linked lists in memory, Traversal, Insertion, Deletion, Searching in a linked list, Header linked list, Circular linked list, Two-way linked list, Threaded lists, Garbage collection, Applications of linked lists.
Tree: Introduction, Definition, Representing Binary tree in memory, Traversing binary trees, Traversal algorithms using stacks.
Graph: Introduction, Graph Theory terminology, Sequential and Linked representation of Graphs.

Unit – IV

Introduction to file structures: Concept of a file, types of files, File operations - open, read, write, close. External storage devices, Concepts of record, file, database and database system.

File Organization: Sequential file organisation – structures and processing, Record structures and access methods. Indexed sequential file organisation – structures and processing, Indexing techniques, B-trees and hashing for indexed files. Direct file organisation. Hashed File Organization - Hash function implementation.

Suggested Readings:

1. Seymour Lipschutz, “Data Structure”, Tata-McGraw-Hill
2. Horowitz, Sahni & Anderson-Freed, “Fundamentals of Data Structures in C”, Orient Longman.
3. Trembley, J.P. And Sorenson P.G., “An Introduction to Data Structures With Applications”, McGraw-Hill International Student Edition, New York.
4. Mark Allen Weiss Data Structures and Algorithm Analysis In C, Addison- Wesley, (An Imprint Of Pearson Education), Mexico City.Prentice- Hall Of India Pvt. Ltd., New Delhi.
5. Yedidyan Langsam, Moshe J. Augenstein, and Aaron M. Tenenbaum, “Data Structures Using C”, Prentice- Hall of India Pvt. Ltd., New Delhi.
6. Any other book covering the contents of the subject.

Note: Latest and additional good books may be suggested and added from time to time.

List of Programs

1. Write a C program to demonstrate basic data structure operations such as creating, inserting, deleting, and displaying elements in an array.
2. Write a C program to insert and delete an element at a specified position in a linear array.
3. Write a C program to implement and demonstrate a sequential search on an array.
4. Write a C program to implement and demonstrate a binary search on a sorted array.
5. Write a C program to implement and demonstrate the Bubble sort algorithm.
6. Write a C program to implement and demonstrate the Selection sort algorithm.
7. Write a C program to implement and demonstrate the Insertion sort algorithm.
8. Write a C program to implement and demonstrate the Quick sort algorithm.
9. Write a C program to implement and demonstrate the Merge sort algorithm.
10. Write a C program to implement a stack using an array and perform basic stack operations: push, pop, and display.
11. Write a C program to implement a queue using an array and perform basic queue operations: enqueue, dequeue, and display.
12. Write a C program to implement a singly linked list and perform insertion, deletion, and traversal operations.
13. Write a C program to implement a binary tree and perform in-order, pre-order, and post-order traversal using recursion.
14. Write a C program to implement a binary search tree (BST) and perform insertion, deletion, and search operations.
15. Write a C program to represent a graph using an adjacency matrix and perform a depth-first search (DFS).
16. Write a C program to represent a graph using an adjacency list and perform a breadth-first search (BFS).
17. Write a C program to demonstrate basic file operations: open, read, write, and close.

Any other programs assigned by the teachers.

Name of the Program	**4-Year Bachelor of Computer Applications**	**Program Code**	**-----**
Name of the Course	**Programming in C & Data Structure**	**Course Code**	**23BCA402MI01**
Hours/Week	**5 (3+2)**	**Credits (L:T:P)**	**3:0:1**
Max. Marks.	**Theory: 75 (50+25)** **Practical: 25 (20+5)**	**Time of end term examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of 5 parts (short-answer type questions) covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory.

Course Objectives:

The objective of this course is to inculcate logical thinking amongst the young minds and to teach the C Programming Language and data structure. Students will be able to develop logics which will help them to create programs and applications in C. By learning the basic programming constructs, they can easily switch over to any other language in future.Students will also understand the concept of data structure which will help them to develop more complex programs.

Course Outcomes:

By the End of course the students will be able to:
CO1 Achieve Knowledge of design and development of C problem solving skills
CO2 Understand the basic principles of Programming in C language
CO3 Implement the concept of functions, arrays and pointrrs in C language
CO4 Understand and implement the concept of linear data structure such as link list, stack and queue.
CO5 Understand and implement the non linear data structures such as tree and graph.

Unit – I

C Programming Fundamentals: Basic concepts of a C program, Declaration, Assignment & Print statements, Data Types, operators and expressions, decision making & branching, Programming examples.

Unit – II

Functions: Standard Mathematical functions, Input/output: Unformatted & formatted I/O function in C, Input functions, output functions.

User defined functions: Introduction/Definition, prototype, Local and global variables, passing parameters.

Introduction to array and pointers.

Unit – III

Introduction: Elementary data organization, Data Structure, Categories of data structures, Data structure operations, Applications of data structures.

Linear Data Structure: Array, Linked List, Stack, Queue and their Implementation.

Unit – IV

Non-Linear Data Structure: Tree; Introduction, Definition, Representing Binary tree in memory, Traversing binary trees, Traversal algorithms using stacks.

Graph: Introduction, Graph theory terminology, Sequential and linked representation of graphs.

Suggested Readings:

1. Yashwant Kanetker: Let us C, BPB.
2. Rajaraman, V.: Computer Programming in C, PHI.
3. Yashwant Kanetker: Working with C, BPB.
4. Trembley, J.P. And Sorenson P.G.: An Introduction to Data Structures With Applications, McGraw-Hill International Student Edition, New York.
5. Mark Allen Weiss: Data Structures and Algorithm Analysis In C, Addison- Wesley, (An Imprint Of Pearson Education), Mexico City.Prentice- Hall Of India Pvt. Ltd., New Delhi.
6. Yedidyan Langsam, Moshe J. Augenstein, and Aaron M. Tenenbaum: Data Structures Using C, Prentice-Hall of India Pvt. Ltd., New Delhi.
7. Any other book covering the contents of the subject.

Note: Latest and additional good books may be suggested and added from time to time.

List of Programs

1. Write a C program to declare variables, assign values to them, and print the values using printf().
2. Write a C program to demonstrate the working of various operators (arithmetic, relational, logical, and bitwise)
3. Write a C program to implement decision making using if-else statements.
4. Write a C program to demonstrate the use of switch statement
5. Write a C program to demonstrate the use of loops (for, while, do-while) by printing the first 10 natural numbers.
6. Write a C program to use standard mathematical functions like sqrt(), pow(), and abs(). Prompt the user to enter values and display the results.
7. Write a C program to demonstrate the use of unformatted and formatted input/output functions (getchar(), putchar(), scanf(), printf()).
8. Write a C program to define and use a user-defined function to calculate the area of a circle. Use function prototypes and pass parameters by value.
9. Write a C program to swap two numbers using call by value and call by reference (pointers).
10. Write a C program to declare and initialize an array, then find and print the largest and smallest elements in the array.
11. Write a C program to implement a stack using an array and perform basic operations: push, pop, and display.
12. Write a C program to implement a queue using an array and perform basic operations: enqueue, dequeue, and display.
13. Write a C program to implement a singly linked list and perform insertion, deletion, and traversal operations.
14. Write a C program to implement a binary tree and perform in-order, pre-order, and post-order traversal using recursion.
15. Write a C program to implement a binary search tree (BST) and perform insertion, deletion, and search operations.

Any other programs assigned by the teachers.

Multidisciplinary Courses (MDC) 2nd Semester

[To be chosen by the students other than that of Bachelor Program in Computer Applications]

Name of the Program	**4-Year Bachelor of Computer Applications**	**Program Code**	**-----**
Name of the Course	**Office Automation**	**Course Code**	**23BCAX02MD01**
Hours/Week	**5 (1+4)**	**Credits (L:T:P)**	**1:0:2**
Max. Marks.	**Theory: 25 (20+5)** **Practical: 50 (35+15)**	**Time of end term examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of 4 short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory.

