

Chaudhary Bansi Lal University, Bhiwani, Haryana

Scheme of Examination and Syllabus for Undergraduate Programme

Course: Computer Applications (Bachelor of Computer Applications Industry Integrated) (Under Multiple Entry-Exit, Internship and CBCS-
LOCF in accordance to NEP-2020) w.e.f. 2024-25 in Phased Manner

Semester	Discipline-Specific Courses (DSC)	Paper Code	Nomenclature of Paper	Credit	Minor (MIC) /Vocational (VOC)	Credit	Multidisciplinary Courses (MDC)	Ability Enhancement Courses (AEC)	Skill Enhancement Courses (SEC) /Internship/ Dissertation	Value - Added Courses (VAC)	Total Credits
I	CC-A1	24BCAI-N-101	Computer Fundamentals	4	24BCAI-N-104 Office Management Tools	2+1	24UN-BBA-MDC 101 Personal Finance@ 3 Credits	24UN-ENG--AEC101 Communication Skills in English @ 2 Credit	24BCAI- N -SEC -101 Fundamentals of Web @ 2+1 Credit	24UN-BOT-VAC101 Environmental Science @2 Credit	25
	CC-B1	24BCAI- N-102	Problem solving Technique using C Language	3+1							
	CC-C1	24BCAI- N-103	Computer Organization	4							
II	CC-A2	24BCAI- N- 201	Database Management system	3+1	24BCAI-N-204 Mathematical Foundation of Computer Science	3	24UN-BBA-MDC 201 Introduction to Entrepreneurship Development @ 3 Credit	24UN-ENG-AEC201 Business Communication@ 2 Credit	24BCAI- N -SEC -201 Front End Engineering @ 2+1 Credit	24UN-PSY-VAC101 Human Values and Ethics @ 2 Credit	25
	CC-B2	24BCAI- N- 202	Object Oriented Programming Using C++	3+1							
	CC-C2	24BCAI- N- 203	Software Engineering	4							
III	CC	24BCAI- N- 301	Java Programming	3+1	24BCAI-N-304 Data Analysis using Excel	2+1	24UN-BBA-MDC 301 Fundamentals of Indian Capital Market@ 3 Credit	24UN-ENG-AEC301 English Grammar and Usage@2 Credit	24BCAI- N - SEC -301 Internship@4 Credit	24UN-GEO-VAC303 Culture of India @ 2 Credit	25
	CC	24BCAI- N-302	Operating system	3							
	CC	24BCAI-N-303	Discrete Mathematics	4							
IV	CC	24BCAI- N- 401	Data Structure	3+1	24BCAI- N-404 Cloud Computing	3		24UN-ENG-AEC401 Vocabulary Building@ 2 Credit			17
	CC	24BCAI- N-402	Linux and Shell Programming	3+1							
	CC	24BCAI-N-403	Back End Engineering	3+1							
V	CC	24BCAI- N- 501	Programming in Python	3+1	24BCAI-N-504 Web Security	3			24BCAI- N -SEC -501 Internship@4 Credit		19
	CC	24BCAI-N-502 /	Computer Networks	4							
	CC	24BCAI-N-503	Artificial Intelligence	4							
VI	CC	24BCAI-N-601	Internship	20							20

Scheme of Examination for BCA (Industry Integrated)**Semester: 1st (w.e.f. Session 2024-25)**

S r · N o ·	Course/ Paper Code	Title of the Course	Mode of Learn ing	Course Type	Credit			Contact Hours per Week			Examination Scheme				Tot al Mar ks
					Theor y	Practical/ Seminar	Tot al	Theo ry	Practi cal /Semi nar	Tot al	End Semest er Exa m)	Intern al Assessm ent	Prac tical (Exte r nal)	Practi cal (Inter nal)	
1	24BCAI-N-101	Computer Fundamentals	CL/BL	CC	4		4	4		4	70	30			100
2	24BCAI- N-102	Problem solving Technique using C Language	CL/BL	CC	3	1	4	3	2	5	50	20	20	10	100
3	24BCAI- N-103	Computer Organization	CL/BL	CC	4	..	4	4		4	70	30	.		100
4	24BCAI-N-104	Office Management Tools	CL/BL	MIC	2	1	3	2	2	4	35	15	15.	5	70
5	24UN-BBA-MDC101	Personal Finance	CL/BL	MDC	3	..	3	3	..	3	50	20	.		70
6	24UN-ENG--AEC101	Communication Skills in English	CL	AEC	2	0	2	2		2	35	15			50
10	24BCAI- N - SEC -101	Fundamentals of Web	CL	SEC	2	1	3	3	2	5	35	15	15	5	70
11	24UN-BOT-VAC101	Environmental Science	CL	VAC	2	..	2	2	..	2	35	15			50
	Total						25			29					610

CC - Core Course; SEC-Skill Enhancement Course; AEC: Ability Enhancement Course MDC: Multidisciplinary Course VAC: Value Added course**MIC: Minor course**

24BCAI- N-101	Computer Fundamentals	4L:0T:0P	4 Credits
----------------------	----------------------------------	----------	-----------

Maximum Marks:100
External Examination: 70
Internal Assessment: 30
Max. Time: 3 Hrs

Course Objectives

Note: There shall be nine questions in all. Question no. 1 shall be compulsory, consisting of eight short answer type questions covering the entire syllabus. Two questions will be asked from each unit. Students will have to attempt one question from each unit. Each question shall carry equal marks.

Unit I

Evolution of Computers through generations, Characteristics of Computers, Strengths and Limitations of Computers, Classification of Computers, Functional Components of a Computer System, Applications of computers in Various Fields. Types of Software: System software, Application software, Utility Software, Shareware, Freeware, Firmware, Free Software. Memory Systems: Concept of bit, byte, word, nibble, storage locations and addresses, measuring units of storage capacity, access time, concept of memory hierarchy. Primary Memory - RAM, ROM, PROM, EPROM. Secondary Memory - Types of storage devices, Magnetic Tape, Hard Disk, Optical Disk, Flash Memory.

Unit II

I/O Ports of a Desk Top Computer, Device Controller, Device Driver. Input Devices: classification and use, keyboard, pointing devices - mouse, touch pad and track ball, joystick, magnetic stripes, scanner, digital camera, and microphone. Output Devices: speaker, monitor, printers: classification, laser, ink jet, dot-matrix. Plotter. Introduction to Operating System: Definition, Functions, Features of Operating System, Icon, Folder, File, Start Button, Task Bar, Status Buttons, Folders, Shortcuts, Recycle Bin, Desktop, My Computer, My Documents, Windows Explorer, Control Panel.

Unit III

Introduction to networks and internet, history, Internet, Intranet & Extranet, Working of Internet, Modes of Connecting to Internet. Electronic Mail: Introduction, advantages and disadvantages, User Ids, Passwords, e-mail addresses, message components, message composition, mailer features. Browsers and search engines.

Unit IV

Threats: Physical & non-physical threats, Virus, Worm, Trojan, Spyware, Keyloggers, Rootkits, Adware, Cookies, Phishing, Hacking, Cracking. Computer Security Fundamentals: Confidentiality, Integrity, Authentication, Non-Repudiation, Security Mechanisms, Security Awareness, Security Policy, anti-virus software & Firewalls, backup & recovery.

Course Outcomes:

A student will be able to:

1. Understanding the concept of input and output devices, memory of Computers
2. Learn the functional units and classify types of computers, how they process information and how individual computers interact with other computing systems and devices.
3. Understand an operating system and its working, and solve common problems related to operating systems
4. Learn about basic threats on internet.
5. Study to use the Internet safely, legally, and responsibly

Suggested Readings:

1. Byron Gottfried, Schaum's Outline of Programming with C, McGraw-Hill.
2. R.S. Salaria, Programming for Problem Solving, Khanna Publishing House
3. E. Balaguruswamy, Programming in ANSI C, Tata McGraw-Hill.
4. Brian W. Kernighan and Dennis M. Ritchie, The C Programming Language, Prentice Hall of India.