Course Objectives:

Office automation course enables students in crafting professional word documents, excel spread sheets, power point presentations using the Microsoft suite of office tools. To familiarize the students in preparation of documents and presentations with office automation tools.

Course Outcomes:

By the end of the course the students will be able to:
CO1: Understand the fundamental concepts of window operating system.
CO2: Understand and use of various functions of windows.
CO3. Use Word processing tool for ext processing.
CO4: Learn Data analysis tool for effective data analysis.
CO5: Learn about Presentation tool for creating presentations.

Unit – I

MS-Windows: Operating system-Definition & functions, basics of Windows. Basic components of windows, icons, types of icons, taskbar, activating windows, using desktop, title bar, running applications, exploring computer, managing files and folders, copying and moving files and folders. Control panel – display properties, adding and removing software and hardware, setting date and time, screensaver and appearance. Using windows accessories.

Unit – II

Documentation Using MS-Word: Introduction to word processing interface, Toolbars Creating & Editing Document, Formatting Document, Finding and replacing text, Format painter, Header and footer, Drop cap, Auto-text, Autocorrect, Spelling and Grammar Tool, Document Dictionary, Page Formatting, Bookmark, Previewing and printing document, Advance Features of MS-Word-Mail Merge, Macros and Tables

Unit – III

Electronic Spread Sheet using MS-Excel: Introduction to MS-Excel, Cell, cell address, Creating & Editing Worksheet, Formatting and Essential Operations, Moving and copying data in excel, Header and footer, Formulas and Functions, Charts, Cell referencing, Page setup, Macros, Advance features of MS-Excel-Pivot table & Pivot Chart, Linking and Consolidation, Database Management using Excel-Sorting, Filtering, Validation, What if analysis with Goal Seek

Unit – IV

Presentation using MS-PowerPoint: Presentations, Creating, Manipulating & Enhancing Slides, Organizational Charts, Excel Charts, Word Art, Layering art Objects, Animations and Sounds, Inserting Animated Pictures or Accessing through Object, Inserting Recorded Sound Effect or In-Built Sound Effect.

Suggested Readings:

1. Microsoft Office – Complete Reference – BPB Publication
2. Russell A. Stultz: Learn Microsoft Office — BPB Publication
3. Courter, G Marquis: Microsoft Office 2000, Professional Edition. BPB.
4. Koers, D: Microsoft Office XP Fast and Easy. PHI.
5. Nelson, S L and Kelly, J: Office XP: The Complete Reference. Tata McGraw- Hill.
6. Any other book covering the contents of the subject.

Note: Latest and additional good books may be suggested and added from time to time.

List of Programs

1. Create a document in MS-Word introducing the word processing interface.
2. Write a step-by-step guide on how to find and replace text in MS-Word. Include examples and screenshots.
3. Create a document in MS-Word showcasing the use of Format Painter, Header and Footer, Drop Cap, AutoText, and Autocorrect features.
4. Write a tutorial document in MS-Word demonstrating the use of the Spelling and Grammar Tool, Document Dictionary, and Page Formatting options.
5. Create a document in MS-Word with bookmarks and demonstrate how to navigate through them. Include a section on previewing and printing documents.
6. Write a guide on using Mail Merge in MS-Word to create personalized letters. Include steps for merging data from an Excel sheet.
7. Create a document in MS-Word with macros to automate repetitive tasks. Explain how to record and execute macros.
8. Design a table in MS-Word to organize data and demonstrate sorting, filtering, and validation functionalities.
9. Create a budget spreadsheet in MS-Excel, including essential operations like creating & editing worksheets, formatting cells, and formulas.
10. Design a worksheet in MS-Excel with a header and footer. Demonstrate how to customize headers and footers for printing.
11. Create a spreadsheet in MS-Excel with various formulas and functions. Include examples of basic arithmetic operations and statistical functions.
12. Develop a chart in MS-Excel to visualize data. Include bar, line, and pie charts, and explain when to use each type.
13. Create a PowerPoint presentation with slides showcasing different slide manipulation features like transitions, animations, and organizational charts.
14. Design a presentation in MS-PowerPoint with Excel charts and WordArt. Demonstrate how to import and manipulate these objects.
15. Create a PowerPoint presentation with animated pictures and sound effects. Include slides with recorded sound effects and animated pictures accessed through objects.

Any other programs assigned by the teachers.

Name of the Program	**4-Year Bachelor of Computer Applications**	**Program Code**	**-----**
Name of the Course	**Python Programming**	**Course Code**	**23BCA402SE01**
Hours/Week	**5 (1+4)**	**Credits (L:T:P)**	**1:0:2**
Max. Marks.	**Theory: 25 (20+5)** **Practical: 50 (35+15)**	**Time of end term examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of 4 short-answer type questions covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory.

Course Objectives:

The course is designed to impart knowledge of one of the latest and most powerful programming languages – Python. Python programming is intended for software engineers, system analysts, program managers and user support personnel who wish to learn the Python programming language.

Course Outcomes:

By the end of the course the students will be able to:
CO1: Develop problem-solving skills and critical thinking in Python using basic programming constructs including variables, operators and data types.
CO2: Will be able to learn the automating repetitive tasks using loop and conditional controlled statements.
CO3:Understand the complex data types including lists, tuples, dictionaries and Function packages.
CO4: Identify and use libraries for algorithmic thinking to implement various data structures.
CO5: Will be able to implement important concept of Database programming.

Unit – I

Introduction to Python: History and Features of Python Programming, Basics of Python:Keywords, Variables, Operators, I/O Statements, Indentation, and Comments. Python Basic Data Types, Data Types Declaration, and Implementation.

Unit – II

Flow Control Statement: if statement, if-else statement, nested-if statement,if-elif-else ladder, While loop, range() Function, For Loop, Nested Loops, Infinite Loop, Break Statement, Continue Statement, Pass Statement

Unit – III

Python Complex data types: String Data Type, String Manipulation Methods and implementation using Python Programming
List and Dictionary Data Type, Declaration, and Implementation using Various built-in Functions and Libraries

Unit – IV

Python File Operations: Reading Files, Writing Files in Python, Understanding Read Functions:read(), readline(), readlines(), Understanding Write Functions: write() and writelines() Manipulating file pointer using seek Programming, using file **operations**.

Database Programming: Connecting to a Database, Creating Tables, INSERT, UPDATE, DELETE and READ operations, Transaction Control, Disconnecting from a database, and Exception Handling in Databases.

Suggested Readings:

1. Al Sweigart: Automate the Boring Stuff with Python.
2. Allen B. Downey: Think Python: How to Think Like a Computer Scientist, 2nd Edition, Green Tea Press
3. Charles Dierbach: Introduction to Computer Science Using Python, 1st Edition, Wiley India Pvt Ltd.
4. Wesley J Chun: Core Python Applications Programming, 3rd Edition, Pearson Education India
5. Roberto Tamassia, Michael H Goldwasser, Michael T Goodrich: Data Structures and Algorithms in Python, 1st Edition, Wiley India Pvt Ltd
6. Reema Thareja: Python Programming using problem solving approach, Oxford University press.
7. Charles R. Severance: Python for Everybody: Exploring Data Using Python 3, 1 st Edition, Shroff Publishers.
8. Any other book covering the contents of the subject.

Note: Latest and additional good books may be suggested and added from time to time.