24BCAI- N-102	Problem solving Technique using C Language	3L:0T:0P	3 Credits
----------------------	---	----------	-----------

Maximum Marks:70
External Examination: 50
Internal Assessment: 20
Max. Time: 3 Hrs

Course Objectives

- To understand basic concepts of C
- To enable the students to solve the problems efficiently using different problem-solving methodologies and implement them in C.

Note: There shall be nine questions in all. Question no. 1 shall be compulsory, consisting of eight short answer type questions covering the entire syllabus. Two questions will be asked from each unit. Students will have to attempt one question from each unit. Each question shall carry equal marks.

Unit I

Introduction to programming in C: History, Flowchart, Algorithm and pseudocode, Structure and execution flow of C Program, Documentation of programs, Character Set, Constants and Variables, Identifiers and Keywords, Data Types, Assignment Statement, Input/Output: Basic and Formatted I/O Function, Operators & Expression: Arithmetic, Relational, Logical, Bitwise, Unary, Assignment and Special Operators Operator Hierarchy; Arithmetic Expressions, Evaluation of Arithmetic Expression, Type Casting and Conversion.

Unit II

Decision making and Loop Statements: if statement, if-else statement, nested if statement, else-if ladder, switch and break statement, goto statement, Conditional Operators, Short-circuit in C, Looping Statements: for, while, and do-while loop, jumps in loops, Scope and life of a variable, Storage Classes: auto storage class, extern storage class, static storage class, register storage class.

Unit III

Arrays: One Dimensional arrays - Declaration, Initialization and Memory representation; Two Dimensional arrays -Declaration, Initialization and Memory representation.

Functions: definition, prototype, function call, passing arguments to a function: call by value; call by reference, recursive functions.

Strings: Declaration and Initialization, String I/O, Array of Strings, String Manipulation Functions: String Length, Copy, Compare, Concatenate etc., Search for a Substring.

Unit IV

Pointers in C: Declaring and initializing pointers, accessing address and value of variables using pointers; Pointers and Arrays.

User defined data types: Structures - Definition, Advantages of Structure, declaring structure variables, accessing structure members, Structure members initialization, Array of Structures; Unions - Union definition; difference between Structure and Union.

Course Outcomes

A student will be able to:

- Learn to develop simple algorithms and flow charts to solve a problem.
- Develop problem solving skills coupled with top-down design principles.
- Learn about the strategies of writing efficient and well-structured computer algorithms/programs.
- Develop the skills for formulating iterative solutions to a problem.
- Learn array processing algorithms coupled with iterative methods.
- Learn text and string processing efficient algorithms.
- Learn searching techniques and use of pointers.
- Understand and apply recursive techniques in programming.

Suggested Readings:

1. Byron Gottfried, Schaum's Outline of Programming with C, McGraw-Hill.
2. R.S. Salaria, Programming for Problem Solving, Khanna Publishing House
3. E. Balaguruswamy, Programming in ANSI C, Tata McGraw-Hill.
4. Brian W. Kernighan and Dennis M. Ritchie, The C Programming Language, Prentice Hall of India.

24BCAI- N-103	Computer Organization	4L:0T:0P	4 Credits
----------------------	------------------------------	----------	-----------

Maximum Marks:100
External Examination: 70
Internal Assessment: 30
Max. Time: 3 Hrs

Course Objectives

- To understand the basic organization of Digital Computer.
- To understand the working of different parts of the system.
- To understand data flow between different units.

Note: There shall be nine questions in all. Question no. 1 shall be compulsory, consisting of eight short answer type questions covering the entire syllabus. Two questions will be asked from each unit. Students will have to attempt one question from each unit. Each question shall carry equal marks.

Unit I

Digital Computer: Block diagram, its parts and working, Data flow in a digital computer, Number System, Types of Number Systems, Conversion from one number system to another, Binary Arithmetic, Data Types, Complements, Fixed-Point Representation, Floating-Point Representation.

Unit II

Basic Digital Devices: Binary Codes, Error Detection Codes, Logic Gates, Boolean Algebra, Map Simplification, Combinational Circuits, Flip-flops.

Register Transfer and Micro operations: Register Transfer Language, Register Transfer, Bus & Memory Transfer, Arithmetic Microoperations, Logic Micro Operations, Shift Micro Operation.

Unit III

Basic Computer Organization: Instruction codes, Computer Registers, Computer Instructions, Timing & Control, Instruction Cycles, Memory Reference Instruction, Input - Output & Interrupts, Complete Computer Description & Design of Basic Computer.

Processor and Control Unit: Hardwired vs. Microprogrammed Control Unit, General Register Organization, Stack Organization

Unit IV

Instruction Format, Data Transfer & Manipulation, Program Control, RISC, CISC, Pipelining and Parallelism

Memory and I/O Systems: Peripheral Devices, I/O Interface, Data Transfer Schemes, Program Control, Interrupt, DMA Transfer, I/O Processor. Memory Hierarchy, Processor vs. Memory Speed, High-Speed Memories, Cache Memory, Associative Memory, Interleave, Virtual Memory, Memory Management.

Course Outcomes

A student will be able to:

- To make students understand the basic structure, operation and characteristics of digital computers.
- To familiarize the students with arithmetic and logic units as well as the concept of pipelining.
- To familiarize the students with hierarchical memory systems including cache memories and virtual memory.
- To make students know the different ways of communicating with I/O devices and standard I/O interfaces.

Suggested Readings:

1. Computer System Architecture, M. Morris Mano, 3rd Edition, Prentice Hall.
2. Computer Organization and Design, David A. Patterson and John L. Hennessey, Fifth edition, Morgan Kaufman / Elsevier, 2014. 30
3. Computer Architecture: A Quantitative Approach, John L. Hennessy, David A. Patterson, 4th Edition.
4. Computer Organization and Architecture, William Stallings, Prentice Hall.

24BCAI-N-104	Office Management Tools	2L:0T:0P	2 Credits
---------------------	--------------------------------	----------	-----------

Maximum Marks:50
External Examination: 35
Internal Assessment: 15
Max. Time: 3 Hrs

Course Objectives

Note: There shall be nine questions in all. Question no. 1 shall be compulsory, consisting of eight short answer type questions covering the entire syllabus. Two questions will be asked from each unit. Students will have to attempt one question from each unit. Each question shall carry equal marks.

Unit I

Definition, Functions, Types of Operating System, Basics of Popular Operating Systems, The User Interface Exploring Computer, Icons, taskbar, desktop. Using Menu and Menu-selection, managing files and folders, Control panel - display properties, add/remove software and hardware, Common utilities.

Unit II

Introduction to Word Processing, Menus, Creating, Editing & Formatting Document, Spell Checking, Printing, Views, Tables, Word Art, Mail Merge, Macros, Inserting hyperlinks, Searching for text, Modifying page setup, Applying document themes. Applying document style sets, Inserting headers and footers, Table

Unit III

Spread Sheet: Elements of Electronics Spread Sheet, Applications, Creating and Opening of Spread Sheet, Menus, Manipulation of cells Enter texts numbers and dates, Cell Height and Widths, Copying of cells, Mathematical, Statistical and Financial function, Drawing different types of charts, Sort and Filter Data. Conditional formatting, Data validation, pivot table, chart

Unit IV

Presentation Software: Creating, Modifying and enhancing a presentation, Type of presentation views, Using sound, Animation Working with Objects. Printing,

Suggested Readings:

1. Computer Application in Business –Dr.S.V.Srinivasa Vallabhan, Sultan Chand and Sons, New Delhi
2. MS-Office and Internet by Alexis Leon

3. Computer Application in Business – K.Mohan Kumar, Vijay Nicole imprints Private Limited Dr.S.Rajkumar –Chennai
4. Computer Basics – V.Rajaraman – PHI.

24BCAI- N -SEC -101	Fundamentals of Web	2L:0T:0P	2 Credits
---------------------	---------------------	----------	-----------

Maximum Marks:50
External Examination: 35
Internal Assessment: 15
Max. Time: 3 Hrs

Course Objectives

- To learn HTML Tags and make structure of web pages.
- To apply CSS properties to look rich and attractive.
- To learn how to work with Flex and Grid.
- To use Bootstrap to make it responsive.
- To develop a real website by using a commercial approach.

Note: There shall be nine questions in all. Question no. 1 shall be compulsory, consisting of eight short answer type questions covering the entire syllabus. Two questions will be asked from each unit. Students will have to attempt one question from each unit. Each question shall carry equal marks.

Unit I

HTML Anatomy: HTML Structure (<html> tag, Head, Page Title, Body), Headings, Div's, Attributes, Displaying Text, Line Breaks, Unordered Lists, Ordered Lists, Images, Anchor Tag

Tables: Create a Table, Table Rows, Table Data, Table Headings, Table Borders, Spanning Columns & Rows, Table Body

HTML Forms: Input (Text, Email, Password, Range, Checkbox, Radio button, Date & Time), Label, Select Tag.

Unit II

Introduction to CSS: Inline Styles, <style> Tag, the .css file, Linking the CSS File, Tag Name, Class Name, Multiple Classes, ID Name, Classes and IDs, Specificity, Chaining Selectors, Nested Elements, Chaining and Specificity, Multiple Selectors

Box Model: Height and Width, Borders, Border Radius, Padding, Margins, Auto Margin Collapse, Minimum and Maximum Height and Width, Overflow, Resetting Defaults, Visibility, Flow of HTML, Positions (Relative, Absolute, Fixed, Z-Index), Inline Display, Block Display, Inline-Block Display, Float, Clear, Flow of HTML, Positions (Relative, Absolute, Fixed, Z-Index), Inline Display, Block Display, Inline-Block Display, Float, Clear

Font Styling: Text Alignment, Color, Opacity, Background Image Word Spacing, Letter Spacing, Line Height Anatomy, Linking Fonts, Font-Face

Sizing Elements: Relative Measurements, Em, Rem, Percentages (Height, Width, Padding, Margin), Scaling Images and Video

Unit III

Flexbox: display, flex, inline-flex, justify-content, align-items, flex-grow, flex-shrink, flex-basis, flex, flex-wrap, Align-content, flex-direction, flex-flow, Nested Flexboxes,

Grids: Grid Template, Grid Contained, Grid Items, Fraction, Repeat, minmax, Grid Gap.

Media Queries - Responsive Web Design, Media Queries, Range, Dots Per Inch (DPI), And Operator, Comma Separated List, Breakpoints

Unit IV

BootStrap: Intro, Need of BootStrap, Download Bootstrap, CDN, Create First Web Page with Bootstrap, Add the HTML5 Doctype, Bootstrap is Mobile First, Containers, Basic Bootstrap Pages, Bootstrap Grids, Bootstrap Grid System, Grid Classes, Basic Structure of a Bootstrap Grid, Equal Columns Unequal Columns, Bootstrap Text / Typography, Default Settings Bootstrap vs. Browser Defaults, Bootstrap Tag Elements, Contextual Colors and Background, Typography Classes, Bootstrap Table, Rounded Corners, Circle, Thumbnail, Responsive Images, Image Gallery, Responsive Embeds.

Course Outcomes

A student will be able to:

- Understand the design principles of Web pages and how they are created.
- Students will be able to apply HTML Tags to create the structure of Web Pages.
- Students will be able to apply CSS properties to make the page rich in look and feel.
- Will be able to use Bootstrap framework for making pages responsive.

Suggested Readings:

1. Head First HTML and CSS – Elisabeth Robson & Eric Freeman
2. Web Design Playground: HTML & CSS the Interactive Way – Paul McFedries
3. Bootstrap 5 Foundations – Daniel Foreman

Lab-I (Based on 24BCAI- N-102 Problem solving Technique using C Language)	0L:0T:2P	1 Credits
--	----------	-----------

Maximum Marks:30
External Examination: 20
Internal Assessment: 10

Course Outcomes: By the end of the course the student will be able to:

- Develop problem solving skills coupled with top-down design principles.
- Develop programs in C by using various conditional and looping structures.
- Develop programs in C by using various conditional and looping structures.
- Work on some real projects used in Industry nowadays.

List of Practical:

Students are advised to do laboratory/practical practice not limited to, but including the following types of problems:

1. To read radius of a circle and to find area and circumference
2. To read three numbers and find the biggest of three
3. To check whether the number is prime or not
4. To read a number, find the sum of the digits, reverse the number and check it for palindrome
5. To read numbers from keyboard continuously till the user presses 999 and to find the sum of only positive numbers
6. To read percentage of marks and to display appropriate message (Demonstration of else-if ladder)
7. To find the roots of quadratic equation
8. To read marks scored by n students and find the average of marks (Demonstration of single dimensional array)
9. To remove Duplicate Element in a single dimensional Array
10. To perform addition and subtraction of Matrices
11. To find factorial of a number
12. To generate Fibonacci series
13. To remove Duplicate Element in a single dimensional Array
14. To find the length of a string without using built in function
15. To demonstrate string functions
16. To read, display and add two m x n matrices using functions
17. To read a string and to find the number of alphabets, digits, vowels, consonants, spaces and special characters
18. To Swap Two Numbers using Pointers
19. To demonstrate student structure to read & display records of n students
20. To demonstrate the difference between structure & union.

Lab-2 (Based on 24BCAI- N -SEC -101 Fundamentals of Web)	0L:0T:2P	1 Credits
---	----------	-----------

Maximum Marks:20
External Examination: 15
Internal Assessment: 5

Course Outcomes: By the end of the course the student will be able to:

- Use tags of HTML practically to build a web page.
- Use CSS for styling the web pages.
- Use to design forms and Tables for a web page.
- Use Flex and Grid for layout of a web page.
- Develop a website with all the HTML, CSS and Bootstrap features.

List of Practical:

1. Build a resume webpage by using HTML elements.
2. Nested List- create a nested list as per the task defined.
3. HTML Table- create a table as per the task defined.
4. HTML Form – create an HTML form as the task given.
5. CSS Typography- design an article using the CSS. Apply the CSS properties as defined in the user stories.
6. CSS Display and Positioning: Change the display or position property of a paragraph element to show that they are rendered side by side.
7. Horizontal Dropdown Menu: Create a dropdown navigation bar using HTML and CSS.
8. CSS display and positioning: Float and Clear
9. Flex Cards: Create a responsive card layout by applying CSS properties as mentioned in user stories.
10. CSS Grid: Create a grid layout as per the user stories.
11. Webpage layout using Flexbox: create a responsive webpage layout using flexbox.
12. Bootstrap Carousel: Make an image carousel using bootstrap and make changes in it using CSS as per the given task.

Lab-3 (Based on 24BCAI-N-104 Office Management Tools)	0L:0T:2P	1 Credit
--	----------	----------

Maximum Marks:20
External Examination: 15
Internal Assessment: 5

Course Outcomes: By the end of the course the student will be able to:

- To create and format documents, apply styles, and use different paragraph options for professional-looking content.
- To acquire essential formulas and functions, sort and filter data, and create charts to visualize data effectively.
- To develop skills in creating engaging presentations, utilizing multimedia, slide transitions, and animations.
- To practice using tables, lists, and headers/footers to organize information systematically.
- To utilize ChatGPT responsibly with proper prompts for practical applications.
- To acquire practical skills and knowledge that will enhance their productivity, communication, and overall proficiency in using computers and popular software applications.

List of Practical:

1. Creating and Formatting Documents: Students will learn how to create new documents, format text, change fonts, apply styles, and use different paragraph formatting options.
2. Headers and Footers: Students will explore adding headers and footers to documents, inserting page numbers, and including relevant information in headers/footers.
3. Tables and Lists: Participants will practice creating tables, formatting them, and using lists (bulleted and numbered) to organize content effectively.
4. Inserting Images and Objects: This practical will involve inserting images, shapes, and other objects into the document and managing their placement and alignment.
5. Basic Formulas and Functions: Participants will learn essential formulas (e.g., SUM, AVERAGE, MAX, MIN) and functions to perform calculations in a spreadsheet.
6. Formatting and Conditional Formatting: Students will practice formatting cells, applying cell styles, and using conditional formatting to highlight data based on specific conditions.
7. Creating Charts: This practical will involve creating various types of charts (e.g., bar, line, pie) to visualize data in a spreadsheet.
8. Data Sorting and Filtering: Participants will learn how to sort data in ascending/descending order and use filtering to display specific data subsets.
9. Data Validation: Students will explore data validation to restrict cell entries and ensure data consistency in the spreadsheet.
10. Creating Slides: Students will learn how to create new slides, select slide layouts, and add content (text, images, shapes) to slides.
11. Slide Transitions and Animations: Participants will practice applying slide transitions and animations to make their presentations visually appealing.
12. Inserting Multimedia: Students will learn to embed videos, audio, and hyperlinks into their presentations.
13. Slide Master: Participants will explore slide master features to create consistent design elements and layouts across all slides.