List of Python Programs

1. Write a program to convert the given temperature from Fahrenheit to Celsius and vice versa depending upon user’s choice.
2. Write a program, using user-defined function to find the area of a rectangle, square, circle and triangle by accepting suitable input parameters from the user.
3. Write a program to display the first n terms of the Fibonacci series.
4. Write a program to find the factorial of the given number.
5. Write a program to count the number of even and odd numbers from N numbers.
6. Write a program to create a function that accepts a string and calculates the number of upper case letters and lower case letters.
7. Write a program to reverse a given string and check whether the given string is a palindrome or not.
8. Write a program to find the sum of all items in a dictionary.
9. Write a program to perform arithmetic operations (addition, subtraction, multiplication, and division) on two numbers entered by a user.
10. Write a program to find the largest and smallest numbers in the list entered by the user.
11. Write a program to find whether the given number is Armstrong Number or not.
12. Write a program to print the multiplication table of a given number.
13. Write a program to check whether a given number is a prime number or not.
14. Write a program to perform string operations (concatenation, slicing, indexing, and length).
15. Write a program to find the largest and smallest number in a matrix entered by the user.
16. Write a program to sort a list of elements using the bubble sort algorithm.
17. Write a program to implement a simple calculator using functions.
18. Write a program to implement a linear search algorithm to search an element in the list entered by a user.
19. Write a program to implement a binary search algorithm to searchan element in the list entered by a user.
20. Write a program to implement a selection sortalgorithm to sort all the elements in the list entered by a user.

Any other programs assigned by the teachers.

Third Semester

Name of the Program	**4-Year Bachelor of Computer Applications**	**Program Code**	-----
Name of the Course	**Operating Systems**	**Course Code**	**24BCA403DS01**
Hours/Week	**4**	**Credits (L:T:P)**	**4:0:0**
Max. Marks.	**Theory: 100 (70+30)**	**Time of end term examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of 7 parts (short-answer type questions) covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory.

Course Objectives:

This objective of this course is to enable students to learn about important concepts related to Operating Systems. It will help the students to enrich their knowledge and understanding of major functions performed by Operating System. It will give indepth knowledge to students covering process management, memory management, secondary storage structure, file management and Input/Output management.

Course Outcomes:

By the end of the course, the students will be able to:
CO1: Understand the basic concepts of Operating Systems and its classification.
CO2: Learn about the functions and structures of Operating system and relevance of System Calls and Services.
CO3: Understand the major: functions of Operating System.
CO4: Learn about different Scheduling algorithms.
CO5: Learn about the the concept of controlled access to system resources through authentication and disk scheduling.

Unit – I

Introduction to Operating Systems: Objectives and Characteristics. Classification: Batch, Multi-programming, Multi-processing, Multi-tasking, Time-sharing, Distributed, Network and Real time Operating systems. System Calls and Services.

Functions and Structures: Operating System Functions- Process management, Memory management, Secondary storage management, I/O management, File management, Protection and Security. Structures- Simple Structure, Monolithic structure, Layered approach, Microkernel, Exokernel and Virtual Machines.

Unit – II

Process Management and Scheduling: Process concept- Process State Model, Process Control Block and Threads. Process Scheduling- Scheduling Queues, Schedulers and Context Switch. Operations on Processes, Cooperating processes and Inter-Process Communication.

Process Scheduling: Scheduling Criteria, Scheduling Algorithms: Single Processor Scheduling: FCFS, SJF, Round Robin, Multi Feedback Queue. Multiple Processor Scheduling and Real Time scheduling. Scheduling Algorithm Evaluation.

Unit – III

Memory Management: Concepts of Memory Management, Logical and Physical address space, Swapping, Memory allocation: Contiguous and Non-Contiguous. Paging: Hardware Support. Page Map Table and Protection. Segmentation: Hardware Support and Protection and Sharing.

Virtual Memory: Need of Virtual Memory, Demand paging, Pure Demand Paging. Handling page faults, Performance of Demand Paging. Page replacement Algorithms and Allocation of Frames: Allocation algorithms and Global vs Local Allocation. Thrashing.

Unit – IV

I/O Management: Basic I/O Devices, Types of I/O Devices: Block and Character Devices. I/O Software: Device Independent I/O, User Space I/O and Kernel I/O Software. Device Controllers, Device Drivers and Interrupt Handlers. Communication Approaches to I/O Devices: Special Instruction I/O, Memory Mapped I/O and Direct Memory Access (DMA). Secondary Storage Structure: Disk Structure and Disk Scheduling Algorithms.

File System Interface: File Concept: Attributes, Operations and Types. File Access Methods: Sequential Access, Direct Access and Indexed Sequential. Free Space Management. Directory Structures: Single Level, Two level and Tree Structured. File Protection and Sharing.

Suggested Readings:

1. Silberschatz & Galvin: Operating System Concept, Wiley.
2. Milan Milenkovic: Operating Systems, Tata McGraw Hill.
3. William Stallings: Operating Systems, PHI.
4. Yashawant Kanetkar: Operating System Concepts, BPB.
5. H.M. Deitel, P.J Deitel: Operating Systems, Pearson .
6. A.S. Tanenbaum: Modern Operating Systems, Pearson/PHI.
7. Dhamdhere: Operating Systems, Tata McGraw Hill.
8. Any other book(s) covering the contents of the paper in more depth.

Note: Latest and additional good books may be suggested and added from time to time.

Name of the Program	**4-Year Bachelor of Computer Applications**	**Program Code**	
Name of the Course	**Object-Oriented Programming using C++**	**Course Code**	**24BCA403DS02**
Hours/Week	**5 (3+2)**	**Credits (L: T:P)**	**3:0:1**
Max. Marks.	**Theory: 75 (50+25)** **Practical: 25(20+5)**	**Time of end term examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of 5 parts (short-answer type questions) covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory.

Course Objectives:

The objective of this course is to introduce students to the principles and concepts of object-oriented programming (OOP) using the C++ programming language. The course aims to develop students' skills in designing and implementing object-oriented solutions to real-world problems.

Course Outcomes:

By the end of the course the students will be able to:
CO1: Understand the basic concepts of object-oriented programming.
CO2: Design and implement C++ programs using classes, objects, and inheritance.
CO3: Apply polymorphism and templates to develop reusable code.
CO4: Learn about the concept of excdeption handling and working with files.
CO5: Utilize advanced features of C++ to develop efficient and modular programs.

Unit – I

Introduction to OOP concepts: Procedural Vs. Object- Oriented Programming, Principles of OOP and their benefits. Object, classes, Inheritance, Abstraction, Encapsulation Polymorphism, Dynamic Binding, Message passing.
C++ Programming Basics: Syntax and structure of C++ programs, Data types, variables, and constants in C++, Control structures: decision making and looping constructs

Unit – II

Classes and objects: Defining and using classes and objects, Member functions and data members, Access specifiers: public, private, protected, Functions and parameter passing in C++, Arrays and strings in C++, Pointer, Constructors and destructors.
Inheritance: Derived class and Base class, Types of inheritance: single, multiple, multilevel, hierarchical, Access control in inheritance.

Unit – III

Polymorphism: function overloading, Operator overloading, Virtual functions and dynamic polymorphism, Abstract classes and pure virtual functions, Encapsulation and data hiding, Friend functions, static function.
Memory Management: Dynamic Memory Allocation: new, delete, Object Creation at run time.

Unit – IV

Exception handling: Throwing, Catching, Re-throwing an exception, specifying exception: processing unexpected exceptions; try-catch blocks, Exception propagation, Templates: class and function templates, Standard Template Library: benefits of STL and generic programming
Working with Files: Stream Classes, File input and output Operations in C++, Error handling during file operations.

Suggested Readings:

1. Bala Guruswamy : Object Oriented Programming and C++, THM
2. Rajaram : Object Oriented Programming and C++, New Age International.
3. Herbert Schildt : The complete Reference, 4th Edition, McGraw Hill Publications.
4. Shah & Thakker: Programming in C++, ISTE/EXCEL.
5. Subburaj : Object Oriented Programming with C++, Vikas.
6. Samanta : Object Oriented Programming with C++ & JAVA,PHI.
7. Stanley B. Lippman, Jose Lajoie, and Barbara E.Moo : C++ Primer .

Note: Latest and additional good books may be suggested and added from time to time.

List of Programs

1. Create a C++ program to take two numbers as input from the user and display their sum.

2. Write a C++ program to demonstrate decision-making constructs like if-else and looping constructs like for and while loops.
3. Write a C++ program to define a class called `Rectangle` with attributes `length` and `width`, and display the area of the rectangle.
4. Implement a C++ program to demonstrate the concept of inheritance by creating a base class `Shape` and derived class `Rectangle`. Display the area of the rectangle using inheritance.
5. Create a C++ program to define a class called `Student` with attributes `name` and `roll number`. Use member functions to input and display student details.
6. Write a C++ program to demonstrate the use of constructor and destructor in a class.
7. Implement a C++ program to showcase the use of access specifiers (`public`, `private`, `protected`) in a class.
8. Create a C++ program to demonstrate function overloading by defining multiple functions with the same name but different parameters.
9. Write a C++ program to demonstrate dynamic polymorphism using virtual functions.
10. Write a C++ program to demonstrate the working of friend function.
11. Implement a C++ program to demonstrate the use of pointers to objects. Define a class `Book` with attributes `title` and `author`, and use pointers to access and display book details.
12. Write a C++ program to handle exceptions using `try-catch` blocks.
13. Create a C++ program to implement a simple template function to find the largest amomg two numbers. Test the function with different data types.
14. Write a C++ program to perform file input and output operations, including opening, reading, writing, and closing files.
15. Implement error handling during file operations in a C++ program, by handling exceptions and error codes.

Any other programs assigned by the teachers.

Name of the Program	**4-Year Bachelor of Computer Applications**	**Program Code**	**-----**
Name of the Course	**Database Management System**	**Course Code**	**24BCA403DS03**
Hours/Week	**5 (3+2)**	**Credits (L:T:P)**	**3:0:1**
Max. Marks.	**Theory: 75 (50+25)** **Practical: 25 (20+5)**	**Time of end term examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of 5 parts (short-answer type questions) covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory.

Course Objectives:

The objective of this course is to teach the student concepts related to database, database design techniques, transaction management, crash recovery and backup.

Course Outcomes:

By the end of the course the students will be able to:
CO1: Learn about basics of database and database management system.
CO2: Understand about data models and file organization.
CO3: Indepth understanding of relational data models, databases and normalization process.
CO4: Learn about transaction processing and concurrency control techniques.
CO5: Understanding about relevance of recovery and backups.

Unit – I

Database Management System: Introduction, Database System Applications, History of Database Systems, Database System Vs. File Processing System, View of Data, Data Abstraction, Instances and Schemas. DBMS Environment, Database languages, Database Models.

Database design and ER Model: Physical, Conceptual and Logical Database design, Entity- Relationship Model: Entities, Relationships, Representation of entities, attributes, Representation of relationship set, Generalization, Aggregation, Conceptual design with ER Model

Unit – II

Relational Model: Introduction to the Relational Model, Attributes, Domains, Tuples, Relations and their schemes, relation representation, Keys, relationship, relational operations, , Integrity Constraint Over relations, Enforcing Integrity constraints, Querying relational data, View: Introduction to Views, Destroying / altering Views.

Relational Algebra and Calculus: Relational Algebra & its operations, Relational calculus & its types, Power of Algebra and calculus.

Lab Problem(s): ***Creation and Querying relational data with SQL***

Unit – III

Normalization: Schema Refinement, Problems caused by redundancy, Decomposition & its properties; Normalization: First, Second, Third Normal forms, BCNF, Multivalued Dependencies, Join Dependencies.

Transaction Management & Concurrency Control: ACID properties, Transactions and Schedules, Concurrent execution of transaction, Serializability and Recoverability, Lockbased Concurrency control, Lock Management, Lock Conversion, Dealing with deadlocks, Concurrency without Locking.

Unit – IV

Crash Recovery and Backup: Failure classifications, storage structure, Recovery & Atomicity, Log base recovery, Recovery with concurrent transactions, Failure with loss of nonvolatile storage, Database backup & recovery from catastrophic failure, Remote Backup System.

Storage and File Organization : Overview of physical storage media, Storage access; File organization, Operations on Files, Serial Files, Sequential Files , Index-Sequential Files, Direct Files.

Suggested Readings:

1. Ramez Elmasri, Shamkant B. Navathe: Fundamentals of Database Systems, Pearson.
2. Silberschatz Abraham : Database System Concept, Tata Mc Graw Hill, Latest edition.
3. C. J Date : Introduction to Database Systems, Pearson Education, Latest edition.
4. Krishnan Ram and Gehrke : Database Management System, Tata McGraw Hill.
5. Byross Ivan : Oracle 10 G The Database with HTML Database, BPB publication.

6. Any other book covering the contents of the subject.

Note: Latest and additional good books may be suggested and added from time to time.

List of Programs

1. Write SQL queries for DDL commands: Create, Desc, Alter, Rename, Drop.
2. Write SQL queries by using DML commands: Insert, Select, Update, Delete.
3. Write SQL queries using Logical operations: AND, OR, NOT, IN, BETWEEN, LIKE.
4. Write SQL queries using Arithmetic operators: +, -, *, /, %.
5. Write SQL demonstrating the use of Comparison operators: =, <>, >, >=, <, <=.
6. Write SQL queries using Aggregate functions: AVG, MIN, MAX, SUM, COUNT.
7. Write SQL queries to get current date and time: NOW, CURDATE, CURTIME, LAST_DAY, DATE_FORMAT.
8. Write SQL queries using Character functions: ASCII, CHAR, CONCAT, INSTR, INSERT, LENGTH, LOWER, UPPER.
9. Write SQL queries using Number functions: POWER, ROUND, SQRT, EXP, GREATEST, LEAST, MOD, FLOOR, CEIL.
10. Write SQL queries for Relational algebra operations: SELECT, PROJECT, UNION, INTERSECTION, DIFFERENCE.
11. Write SQL queries to perform table joins: INNER JOIN, LEFT OUTER JOIN, RIGHT OUTER JOIN, FULL OUTER JOIN.
12. Write SQL queries to create, update, and delete views: CREATE VIEW, UPDATE VIEW, DROP VIEW.
13. Write SQL queries for subqueries and nested queries.
14. Write SQL queries to create and execute procedures.
15. Write SQL queries to create and manage triggers: CREATE TRIGGER, DROP TRIGGER.

Any other programs assigned by the teachers.

Name of the Program	**4-Year Bachelor of Computer Applications**	**Program Code**	**-----**
Name of the Course	**Database Management System & SQL**	**Course Code**	**24BCA403MI01**
Hours/Week	**5 (3+2)**	**Credits (L:T:P)**	**3:0:1**
Max. Marks.	**Theory: 75 (50+25)** **Practical: 25 (20+5)**	**Time of end term examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of 5 parts (short-answer type questions) covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory.

Course Objectives:

To gain foundational knowledge of database management systems (DBMS) and SQL, enabling students to understand database concepts, design efficient database schemas, and write SQL queries proficiently. This course also prepares students to analyze and optimize database performance, ensure data integrity, and apply transaction processing principles in real-world scenarios.

Course Outcomes:

By the end of the course, students will be able to:
CO1: Demonstrate proficiency in fundamental database concepts and principles.
CO2: Apply SQL for data definition, manipulation, and retrieval tasks effectively.
CO3: Design and implement efficient database schemas and enforce data integrity constraints.
CO4: Analyse and optimize database performance, including query optimization and indexing strategies.
CO5: Understand transaction processing concepts in database systems.