14.

Scheme of Examination for B.CA. (Industry Integrated)

Semester: 2nd (w.e.f. Session 2023-24)

S r. N o.	Course/ Paper Code	Title of the Course	Mode of Learni ng	Course Type	Cre dit			Contact Hours per Week			Examination Scheme			Prac tic al (Int ern al)	Tot al Mar ks
					Th eor y	Practical / Semin ar	Total	The ory	Practi cal /Semin ar	Tota l	End Sem ester E xa m	Intern al Assessm ent	Practi cal (Exter nal)		
1	24BCAI-N-201	Database Management system	CL/ BL	CC	3	1	4	3	2	5	50	20	20	10	100
2	24BCAI-N-202	Object Oriented Programming Using C++	CL/ BL	CC	3	1	4	3	2	5	50	20	20	10	100
3	24BCAI-N-203	Software Engineering	CL/ BL	CC	4	..	4	4	..	4	70	30	..	..	100
4	24BCAI-N-204	Mathematical Foundation of Computer Science	CL/ BL	MIC	3	..	3	3	..	3	50	20	..	..	70
5	24UN-BBA-MDC201	Introduction to Entrepreneur ship Development	CL/ BL	MDC	3	..	3	3	..	3	50	20	..	..	70
6	24UN-ENG-AEC201	Business Communication	CL/BL	AEC	2	..	2	2	..	2	35	15	..	..	50
7	24BCAI-N-SEC-201	Front End Engineering	CL/ BL	SEC	2	1	3	2	2	4	35	15	15	5	70
8	24UN-PSY-VAC101	Human Values and Ethics	CL/ BL	VAC	2	..	2	2	..	2	35	15	..	..	50
	Total						25			28					610

24BCAI-N-202	Object Oriented Programming Using C++	3L:0T:0P	3 Credits
--------------	--	----------	-----------

Maximum Marks:70
External Examination: 50
Internal Assessment: 20
Max. Time: 3 Hrs

Course Objectives

- To learn about OOPS concepts and linking C++ as a powerful OOPS language.

Note: There shall be nine questions in all. Question no. 1 shall be compulsory, consisting of eight short answer type questions covering the entire syllabus. Two questions will be asked from each unit. Students will have to attempt one question from each unit. Each question shall carry equal marks.

Unit I

Introduction to Object Oriented Programming - Basic concept of OOP, Comparison of Procedural Programming and OOP, Benefits of OOP, C++ compilation, Abstraction, Encapsulation, Inheritance, Polymorphism, Review of C, Difference between C and C++ - cin, cout, new, delete, operators. Program Structure of the C++ program, Namespace.

Warm-up with C++ Syntax: Decision and Control Structures: if statement, if-else statement, switch statement, Loop: while, do-while, for; Jump statements: break, continue, go to.

Functions - main() function, components of function: prototype, function call, definition, parameter; passing arguments; types of function, inline function, function overloading.

Unit II

Introduction to Classes and Objects - Classes in C++, class declaration, declaring objects, Defining Member functions, Inline member function, Array of objects, Objects as function argument, Static data member and member function, Friend function and friend class.

Constructors and Destructors - Constructors, Instantiation of objects, Default constructor, Parameterized constructor, Copy constructor and its use, Destructors, Constraints on constructors and destructors, Dynamic initialization of objects.

Unit III

Operator Overloading - Overloading unary operators: Operator keyword, arguments and return value; overloading unary and binary operators: arithmetic operators, manipulation of strings using operators; Type conversions.

Inheritance - Derived class and base class: Defining a derived class, Accessing the base class member, Inheritance: multilevel, multiple, hierarchical, hybrid; Virtual base class, Abstract class

Virtual Functions and Polymorphism - Virtual functions, pure virtual functions; Polymorphism, Categorization of polymorphism techniques: Compile time polymorphism, Runtime polymorphism

Unit IV

File Handling - File classes, Opening and Closing a file, File modes, Manipulation of file pointers, Functions for I/O operations.

Exception handling- Try, throw, and catch.

Standard Template Library- Fundamental idea about string, iterators, hashes, iostreams and other types.

Course Outcomes

A student will be able to:

- **To** understand basic concept of C++
- To acquire the knowledge of C++ operators, hierarchy and precedence and various control structures.
- To learn to use arrays and string in C++ programs.
- To get familiar with OOPS concepts with C++.

Suggested Readings:

1. Bjarne Stroustrup, The C++ Programming Language, Pearson Education
2. Balaguruswami, E., Object Oriented Programming In C++, Tata McGraw-Hill.
3. Richard Johnson, An Introduction to Object-Oriented Application Development, Thomson Learning.

Lab- I: (Based on 24BCAI-N-202) Object Oriented Programming Using C++	0L:0T:2P	1 Credits
--	----------	-----------

Maximum Marks:30
External Examination: 20
Internal Assessment: 10
Max. Time: 3 Hrs

Course Outcomes: By the end of the course the student will be able to:

- Develop problem solving skills coupled with top-down design principles.
- Develop programs in C++ by using various conditional and looping structures.
- Develop programs in C++ using Class and Objects Concepts.
- Work on some real projects used in Industry nowadays.

List of Practical:

1. Raising a number n to a power p is the same as multiplying n by itself p times. Write a function called power() that takes a double value for n and an int value for p, and returns the result as double value. Use a default argument of 2 for p, so that if this argument is omitted, the number will be squared. Write a main () function that gets values from the user to test this function.
2. A point on the two dimensional plane can be represented by two numbers: an X coordinate and a Y coordinate. For example, (4,5) represents a point 4 units to the right of the origin along the X axis and 5 units up the Y axis. The sum of two points can be defined as a new point whose X coordinate is the sum of the X coordinates of the points and whose Y coordinate is the sum of their Y coordinates. Write a program that uses a structure called point to model a point. Define three points, and have the user input values to two of them. Then set the third point equal to the sum of the other two, and display the value of the new point. Interaction with the program might look like this:

Enter coordinates for P1: 3 4

Enter coordinates for P2: 5 7

Coordinates of P1 + P2 are : 8, 11

3. Create the equivalent of a four function calculator. The program should request the user to enter a number, an operator, and another number. It should then carry out the specified arithmetical operation: adding, subtracting, multiplying, or dividing the two numbers. (It should use a switch statement to select the operation). Finally it should display the result. When it finishes the calculation, the program should ask if the user wants to do another calculation. The response can be 'Y' or 'N'. Some sample interaction with the program might look like this.