Unit – I

Database Management System: Introduction, History of Database Systems, Database System Vs. File Processing System, View of Data, Data Abstraction, Instances and Schemas. DBMS Environment, Database languages, Database Models.

Database design and ER Model: Physical, Conceptual and Logical Database design, ER Modelling, Conceptual design with ER Model
Extended Entity-Relationship (EER) Model: Superclass/Subclass relationship, Specialization, Generalization, Union, Aggregation.

Unit – II

Relational Model: Introduction to the Relational Model, Integrity Constraint Over relations, Enforcing Integrity constraints, Querying relational data, View: Introduction to Views, Destroying / altering Views.

Relational Algebra and Calculus: Relational Algebra & its operations, Relational calculus & its types, Power of Algebra and calculus.

Unit – III

Normalization: Schema Refinement, Problems caused by redundancy, Functional dependencies. Decomposition & its properties; Normalization: First, Second, Third Normal forms, BCNF, Computing closures of set FDs, Multivalued Dependencies, Join Dependencies.

SQL: Types of SQL, Components of SQL, data types, specifying constraints,Querying the Relational data, Create Simple Queries Using (Where, Like, Group By, Having, Order By), View table Structure, Range Searching, Pattern Matching , Nested Queries, Creation of Table using another table, Insertion of data using another table, Views: Creation, updation and deletion of views; Join: Equi Join, Natural Join, Inner Join and Outer Join.

Unit – IV

Transaction Management Introduction to transaction processing: ACID properties, Transactions and Schedules, Concurrent execution of transaction, Serializability and Recoverability, Lock based Concurrency control, Lock Management, Lock Conversion, Dealing with deadlocks,
Data storage: Introduction to indexing structures for files. Data Storage Structures: Pages, Blocks, and Extents, Buffer Management and Buffer Pooling, Disk Management: Allocation, Access Methods, and File Structures

Suggested Readings:

1. Elmasri & Navathe: Fundamentals of Database Systems,Latest edition, Pearson Education.
2. Thomas Connolly, Carolyn Begg: Database Systems, Pearson Education.

3. C. J. Date: An Introduction to Database Systems, Latest edition, Addison Wesley N. Delhi.

4. Any other book(s) covering the contents of the paper in more depth.

Note: Latest and additional good books may be suggested and added from time to time.

List of Programs

1. Write a query using SQL commands to create a table with columns of your choice and perform the following operations: Add a new column in the table, and delete the table from the database.
2. Write an SQL command to insert a new record , to update a column value and delete a specific record from the table.
3. Write SQL commands to showcase the use \`ROUND\`, \`COUNT\` , \`UPPER\` , \`SYSDATE\` , and \`TO_CHAR\` functions
4. Write SQL command to perform arithmetic operations on numeric columns of the given table.
5. Write SQL command to find records that satisfy either of two conditions using logical operators.
6. Write SQL command using the \`BETWEEN\` operator to find records within a specific range.
7. Write SQL command to perform a natural join between two tables.
8. Write SQL command to group records by a column and count the number of records in each group, displaying only groups that meet a specific condition.
9. Write SQL command to order records by a specific column in ascending or descending order.
10. Write SQL commands to create a view that displays specific columns from a table.
11. Write SQL commands to implement \`PRIMARY KEY\`, \`FOREIGN KEY\`, \`UNIQUE\`, \`CHECK\`, and \`NOT NULL\` constraints on a table.
12. Write SQL commands to demonstrate transaction control using \`ROLLBACK\`, \`COMMIT\`, and \`SAVEPOINT\`.
13. Write SQL command to create a new database and tablespace.
14. Write SQL commands to create a user and delete a user.
15. Write SQL commands to grant and revoke permissions on the given table.

Any other programs assigned by the teachers.

Name of the Program	**4-Year Bachelor of Computer Applications**	**Program Code**	**-----**
Name of the Course	**Android Programming**	**Course Code**	**24BCA403SE03**
Hours/Week	**5 (1+4)**	**Credits (L:T:P)**	**1:0:2**
Max. Marks.	**Theory: 25 (20+5)** **Practical: 50 (35+15)**	**Time of end term examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of 4 parts (short-answer type questions) covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory.

Course Objectives:

This course facilitates classroom and laboratory learning, letting students develop competence and confidence in android programming and understand the entire Android Apps Development Cycle, as well as it would also enable the students to independently create Android Applications.

Course Outcomes:

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

CO1: Understand the basics of Android Programming.
CO2: Develop ability in mobile application development.
CO3: Discover the life cycles of Activities, Applications, intents and fragments of Android apps.
CO4: Design the Android apps by using Java.
CO5: Gain understanding about iOS SDK for developing Andoid apps

Unit – I

Basic of Android Programming: Introduction to Android OS, Setting up the Android Application Development Environment, Creating, Testing and Debugging Applications, Android Stack, Android applications structure, Activity life cycle, Understanding implicit and explicit intents.

User Interface in Android: Adaptive and responsive user interfaces, User Input Controls, Menus, Screen Navigation, Recycler View, Drawable, Themes and Styles, Fragments Fragment Life Cycle, Introduction to Material Design, Testing the user interface.

Unit – II

Background tasks: AsyncTask, AsyncTaskLoader, Connecting App to Internet, Broadcast receivers, Services, Notifications, Alarm managers.

Sensor, Location and Maps: Sensor Basic, Motion and Position Sensors, Location services, Google maps API, Google Places API.

Working with data in Android: Shared Preferences, App Setting, SQLite primer, Store data using SQLite database, Content Providers, Content Resolver, Loader

Unit – III

Advances in Android: Android Debugging, Other view, Notification, Toast, Thread, AsyncTask, Handler & Runnable, Gradle plugins, localization, NFC, SMS sending receiving, Phone calls, Sending Emails, GPS, MAPS, Location based service, Sensors, Network Connectivity Services, adb tools, Interfacing with PHP and MySQL for storing data. SQLite Overview, Query Introduction, Greed Dao: Android ORM for Sqlite Database, Core Classes, Modelling entities, Session, Queries, Relations, Joins, Create a mini-project.

Unit – IV

Performance Improvement of App: Performance Parameters, Profiling Tools, Rendering and Layout, Garbage Collection and Memory Leaks, Best Practices.

Other Mobile Application Development: iOS Platform Overview, basics of iOS Dev Center, iOS SDK, Understanding the Skeleton APP, Simple Application creation. Understanding cordova, Environment understanding, Application Skeleton, Basics on Cordova Core Components and Cordova Plugins.

Suggested Readings:

1. J.F. DiMarzio : Android, A Programming Guide, Osborne/McGraw-Hill.
2. Ed Burnett: Hello, Android: Introducing Google's Mobile Development Platform, Shroff/Pragmatic Bookshelf.
3. Zigurd Mednieks : Programming Android, O′Reilly.
4. Ian G. Clifton: Android User Interface Design: Turning Ideas and Sketches into Beautifully Designed

Apps, Addison-Wesley Professional.

5. Android Developer Fundamental Course, Google.
6. Advance Android Developer Course, Google .
7. Reto Meier: Profession Android Application Development, Wrox Publication.
8. Marko Gargenta and Masumi Nakamura: Learning Android, O'REILLY.
9. Any other book(s) covering the contents of the paper in more depth.

Note: Latest and additional good books may be suggested and added from time to time.