Enter first number, operator, and second number: 10/ 3

Answer = 3.333333

Do another (Y/ N)? Y

Enter first number, operator, second number 12 + 100

Answer = 112

Do another (Y/ N) ? N

4. A phone number, such as (212) 767-8900, can be thought of as having three parts: the area code (212), the exchange (767) and the number (8900). Write a program that uses a structure to store these three parts of a phone number separately. Call the structure phone. Create two structure variables of type phone. Initialize one, and have the user input a number for the other one. Then display both numbers. The interchange might look like this:
 - Enter your area code, exchange, and number: 415 555 1212
 - My number is (212) 767-8900
 - Your number is (415) 555-1212
5. Create two classes DM and DB which store the value of distances. DM stores distances in metres and centimeters and DB in feet and inches. Write a program that can read values for the class objects and add one object of DM with another object of DB. Use a friend function to carry out the addition operation. The object that stores the results maybe a DM object or DB objects, depending on the units in which the results are required. The display should be in the format of feet and inches or metres and centimetres depending on the object on display.
6. Create a class rational which represents a numerical value by two double values- NUMERATOR and DENOMINATOR. Include the following public member Functions:
constructor with no arguments (default).
constructor with two arguments.
void reduce() that reduces the rational number by eliminating the highest common factor between the numerator and denominator.
Overload + operator to add two rational number.
Overload >> operator to enable input through cin.
Overload << operator to enable output through cout.
Write a main () to test all the functions in the class.
7. Consider the following class definition

```
class father {  
protected : int age;  
public;  
father (int x) {age = x;}  
virtual void iam ( )  
{ cout<< "I AM THE FATHER, my age is : "<< age<< endl;}  
};
```

Derive the two classes son and daughter from the above class and for each, define iam () to write our similar but appropriate messages. You should also define suitable constructors for these classes. Now, write a main () that creates objects of the three classes and then calls iam () for them. Declare pointer to father. Successively, assign addresses of objects of the two derived classes to this pointer and in each case, call iam () through the pointer to demonstrate polymorphism in action.
8. Write a program that creates a binary file by reading the data for the students from the terminal. The data of each student consist of roll no., name (a string of 30 or lesser no. of characters) and marks.
9. A hospital wants to create a database regarding its indoor patients. The information to store include
Name of the patient
Date of admission
Disease

Date of discharge

Create a structure to store the date (year, month and date as its members). Create a base class to store the above information. The member function should include functions to enter information and display a list of all the patients in the database. Create a derived class to store the age of the patients. List the information about all the to store the age of the patients. List the information about all the pediatric patients (less than twelve years in age).

10. Make a class Employee with a name and salary. Make a class Manager inherit from Employee. Add an instance variable, named department, of type string. Supply a method to String that prints the manager's name, department and salary. Make a class Executive inherits from Manager. Supply a method to String that prints the string "Executive" followed by the information stored in the Manager superclass object. Supply a test program that tests these classes and methods.
11. Imagine a tollbooth with a class called toll Booth. The two data items are a type unsigned int to hold the total number of cars, and a type double to hold the total amount of money collected. A constructor initializes both these to 0. A member function called payingCar () increments the car total and adds 0.50 to the cash total. Another function, called nopayCar (), increments the car total but adds nothing to the cash total. Finally, a member function called displays the two totals. Include a program to test this class. This program should allow the user to push one key to count a paying car, and another to count a nonpaying car.
12. Write a function called reversit () that reverses a string (an array of char). Use a for loop that swaps the first and last characters, then the second and next to last characters and so on. The string should be passed to reversit () as an argument. Write a program to exercise reversit (). The program should get a string from the user, call reversit (), and print out the result. Use an input method that allows embedded blanks. Test the program with Napoleon's famous phrase, "Able was I ere I saw Elba".
13. Create some objects of the string class, and put them in a Deque. Some at the head of the Deque and some at the tail. Display the contents of the Deque using the forEach () function and a user written display function. Then search the Deque for a particular string, using the first That () function and display any strings that match. Finally remove all the items from the Deque using the getLeft () function and display each item. Notice the order in which the items are displayed: Using getLeft (), those inserted on the left (head) of the Deque are removed in "last in first out" order while those put on the right side are removed in "first in first out" order. The opposite would be true if getRight () were used.
14. Assume that a bank maintains two kinds of accounts for customers, one called as savings account and the other as current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed.
Create a class account that stores customer name, account number and type of account. From this derive the classes
cur_acct and sav_acct to make them more specific to their requirements. Include necessary member functions in order to achieve the following tasks:
Accept deposit from a customer and update the balance.
Display the balance.
Compute and deposit interest.
Permit withdrawal and update the balance.
Check for the minimum balance, impose penalty, necessary and update the balance.
Do not use any constructors. Use member functions to initialize the class members.
15. Create a base class called shape. Use this class to store two double type values that could be used to compute the area of figures. Derive two specific classes called triangle and rectangle from the base shape. Add to the base class, a member function get_data () to initialize baseclass data members and another member function display_area () to compute and display the area of figures. Make display_area () as a virtual function and redefine this function in the derived classes to suit their requirements. Using these

three classes, design a program that will accept dimensions of a triangle or a rectangle interactively and display the area.

Remember the two values given as input will be treated as lengths of two sides in the case of rectangles and as base and height in the case of triangles and used as follows:

Area of rectangle = $x * y$

Area of triangle = $\frac{1}{2} * x * y$

24BCAI- N - SEC -201	Front End Engineering	2L:0T:0P	2 Credits
-------------------------	-----------------------	----------	-----------

Maximum Marks:50
External Examination: 35
Internal Assessment: 15
Max. Time: 3 Hrs

Course Objectives

- To understand advanced tools of web technologies and develop front end applications using JavaScript.

Note: *There shall be nine questions in all. Question no. 1 shall be compulsory, consisting of eight short answer type questions covering the entire syllabus. Two questions will be asked from each unit. Students will have to attempt one question from each unit. Each question shall carry equal marks.*

Unit I

Introduction to Web, Static and dynamic Pages, Introduction and fast revision to html tags and their usages, HTML Tags.

CSS: Selectors, Specificity, Selector Chaining, Box Model- Height & Width, Margin, Padding, Borders, Margin Collapse, Overflow, visibility.

JavaScript: Introduction and its advantages and disadvantages, Internal and external JS, Introduction to Variables and basics programming concepts in JavaScript, Scope of variables, conditional statements, loops Arrays accessing array elements using different methods.

Unit II

JSON and localStorage : Functions: Type of function and their scope anonymous functions and arrow functions

Arrays Revisited: Iterators, foreach, filter, map and creating custom iterators.

Objects: Introduction to Object, managing Objects.

JSON and localStorage: Nested JSON and Different JSON methods, Data transfer/exchange methods using JSON, Advantages and drawbacks of Using localStorage and other possible ways, JSON handling different properties, Different ways of storing and retrieving the JSON

Unit III

DOM: Introduction to DOM, Accessing Elements of HTML using Dom, Various functions for handling DOM, Event Handling, ways of handling the events, Dynamic handling of DOM Elements, Dynamic Creation of different elements of DOM

Async Programming Introduction and its usage, JavaScript methods to perform background tasks, Timeout and Interval,
Async and await, Promises and handling promises, Callback and exceptions.

Unit IV

GIT Version Control System (Cycle, Clone, GIT log, diff, commit, push, pull, Branches, and contributions)

Introduction to React, FED in JavaScript and React, developing code in VS Code and introduction to webpack, Managing webpack and other components, Components in React

Course Outcomes

A student will be able to:

- Students will be able to understand and apply HTML5, CSS3, JavaScript to develop front End.
- Students will have sound practical know-how of using arrays, objects, JSON and local Storage.
- They will be able to manipulate the front-end using DOM manipulation.
- Understand the concept of Front-end development, API's, Async Programming.
- Students will be able to develop Single Page Applications using the component-based approach.

Suggested Readings:

1. Head First HTML and CSS – Elisabeth Robson & Eric Freeman
2. Web Design Playground: HTML & CSS the Interactive Way – Paul McFedries
3. Bootstrap 5 Foundations – Daniel Foreman

Lab-2 :Front End Engineering Lab (Based on 24BCAI- N - SEC -201)	0L:0T:4P	2 Credits
--	----------	-----------

Maximum Marks:20
External Examination: 15
Internal Assessment: 5

Course Outcomes: By the end of the course the student will be able to:

- To practically apply HTML5, CSS3, JavaScript to develop front End.
- Practically know-how of using arrays, objects, JSON and local Storage.
- To manipulate the front-end using DOM manipulation.
- Use the concept of Front-end development, API's, Async Programming.
- To develop Single Page Applications using the component-based approach of React.