List of Programs

1. Create an Android application that displays "Hello, World!" text in the middle of the screen.
2. Create an Android application to display different dialog boxes.
3. Develop a calculator Android application that performs basic arithmetic operations.
4. Implement the concept of AsyncTask in an Android application to perform background operations and update the UI.
5. Create a login activity in an Android application. It should ask for a "username" and "password" from the user. If the username and password are valid, it should display a welcome message; otherwise, it should prompt the user to retry.
6. Create an Android application to select an item from a given list using AutoCompleteTextView (ACTV).
7. Create an Android application to display dropdown menu items and allow the user to pick one item.
8. Create an Android application to display internal storage data using ArrayAdapter.
9. Create an Android application to demonstrate the concept of SQLite database storage by performing CRUD (Create, Read, Update, Delete) operations.
10. Create an Android application to perform different types of operations (sending SMS, making a call, and sending email).
11. Create an Android application to get latitude and longitude values using the Location Service.
12. Create an Android application to display X, Y sensor values using the Sensor Service.
13. Create an Android application to get notifications on the notification bar using the Notification Service.
14. Create an Android application to display the current location on Google Maps using the Google Maps service.
15. Write an Android program to develop a Camera and Gallery application, allowing the user to take pictures and view them in the gallery.

Any other programs assigned by the teachers.

Fourth Semester

Name of the Program	**4-Year Bachelor of Computer Applications**	**Program Code**	**-----**
Name of the Course	**Computer System Architecture**	**Course Code**	**24BCA404DS01**
Hours/Week	**4**	**Credits (L:T:P)**	**4:0:0**
Max. Marks.	**Theory: 100 (70+30)**	**Time of end term examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of 7 parts (short-answer type questions) covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory.

Course Objectives:

Upon completion of this course, students will have better understanding of basic operations, principles, and components of Computer Architecture. Students will learn about different types of registers, instructions and interrupts used in Computer System. This course will empower the students to understand complex concepts related to computer architecture with ease such as addressing modes, types of memory etc.

Course Outcomes:

By the end of the course the students will be able to:
CO1: Understand the basic Computer concepts such as its different functional units and bus structure.
CO2: Interpret the functional architecture of Computer system in terms of its different components.
CO3: Analyze different types of Instructions and Interrupts used in Computer system..
CO4: Identify and compare different methods for computing Input/Output.
CO5: Explore Memory organization and different operations related to it..

Unit – I

Basic Computer Concepts: Computer Types: Analog and Digital, Digital Computer Organization and Architecture: Functional Units, Basic Organizational Concepts, Von-Neumann Architecture, Bus Structure. Data representation: Number Systems, Fixed and Floating point representation.

Register Transfer and Microoperations: Basic concepts and types of Registers, Register Transfer Language, Data Transfer between Registers, Bus and Memory Transfer, Microoperations: Arithmetic, Logic and Shift Microoperations.

Unit – II

Basic Computer Design: Instruction codes, Common Bus System Architecture, Computer Instructions: Instruction Set and Instruction Cycle. Types of Instructions: Register Reference, Memory Reference and Input-Output Reference.

Interrupts: Classifications of Interrupts: Maskable and Non maskable, Hardware and Software. Interrupt Service Routine, Context Switching, Interrupt Identification: Daisy Chaining, Polling and Vectored Interrupt. Interrupt Cycle.

Unit – III

Central Processing Unit: Introduction, General Register Organization, Stack Organization: Register and /memory Stack, Reverse Polish Notation. Instruction Formats: Three Address, Two Address, One Address and Zero Address.

Data Transfer and Manipulation: Addressing Modes: Implied, Immediate, Register Direct and Indirect Mode and Direct Address Mode. Data Transfer and Data Manipulation Instructions, Program Control, Reduced Instruction Set Computer (RISC) and Complex Instruction Set Computer (CISC).

Unit – IV

Memory Organization: Memory Hierarchy, Main Memory Technologies, Auxiliary Memory, Associative Memory: Hardware requisites, Working Principle and Operations. Cache Memory: Characteristics, Types of Mapping, Writing into Cache, Cache Coherence.

Input-Output Organization: Peripheral Devices, Input-Output Interface, Asynchronous Data Transfer, Modes of Transfer: Programmed I/O and Interrupt Initiated I/O. Concepts related to Priority Interrupt: Daisy Chaining Priority and Parallel Priority Interrupt. Direct Memory Access (DMA): DMA Controller and DMA Transfer.

Suggested Readings:

1. Mano, M.M.: Computer System Architecture, Prentice-Hall of India.
2. Stallings, William: Computer Organisation & Architecture, Pearson Education.
3. Gill Nasib Singh and Dixit J.B: Digital Design and Computer Organisation, University Science Press (Laxmi Publications), New Delhi.
4. Mano, M.M.: Digital Logic and Computer Design, Prentice-Hall of India.
5. Hwang Kai: Advanced Computer Architecture, McGraw Hill International.
6. Hayes J. P: Computer System Architecture & Organization, McGraw-Hill Education.
7. Hamacher V. Carl: Computer Organization, McGraw Hill.
8. Any other book(s) covering the contents of the paper in more depth.

Note: Latest and additional good books may be suggested and added from time to time.

Name of the Program	**4-Year Bachelor of Computer Applications**	**Program Code**	**-----**
Name of the Course	**Data Communication & Computer Networks**	**Course Code**	**24BCA404DS02**
Hours/Week	**4**	**Credits (L:T:P)**	**4:0:0**
Max. Marks.	**Theory: 100 (70+30)**	**Time of end term examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of 7 parts (short-answer type questions) covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory.

Course Objectives:

The objective of this course is to inculcate knowledge in students about computer networks and data communication paradigms, about network models and standards, network protocols and their use, wireless technologies.

Course Outcomes:

By the end of the course the students will be able to:
CO1: Understand basics of data communication and computer networks.
CO2: Learn about various network models used in networking.
CO3: Learn basics and categories of various transmission media.
CO4: Learn various routing techniques and congestion control techniques.
CO5: Understand the working of different applications.

Unit – I

Introduction: Data Transmission concepts, transmission impairments, switching, modulation, multiplexing.
Network Hardware: LAN, MAN, WAN, Wireless networks, Internet-works.
Network Software: Layer, Protocols, interfaces and services.
Reference Models: OSI, TCP/IP and their comparison
Transmission Media: Magnetic, twisted pair, coaxial cable, fibre optics, wireless transmission (radio, microwave,infrared).

Unit – II

Data Link Layer : Framing, Error control, Sliding window protocols (one bit, Go back n, selective repeat). DLL Protocols–HDLC, PPP.

Medium Access Sub layer : Channel Allocation, MAC protocols – ALOHA, CSMA protocols, Collision free protocols, Limited Contention Protocols, Wireless LAN protocols, IEEE 802.3, 802.4, 802.5 standards and their comparison.

Unit – III

Network Layer: Design issues, Routing algorithms (shortest path, flooding, flow based, distance vector, hierarchical, broadcast, multicast, for mobile hosts), Congestion control algorithms (Leaky bucket, Token bucket, Choke Packet, Load shedding), Internetworking, IP Protocol, ARP, RARP.

Unit – IV

Transport Layer: Addressing, establishing and releasing connection, flow control, buffering, Internet Transport Protocol (TCP and UDP).

Application Layer: Domain name system, E-mail, File transfer protocol, HTTP, HTTPS,TELNET.

Suggested Readings:

1. Tanenbaum, Andrew S.: Computer Networks (4th Edition), PHI.
2. Gill Nasib Singh: Handbook of Computer Fundamentals, Khanna Book Publishing Company(Pvt.) Limited, New Delhi.
3. Forouzan, B. A.: Data Communications and Networking, Fourth Edition, Tata McGraw Hill.
4. Douglas E. Comer: Internet Working with TCP/IP (Vol.1, 4th Edition), CPE.
5. Stallings, William : Data and Computer Communications (8th Edition), PHI.
6. Nance, Bary: Introduction to Networking, PHI, 4th Edition.
7. Doerr Alan &Levasseur Kenneth: Applied Discrete Structures for Computer Science, Galgotia Pub. Pvt. Ltd.
8. Any other book covering the contents of the subject.