List of Practical:

1. Book Review Blog, you have to create a simple blog page to display the contents of blog with pictures, you have to complete the required tasks.
2. Html Table- basic table generation to get into the understanding, you have to complete the task as defined
3. College Admission Form, create a basic admission form using html forms to capture the admission information from the user.
4. Share Your Recipe, your task is to display the contents of the Food Recipe with various components like ingredients, preparations time.
5. Chess Board Write a program in js that creates a chessboard that represent a grid using
 to separate lines. At each position of the grid there is either a " * " or a "#".
6. Calculate Bill, You have been given 2 arrays of objects (menu and categories) and another json to have bill, from these various object we need to calculate the Bill and generate a JSON.
7. To Do App – Basic Application, you have to build a basic to do app to manage the tasks and complete the user story as defined.
8. Discussion App - D1 Basic Discussion app to include questions and responses to all the questions, application will display questions on the left pane and their response on right pane on the click of question, till then displays the new question form to add new question.
9. Build Pomodoro Clock – Pomodoro is a time management technique developed in the 1980s which uses a timer to break down work into intervals, traditionally 25 minutes in length, separated by short breaks. In this assignment, you'll know how to create such a timer in the browser with JavaScript.
10. Web API, in this assignment, you need Build a basic app to fetch the records using API's with the help of AJAX. This assignment uses the GitHubapi to access data using API.
11. Build a basic app to compile coding questions using API's with the help of AJAX. In this assignment you need to create a compiler app using the codequotientapi and follow the tasks.

24BCAI-N-201	Database Management System	3L:0T:0P	3 Credits
--------------	-----------------------------------	----------	-----------

Maximum Marks:70
External Examination: 50
Internal Assessment: 20
Max. Time: 3 Hrs

Course Objectives

- Understand the relational database design principles. Familiar with the basic issues of transaction processing and concurrency control.
- Familiar with SQL queries, database storage structures and access techniques.

Note: There shall be nine questions in all. Question no. 1 shall be compulsory, consisting of eight short answer type questions covering the entire syllabus. Two questions will be asked from each unit. Students will have to attempt one question from each unit. Each question shall carry equal marks.

UNIT – I

Basic Concepts – Data, Information, Records and files. Traditional file – based Systems-File Based Approach-Limitations of File Based Approach, Database Approach-Characteristics of Database Approach, Database Management System (DBMS), Components of DBMS Environment, DBMS Functions and Components, Advantages and Disadvantages of DBMS, Roles in the Database Environment - Data and Database Administrator, Database Designers, Applications Developers and Users.

Data Models: Records- based Data Models, Object-based Data Models, Physical Data Models and Conceptual Modeling.

UNIT – II

Database System Architecture – Three Levels of Architecture, External, Conceptual and Internal Levels, Schemas, Mappings and Instances, Data Independence – Logical and Physical Data Independence, Normalization with various normal forms.

Entity-Relationship Model – Entity Types, Entity Sets, Attributes Relationship Types, Relationship Instances and ER Diagrams.

Relational Data Model - Terminology in Relational Data Structure, Relations, Properties of Relations, Keys, Domains, Integrity Constraints over Relations, Base Tables and Views, Basic Concepts of Hierarchical and Network Data Model.

UNIT – III

SQL fundamentals, data types, DDL, DML, DCL, Data Constraints, working with Tables, Defining different constraints on the table, Defining Integrity Constraints in the ALTER TABLE Command, Select Command, Logical Operator, Range Searching, Pattern Matching, Oracle Function, Grouping data from Tables in SQL, Manipulation Data in SQL.

UNIT – IV

Joining Multiple Tables (Equi Joins), Joining a Table to itself (self Joins), Sub queries Union, intersect & Minus Clause, Creating view, Renaming the Column of a view

Basics of PL/SQL: PL/SQL - Introduction and advantages Understanding, PL/SQL Block structure, Fundamentals of PL/SQL Language - data types, variables, constants and expressions, Operators, Conditional statement, Controlling loop iterations

Course Outcomes

A student will be able to:

- Gain knowledge of database systems and database management systems software.
- Ability to model data in applications using conceptual modeling tools such as ER Diagrams and design database schemas based on the model.
- Formulate, using SQL, solutions to a broad range of query and data update problems.
- Demonstrate an understanding of normalization theory and apply such knowledge to the normalization of a database.
- Be acquainted with the basics of transaction processing and concurrency control.

Suggested Readings:

1. Database System Concepts, 7th Edition, Silberschatz, Korth and Sudarshan, McGraw-Hill. Indian Edition released 2021 .
2. Fundamentals of Database Systems, 7th Edition, Elmasri and Navathe, Pearson Pubs, 2017 .
3. Principles of Database Management, Lemahieu, Broucke and Baesens, Cambridge University Press, 2018.

Lab-2 :Database Management Lab (Based on 24BCAI-N-201)	0L:0T:2P	1 Credits
---	----------	-----------

Maximum Marks:30
External Examination: 20
Internal Assessment: 10

Course Outcomes: By the end of the course the student will be able to:

- To practically work with SQL Server and Oracle Database Server.
- To work with PL/SQL
- Implement triggers and stored procedures to handle complex business data.

List of Practical:

1. To install the SQL Server.
2. The STUDENT detail database has a table with the following attributes. The primary keys are underlined. STUDENT(regno: int, name: string, dob: date, marks: int, city: string)
 - i) Create the above table.
 - ii) Remove the city attribute from the table.
 - iii) Change the date type of regno from integer to string.
 - iv) Add a new attribute phoneno to the existing table.
 - v) Enter five tuples into the table.
 - vi) Display all the tuples in the student table.
3. A LIBRARY database has a table with the following attributes. LIBRARY(bookid:int, title:string, author:string, publication:string, yearpub:int, price:real)
 - i) Create the above table.
 - ii) Enter the five tuples into the table
 - iii) Display all the tuples in the library table.
 - iv) Display the different publishers from the list.
 - v) Arrange the tuples in the alphabetical order of the book titles.
 - vi) List the details of all the books whose price ranges between Rs. 100 and Rs. 300
4. The SALARY database of an organization has a table with the following attributes. EMPSALARY(empcod:int, empname:string, dob:date, department:string, salary:real)
 - i) Create the above table.
 - ii) Enter the five tuples into the table
 - iii) Display all the number of employees working in each department.
 - iv) Find the sum of the salaries of all employees.
 - v) Find the sum and average of the salaries of employees of a particular department.
 - vi) Find the least and highest salaries that an employee draws.
5. Consider the insurance database given below. The primary keys are underlined and the data types are specified.
PERSON(driver-id-no: string, name: string, address:string)
CAR(regno: string, model: string, year: int)
ACCIDENT(report-no: int, date: date, location: String)
OWNS(driver-id-no: string, regno: string)
PARTICIPATED(driver-id-no: string, regno: string, report-no: int, damage-amount: int)
 - i) Create the above tables by properly specifying the primary keys and the foreign keys
 - ii) Enter at least five tuples for each relation.
 - iii) Demonstrate how you
 - a) Update the damage amount for the car with a specific regno in the accident with report no 12 to 25000.
 - b) Add a new accident to the database.
 - iv) Find total number of people who owned cars that were involved in accidents in 2002
 - v) Find the number of accidents in which cars belonging to a specific model were involved

6. Consider the following database of students enrollment in courses and books adopted for each course.
 STUDENT(regno: string, name: string, major: strong, bdate: date)
 COURSE(course-no: int, cname: string, dept: string)
 ENROLL(reg-no: string, course-no: int, sem: int, marks: int)
 BOOK-ADOPTION(course-no: int, sem: int, book-isbn: int)
 TEXT(book-isbn: int, book-title: string, publisher: string, author: string)
 i) Create the above tables by properly specifying the primary keys and the foreign keys
 ii) Enter atleast five tuples for each relation.
 iii) Demonstrate how you add a new text book to the database and make this book be adopted by some department.
 iv) Produce a list of textbooks (include Course-no, book-isbn, book-title) in the alphabetical order for courses offered by the 'Compute Science' department that use more than two books.
 v) List any department that has all its adopted books published by a specific publisher.
7. The following tables are maintained by a book dealer
 AUTHOR(author-id: int, name: string, city: string, country: string)
 PUBLISHER(publisher-id: int, name: string, city: string, country: string)
 CATALOG(book-id: int, title : string, author-id: int, publisher-id: int, category: int, year: int, price: int)
 CATEGORY(category-id: int, description: string)
 ORDER-DETAILS(order-no: int, book-id: int, quantity: int)
 i) Create above tables by properly specifying the primary keys and the foreign keys.
 ii) Enter atleast five tuples for each relation.
 iii) Give the details of the authors who have 2 or more books in the catalog and the price of the books is greater than the average price of the books in the catalog and the year of publication is after 2010.
 iv) Find the author of the book which has maximum sales.
 v) Demonstrate how to increase price of books published by specific publisher by 10%
8. Consider the following database for BANK.
 BRANCH(branch-name: string, branch-city: string, assets: real)
 ACCOUNT(accno: int, branch-name: string, balance: real)
 DEPOSITOR(customer-name: string, accno: int)
 CUSTOMER(customer-name: string, customer-street: string, customer-city: string)
 LOAN(loan-no: int, branch-name: string, amount: real)
 BORROWER(customer-name: string, loan-no: int)
 i) Create the above tables by properly specifying the primary keys and foreign keys.
 ii) Enter atleast five tuples for each relation.
 iii) Find all the customers who have atleast two accounts at the main branch.
 iv) Find all customers who have an account at all the branches located in a specific city.
 v) Demonstrate how to delete all account tuples at every branch located in a specific city.
9. Consider the following database for ORDER PROCESSING.
 CUSTOMER(cust-no: int, cname: string, city: string)
 ORDER(orderno: int, odate: date, ord-amt: real)
 ORDER_ITEM(orderno: int, itemno: int, qty: int)
 ITEM(itemno: int, unitprice: real)
 SHIPMENT(orderno: int, warehouseno: int, ship-date: date)
 WAREHOUSE(warehouseno: int, city: string)
 i) Create the above tables by properly specifying the primary keys and the foreign keys
 ii) Enter atleast five tuples for each relation.
 iii) List the order number and ship date for all orders shipped from particular warehouse
 iv) Produce a listing: customer name, no of orders, average order amount
 v) List the orders that were not shipped within 30 days of ordering
10. Create two tables: motherboard (motherboard_Name, Price) and processor (processor_Name, Price) and insert 3 records in each table. Now write a query to display all maximum combinations of motherboard and processor with their price.
11. Write a PL/SQL program to find the largest of three numbers
12. Write a PL/SQL program to find the factorial of a number..

24BCAI-N-203	Software Engineering	3L:0T:0P	3 Credits
--------------	----------------------	----------	-----------

Maximum Marks:100
External Examination: 70
Internal Assessment: 30
Max. Time: 3 Hrs

Course Objectives

- The objective of this course is to educate the students about the different models of software development and metrics used in software engineering.
- Students will get knowledge of Software Testing.

Note: There shall be nine questions in all. Question no. 1 shall be compulsory, consisting of eight short answer type questions covering the entire syllabus. Two questions will be asked from each unit. Students will have to attempt one question from each unit. Each question shall carry equal marks.

UNIT – I

Introduction: Program vs. Software, Software Engineering, Programming paradigms, Software Crisis – problem and causes, Phases in Software development: Requirement Analysis, Software Design, Coding, Testing, Maintenance, Software Development Process Models: Waterfall, Prototype, Evolutionary and Spiral models, Role of Metrics.

UNIT – II

Feasibility Study, Software Requirement Analysis and Specifications: SRS, Need for SRS, Characteristics of an SRS, Components of an SRS, Problem Analysis, Information gathering tools, Organizing and structuring information, Requirement specification, validation and Verification.

UNIT – III

Structured Analysis and Tools: Data Flow Diagram, Data Dictionary, Decision table, Decision trees, Structured English, Entity-Relationship diagrams, Cohesion and Coupling. Gantt chart, PERT Chart, Software Maintenance: Type of maintenance, Management of Maintenance, Maintenance Process, maintenance characteristics.

UNIT – IV

Software Project Planning: Cost estimation: COCOMO model, Project scheduling, Staffing and personnel planning, team structure, Software configuration management, Quality assurance plans, Project monitoring plans, Risk Management. Software testing strategies: unit testing, integration testing, Validation testing, System testing, Alpha and Beta testing.

Course Outcomes

A student will be able to:

- Understand concept of Software Engineering and various SDLC; DSE-III(III).
- Learn to calculate the cost of a software project for an Enterprise; DSE-III(III).
- Gather basic information of an enterprise to design a software; DSE-III(III).
- Understand the fundamentals of Software Testing and Software Maintenance.

Suggested Readings:

1. Pressman S. Roger, Software Engineering, Tata McGraw Hill.
2. Jalote Pankaj, An Integrated Approach to Software Engineering, Narosa Publ. House.
3. K. K. Aggarwal, Yogesh Singh, Software Engineering, New Age International.
4. Sommerville, Software Engineering, Pearson Education.
5. Fairley Richard, Software Engineering Concepts, Tata Mc-Graw Hill Ed.
6. Rajib Mall, Fundamentals of Software Engineering, PHI Learning.

24BCAI-N-204	Mathematical Foundation of Computer Science	3L:0T:0P	3 Credits
--------------	--	----------	-----------

Maximum Marks:70
External Examination: 50
Internal Assessment: 20
Max. Time: 3 Hrs

Course Objectives

- This course helps the students to demonstrate various principles involved in solving mathematical problems and thereby reducing the time taken for performing job functions and solving problems.

Note: *There shall be nine questions in all. Question no. 1 shall be compulsory, consisting of eight short answer type questions covering the entire syllabus. Two questions will be asked from each unit. Students will have to attempt one question from each unit. Each question shall carry equal marks.*

Unit I

Heights & Distances : Interest, Heights and Distances, Profit and Loss, Partnership, Mixture and Allegation, Time and distance Series, Time & Work, The Data Interpretation part covers Tabulation., Calendar, Clocks, Heights & Distances

Unit II

Number Problems : Numbers, H.C.F. & L.C.M. of Numbers, Decimal Fractions, Simplification, Square Roots & Cube Roots, Whole numbers problems, Decimals problems, Problems on Trains

Unit III

Database Modeling : Fractions problems, Numbers and Ages, Percentage problems, Boats and Streams, Ratio & Proportion, Square roots, Averages.

Unit IV

Statistics and Correlation: Meaning, Scope and Limitations of Statistics, Types and Sources of Data, Methods and Problems of Collection of Primary and Secondary Data

Measures of Central Tendency (Arithmetic Mean, Median, Partition Values, Mode), Measure of Dispersion (Absolute and Relative Measures Range, Quartile Deviation, Mean Deviation, Standard Deviation and Coefficient of Variation)

Correlation: Definition, Scatter diagram, Karl Pearson's coefficient of correlation, Numerical Problems for determination of Correlation coefficient. Regression: Definition, dependent and independent variables, least Square method only, Numerical problems.

Permutations & Combinations, Probability,

Course Outcomes

A student will be able to:

- Common methodologies of solving questions related to Quantitative Aptitude and basics of Statistics.
- Learn the various question types of aptitude and their basic approaches.
- Create a bridge for the advanced levels of aptitude and prepare a base for problem solving and employability of the student.