Note: Latest and additional good books may be suggested and added from time to time.

Name of the Program	**4-Year Bachelor of Computer Applications**	**Program Code**	**-----**
Name of the Course	**Java Programming**	**Course Code**	**24BCA404DS03**
Hours/Week	**6 (2+4)**	**Credits(L:T:P)**	**2:0:2**
Max. Marks.	**Theory: 50 (35+15)** **Practical: 50 (35+15)**	**Time of end term examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1consisting of 5 parts (short-answer type questions) covering the entire syllabus. Student will be required toattempt fivequestionsinallbyselectingonequestion fromeachUnitand QuestionNo.1,whichiscompulsory.

Course Objectives:

The objective of this course is to enable the students to equip with the skills and knowledge to design, develop, and deploy Java applications, leveraging the language's powerful features and extensive libraries to solve complex programming challenges.

Course Outcomes:

By the end of the course the students will be able to:
CO1: Gaining an understanding about Java Programming and its constructs.
CO2: Learning to develop proficient Java Programs.
CO3: Understanding how to apply Object-Oriented principles.
CO4: Implementing and Utilizing Java Standard Libraries in developing Java applications.
CO5: Developing GUI and Multithreaded Applications.

Unit – I

Introduction to Java: Java Features, Java Virtual Machine (JVM), Byte code, Java API, Java Development Kit (JDK), Garbage Collection. Language Basics: Keywords, Constants, Variables and Data Types, Operators and Expressions, Decision Making, Branching and Looping.

Introducing Classes, Objects and Methods: Defining a Class, Methods Declaration, Creating Objects and accessing Class members, Constructors, Methods Overloading, Wrapper Classes, Inheritance, Methods Overriding, Final Class, variables and methods, Abstract Class and Methods, Interfaces.

Unit– II

Arrays, Strings and Vectors: Creating and using Arrays, String operations, String Buffer, String builder, and String Tokenizer class, Vector class.

Packages and Exceptions: Java API packages, Creating and using packages, static import, Exceptions handling, Types of Exceptions, multiple catch statements, 'throw' and 'throws', using 'finally' statement, Creating your own exceptions

Unit–III

Multithreaded Programming: Single threaded and multi-threaded program, Creating threads using Thread class, Life cycle of a Thread, Stopping and blocking a Thread, getting and setting the Thread Priority, Synchronization, implementing the Runnable interface.

Managing Input/Output Streams: Concept to fstreams, Byte and Character streams, Reading and Writing from Console and Files. Input output exceptions.

Unit– IV

Applet Programming: How Applets differs from Java Application, Applet Life Cycle, APPLET Tag, Running an Applet, Passing Parameters to Applet.

Event Handling: Mechanism, The Delegation Event Model, Event Classes, Event Listener Interfaces, Adapter and inner classes.

GUI Programming: Working with Frame Window, Graphics and Text, AWT Controls and classes. Layout Managers, working with Menus.

Suggested Readings:

1. E.Balagurusamy: Programming with Java: A Primer, McGraw Hill.
2. Herbert Schildt: Java:The Complete Reference, McGraw Hill.
3. Bruce Eckel: Thinking in Java, Prentice Hall.
4. Cay S. Horstmann, Gary Cornell: Core Java Volume I—Fundamentals, Prentice Hall.
5. Any other book covering the contents of the subject.

Note: Latest and additional good books may be suggested and added from time to time.

List of Programs

1. Write a Java program to define a class `Student` with data members for name and age. Include member functions to input and display the details of a student.
2. Create a Java program to demonstrate method overloading by creating multiple methods with the same name but different parameters in a class
3. Create a Java program to demonstrate method overriding by creating a base class `Animal` and a derived class `Dog` that overrides a method from the base class.
4. Implement a Java program to define an abstract class `Shape` with an abstract method `draw()`. Create derived classes `Circle` and `Square` that implement the `draw()` method.
5. Implement a Java program to demonstrate the use of the `Vector` class by performing operations like adding, removing, and displaying elements of a vector.
6. Write a Java program to create a user-defined package and use it in another class.
7. Create a Java program to handle multiple exceptions using try-catch blocks.
8. Implement a Java program to demonstrate the use of the `finally` statement in exception handling.
9. Write a Java program to create and throw a custom exception.
10. Create a Java program to implement a multithreaded application by extending the `Thread` class.
11. Implement a Java program to demonstrate thread synchronization.
12. Create a Java program to read and write data to a file using `FileInputStream` and `FileOutputStream`.
13. Write a Java applet to display "Hello, World!".
14. Create a Java applet to demonstrate the use of the `APPLET` tag and pass parameters to the applet.
15. Create a Java program to demonstrate the use of AWT controls like buttons, labels, and text fields.

Any other programs assigned by the teachers.

Name of the Program	**4-Year Bachelor of Computer Applications**	**Program Code**	**-----**
Name of the Course	**Computer Graphics**	**Course Code**	**24BCA404DS04**
Hours/Week	**5 (3+2)**	**Credits (L:T:P)**	**3:0:1**
Max. Marks.	**Theory: 75 (50+25)** **Practical: 25 (20+5)**	**Time of end term examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of 5 parts (short-answer type questions) covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory.

Course Objectives:

The objective of the Computer Graphics course is to provide students with a comprehensive understanding of the fundamental principles and techniques of computer graphics. Students will learn to create and manipulate visual content, explore 2D and 3D modeling, rendering, and animation, and gain proficiency in using graphics software and programming languages to develop visually compelling applications and simulations. The course aims to equip students with the skills necessary to apply computer graphics in various fields such as gaming, simulations, and data visualization.

Course Outcomes:

By the end of the course the students will be able to:
CO1: Understand the fundamental concepts and principles of computer graphics.
CO2: Develop proficiency in 2D and 3D graphics programming using appropriate tools and libraries.
CO3: Apply graphical techniques to solve real-world problems in areas such as gaming, simulation, and visualization.
CO4: Analyze and critique graphical algorithms and techniques for efficiency and effectiveness.
CO5: Demonstrate creativity and innovation in designing and implementing graphical applications.

Unit – I

Introduction to Computer Graphics: Overview of computer graphics and its applications, Historical development, Application areas e.g. entertainment, scientific visualization, user interfaces, Basics of raster and vector graphics, differences between raster and vector graphics, Representation and storage formats, Graphics pipeline and rendering process, stages of the graphics pipeline (modeling, transformation, rendering), concepts of rendering primitives, vertices, and fragments.

Graphics primitives: Coordinate systems, types of primitives (points, lines, polygons), Cartesian and homogeneous coordinate systems, transformation matrices and their applications

Unit – II

2D Graphics Programming: Pixel operations and drawing algorithms, color models (RGB, CMYK, HSL), pixel operations (blending, interpolation), Line drawing algorithms (Digital differential analyzer algorithm, Bresenham's line drawing algorithm), Circle drawing algorithms (Midpoint circle algorithm, Bresenham's circle drawing algorithm).

2D Transformations: translation, rotation, scaling in 2D space, mirror reflection basic and advance problems involving composite operations.

Windowing techniques: basic terminologies-window, viewport, clipping window, region codes.

Unit – III

Clipping techniques: Mathematics of point clipping, line clipping (midpoint subdivision method, Cohen-Sutherland line clipping algorithm), polygon clipping (Sutherland-Hodgman polygon clipping algorithm)

3D Graphics Programming: 3D transformations, Translation, Rotation, Scaling in 3D space, Homogeneous coordinates and transformations.

Projection techniques: Orthographic projection, Perspective projection and vanishing points

Unit – IV

Lighting and shading models: Phong reflection model, Gouraud and Phong shading techniques

Hidden surface removal algorithms: Z-buffer algorithm, Scan line algorithm for hidden surface removal

Texture mapping and image rendering: Texture coordinates and mapping techniques, Texture filtering and mipmapping.

Suggested Readings:

1. J. F. Hughes, A. van Dam, M. McGuire, D. F. Sklar, and J. D. Foley: Computer Graphics: Principles and Practice, Addison-Wesley.
2. D. Shreiner, G. Sellers, J. M. Kessenich, and B. M. Licea-Kane: OpenGL Programming Guide: The Official Guide to Learning OpenGL, Addison-Wesley.
3. D. Hearn and M. P. Baker: Computer Graphics with OpenGL, Pearson.
4. E. Angel and D. Shreiner: Interactive Computer Graphics: A Top-Down Approach with WebGL, Addison-Wesley.
5. F. Klawonn: Introduction to Computer Graphics: Using Java 2D and 3D, Springer.
6. Any other book covering the contents of the subject.

Note: Latest and additional good books may be suggested and added from time to time.

List of Programs

1. Implement a program to define and use basic graphics primitives such as points, lines, and polygons.
2. Implement a program to draw a line using the Digital Differential Analyzer (DDA) algorithm.
3. Create a program to draw a line using Bresenham's line drawing algorithm.
4. Write a program to draw a circle using the Midpoint Circle algorithm.
5. Implement a program to draw a circle using Bresenham's circle drawing algorithm.
6. Create a program to perform 2D transformations: translation, rotation, and scaling on a given set of points.
7. Write a program to perform mirror reflection of a 2D shape across the x-axis and y-axis.
8. Create a program to draw basic two-dimensional objects like rectangles, triangles, and polygons using inbuilt functions..
9. Write a program to implement the Cohen-Sutherland line clipping algorithm to clip a line within a rectangular clipping window.
10. Write a program to apply a shearing transformation to a 2D object (like a square or triangle) and display the original and transformed objects.
11. Write a program to apply various coloring techniques to 2D pictures.
12. Write a program to implement line clipping using the Cohen-Sutherland line clipping algorithm.
13. Write a program to perform 3D transformations: translation, rotation, and scaling on a set of 3D points.
14. Implement a program to demonstrate simple animations using transformations such as translation, rotation, and scaling.
15. Write a program to apply various coloring techniques to 3D object.

Any other programs assigned by the teachers.

Name of the Program	**4-Year Bachelor of Computer Applications**	**Program Code**	**-----**
Name of the Course	**Object oriented Programming using C++**	**Course Code**	**24BCA404MV01**
Hours/Week	**7 (1+6)**	**Credits (L:T:P)**	**1:0:3**
Max. Marks.	**Theory: 25 (20+5)** **Practical: 75 (50+25)**	**Time of end term examination**	**3 Hours**

Note: The examiner has to set nine questions in all by setting two questions from each Unit and Question No. 1 consisting of 4 parts (short-answer type questions) covering the entire syllabus. Student will be required to attempt five questions in all by selecting one question from each Unit and Question No. 1, which is compulsory.

Course Objectives:

The objective of this course is to inculcate in students the fundamentals of object oriented programming concepts and various features in C++, which help students to develop software using the object oriented paradigm of programming using C++.

Course Outcomes:

By the end of the course the students will be able to:
CO1: Identify the various object oriented concepts.
CO2: Program in C++ programming language.
CO3: Learn how to work with classes and pointers
CO4: Identify various types of inheritance.
CO5: Learn to apply various operations on files.

Unit – I

Introduction to Object Oriented Programming: Characteristics of OOPs: Objects, classes, encapsulation, data abstraction, inheritance, polymorphism, Dynamic binding and Message Passing,

C++ Programming Basics: Data Types, Variables, Operators, Expressions, Control Statements, Arrays, String, Structure, Function, Recursion

Unit – II

Introduction, Specifying a Class, Defining member Functions, C++ Program with Class, Nesting of Member functions, Private member functions, Memory Allocation for Objects, Static Data members, Static Member Functions, Arrays within a Class, Arrays of Objects, Objects as Function Arguments, Friendly Functions, Returning Objects.

Pointers: Declaration and initializing, Manipulation of pointers, Arrays of Pointers, Pointers to objects, this pointers. Constructors, Parameterized Constructors, Multiple Constructors in a class, Copy constructor, Destructors.

Unit – III

Operator Overloading: Defining Operator Overloading, Overloading Unary Operators, Overloading Binary Operators, Type Conversions.

Inheritance and Polymorphisms: Introduction, Defining Derived Classes, Single inheritance, Multiple inheritance, Hierarchical inheritance, Multilevel inheritance, Hybrid inheritance, Virtual Base Classes, Polymorphism, static and dynamic binding, Constructor in Derived Classes, Pointers to Derived Classes, Virtual Functions, Pure Virtual Functions.

Unit – IV

C++ Stream Classes, Unformatted I/O Operations, Formatted I/O operations, Classes for File Streams, Opening and Closing a File : open() and close() functions, Manipulators of File Pointers : seekg(), seekp(), tellg(), tellp() functions, Sequential Input and output Operations : put (), get(), write(), read() functions, Error handling File Operations : eof(), fail(), bad(), good().

Suggested Readings:

1. Balaguruswamy, E.: Object Oriented Programming with C++, TMH.
2. Lafore, Robert: OOP in Turbo C++, Galgotia.
3. Strostrup: The C++ Programming Language, Addison Wesley.
4. Parsa, N.R.: OOPS with C++ from the Foundation, Wiley India Pvt. Ltd.
5. Gaddis, Tonny : Starting out with C++, 3rd Edition, Wiley India Pvt. Ltd.
6. Gaddis, Tonny: Starting out with Object Oriented Programming in C++, 3rd Edition,Wiley India Pvt.

Ltd.

7. Al Steven: Al Steven's C++ Programming, 7th Edition, Wiley India Pvt. Ltd.
8. Any other book covering the contents of the subject.

Note: Latest and additional good books may be suggested and added from time to time.

List of Programs

1. Create a C++ program to define and use structures to store information about a student (name, roll number, marks, etc.).
2. Create a C++ program to implement control statements like if-else, switch-case, and loops
3. Implement a C++ program to demonstrate recursion by calculating factorial or Fibonacci series.
4. Create a C++ program to define a class called `Rectangle` with attributes `length` and `width`, and member functions to calculate area and perimeter.
5. Create a C++ program to illustrate memory allocation for objects using `new` and `delete` operators.
6. Write a C++ program to demonstrate the usage of static data members and static member functions within a class.
7. Implement a C++ program to use arrays within a class and perform operations on them.
8. Write a C++ program to showcase the usage of `this` pointer within member functions.
9. Implement a C++ program to define constructors, parameterized constructors, and multiple constructors within a class.
10. Create a C++ program to demonstrate the concept of copy constructor and destructor within a class.
11. Create a C++ program to demonstrate the working of operator overloading.
12. Write a C++ program to define base and derived classes, and demonstrate single and multiple inheritance.
13. Implement a C++ program to showcase multilevel inheritance.
14. Write a C++ program to illustrate polymorphism using virtual functions and dynamic binding.
15. Write a C++ program to perform formatted input/output operations using stream classes.
16. Create a C++ program to open, close, read from, and write to a file using file stream classes
17. Implement a C++ program to demonstrate the use of file manipulation functions like `seekg()`, `seekp()`, `tellg()`, and `tellp()`.
18. Write a C++ program to perform sequential input and output operations on files using `put()`, `get()`, `write()`, and `read()` functions.

Any other programs assigned by the teachers.