Suggested Readings:

- 1) Quantitative Aptitude for Competitive Examinations by R. S. Aggarwal.
- 2) Business Statistics by TR Jain and SC Aggarwal

Scheme of Examination for BCA(Industry Integrated)**Semester: 3rd (w.e.f. Session 2024-25)**

S r · N o .	Course/ Paper Code	Title of the Course	Mode of Learn ing	Course Type	Credit			Contact Hours per Week			Examination Scheme				Tot al Mar ks
					The ory	Practical/ Seminar	Tot al	Theo ry	Practi cal /Semi nar	Tot al	End Semest er Exa m)	Intern al Assessm ent	Prac tical (Exte rnal)	Practi cal (Inter nal)	
1	24BCAI- N- 301	Java Programming	CL/BL	CC	3	1	4	3	2	5	50	20	20	10	100
2	24BCAI- N-302	Operating system	CL/BL	CC	3	-	3	3	..	3	50	20	.	.	70
3	24BCAI-N-303	Discrete Mathematics	CL/BL	CC	4	..	4	4		4	70	30	.	.	100
4	24BCAI-N-304	Data Analysis using Excel	CL/BL	MIC	2	1	3	2	2	4	35	15	15.	5	70
5	24UN-ENG-AEC301	English Grammar and Usage	CL/BL	AEC	2	..	2	2	..	2	35	15	.	.	50
6	24BCAI- N - SEC301	Internship	CL	SEC	..	4	4	-.					70	30	100
10	24UN-BBA-MDC301	Fundamentals of Indian Capital Markets	CL	MDC	3	..	3	3	..	3	50	20			70
11	24UN-GEO-VAC303	Culture of India	CL	VAC	2	..	2	2	..	2	35	15			50
	Total						25			23					610

CC - Core Course; SEC-Skill Enhancement Course; AEC: Ability Enhancement Course MDC: Multidisciplinary Course VAC: Value Added course**MIC: Minor course**

24BCAI- N- 301	Java Programming	3L:0T:0P	3 Credits
----------------	------------------	----------	-----------

Maximum Marks:70
External Examination: 50
Internal Assessment: 20
Max. Time: 3 Hrs

Course Objectives

- To introduce the object-oriented programming concepts.
- To understand object-oriented programming concepts, and apply them in solving problems.
- To introduce the principles of inheritance and polymorphism; and demonstrate how they relate to the design of abstract classes
- To introduce the implementation of packages and interfaces
- To introduce the concepts of exception handling and multithreading.
- To introduce the design of Graphical User Interface using applets.

Note: There shall be nine questions in all. Question no. 1 shall be compulsory, consisting of eight short answer type questions covering the entire syllabus. Two questions will be asked from each unit. Students will have to attempt one question from each unit. Each question shall carry equal marks.

Unit I

Object Oriented Programming and Java Fundamentals: Structure of Java programs, Classes and Objects, Data types, Type Casting, Looping Constructs.

Unit II

Interfaces: Interface basics; Defining, implementing and extending interfaces; Implementing multiple inheritance using interfaces. Packages: Basics of packages, Creating and accessing packages, System packages, Creating user defined packages

Unit III

Exception handling using the main keywords of exception handling: try, catch, throw, throws and finally; Nested try, multiple catch statements, creating user defined exceptions. File Handling Byte Stream, Character Stream, File I/O Basics, File Operations

Unit IV

AWT and Event Handling: The AWT class hierarchy, Events, Event sources, Event classes, Event Listeners, Relationship between Event sources and Listeners, Delegation event model, Creating GUI applications using AWT.

Course Outcomes

- Able to solve real world problems using OOP techniques.
- Able to understand the use of abstract classes.
- Able to solve problems using java collection framework and I/o classes.
- Able to develop multithreaded applications with synchronization.
- Able to develop applets for web applications.

- Able to design GUI based applications consisting of multiple classes.

Suggested Readings:

1. Balagurusamy, E., “Programming with JAVA, Tata McGraw Hill.
2. The Complete Reference JAVA, TMH Publication.
3. Beginning JAVA, Ivor Horton, WROX Public.
4. JAVA 2 UNLEASHED, Tech Media Publications.
5. Patrick Naughton and HerbertzSchildt, “Java-2 The Complete Reference”, TMH.

24BCAI- N-302	Operating system	3L:0T:0P	3 Credits
---------------	-------------------------	----------	-----------

Maximum Marks:70
External Examination: 50
Internal Assessment: 20
Max. Time: 3 Hrs

Course Objectives

- To understand the basic concepts, functions and types of operating systems.
- To understand Processes and Process scheduling
- To understand the concept of Deadlocks
- To analyze various memory management schemes.
- To understand I/O management and File systems.

Note: There shall be nine questions in all. Question no. 1 shall be compulsory, consisting of eight short answer type questions covering the entire syllabus. Two questions will be asked from each unit. Students will have to attempt one question from each unit. Each question shall carry equal marks.

Unit I

Introduction: Introduction to Operating System Concepts (Including Multitasking, Multiprogramming, Multi User, Multithreading etc.), Types of Operating Systems: Batch Operating System, Time-Sharing System, Distributed Operating System, Network Operating System, Real Time Operating System, Various Operating System services, architecture, System Programs and Calls.

Unit II

Process Management: Process Concept, Process Scheduling, Operations on Processes; CPU Scheduling, Scheduling Criteria, Scheduling Algorithms- First come First Serve (FCFS), Shortest- Job- First (SJF), Priority Scheduling, Round Robin (RR).
Deadlock: Methods for handling deadlock- Deadlock prevention, Avoidance

Unit III

Memory Management: Logical & Physical Address Space, Swapping, Contiguous Memory allocation, non-contiguous memory allocation, paging and segmentation techniques; Virtual Memory Management- Demand Paging & Page Replacement Algorithm; Demand Segmentation.

Unit IV

File System: Different types of files and their access methods, directory structures, various allocation methods, disk scheduling and management and its associated algorithms.

Course Outcomes

- Analyze various scheduling algorithms.
- Understand deadlock, prevention and avoidance algorithms.
- Compare and contrast various memory management schemes.
- Understand the functionality of file systems

Suggested Readings:

1. Operating System Concepts by Silberschatz et al 5th Edition 1998, Addison-Wesley.
2. Modern Operating System by A. Tanenbaum, 1992, Prentice Hall
3. Operating System By Peterson, 1985, AW.
4. Operating System By Milankovik, 1990, THM
5. Operating System Incorporating with Unix & Windows by Colin Ritchie, 1974, THM.

24BCAI-N-303	Discrete Mathematics	4L:0T:0P	4 Credits
--------------	----------------------	----------	-----------

Maximum Marks:100
External Examination: 70
Internal Assessment: 30
Max. Time: 3 Hrs

Course Objectives

- To provide students with a solid foundation in set theory, logic, inference, counting, relations, graph theory, algebraic structures, and computability theory, equipping them with the skills to solve complex mathematical problems and apply these concepts in computer science.

Note: *There shall be nine questions in all. Question no. 1 shall be compulsory, consisting of eight short answer type questions covering the entire syllabus. Two questions will be asked from each unit. Students will have to attempt one question from each unit. Each question shall carry equal marks.*

Unit I

Sets & Logic: Sets - Visualizing sets, understanding membership, operations on sets (union, intersection, difference, complement). Algebra of sets and its laws, Mathematical Induction, Relations, Types of Relation- reflexive, symmetric, antisymmetric, Equivalence Relations
Logic - Introduction to propositional logic, truth tables, Logical Operations, Conditional Statements

Unit II

Vectors & Matrix: Vectors - Operations, Column Vectors, Matrices - Matrix Addition, Matrix Multiplication, Transpose, Square Matrices and Algebra on Square Matrices, Inverse of Square Matrix
Functions & Algorithms: Functions as Relations, One-to-One, Onto and Invertible Functions, Mathematical, Exponential and Logarithmic Functions, Recursively Defined Functions (Examples like- Factorial, Fibonacci), Algorithm and Functions - Polynomial Evaluation (Direct Method, Synthetic Division), Complexity of Algorithm - Best Case, Worst Case, Average Case

Unit III

Counting Techniques: Counting Principles - Sum Rule and Product Rule. Mathematical Functions - Factorial Function, Binomial Coefficients. Binomial Theorem - Binomial Coefficients and Pascal's Triangle. Permutations & Combinations - Understanding arrangements (permutations) and selections (combinations), using formulas for common scenarios. Tree Diagram

Unit IV

Graph - Graphs and Multigraphs, Degree of Vertex, Subgraphs, Isomorphic Graphs, Homeomorphic Graphs, Path, Connected Graph, Labeled Graph, Weighted Graph, Directed Graph, Undirected Graph, Complete Graphs, Regular Graphs, Bipartite Graphs, Planar Graph, Maps & Regions - Euler's Formula, Representing Graphs in Memory - Adjacency Matrix, Linked Representation of a Graph
Tree - Tree Graphs, Spanning Trees, Minimum Spanning Trees, Binary Trees, Representing Binary Trees in Memory - Linked & Sequential, Traversing Binary Tree, Binary Search Tree

Course Outcomes:

A student will be able to:

- Understand sets, sequences, matrices, and logical operations..
- Use statements, connectives, and normal forms in logic.
- Apply inference theories in statement and predicate calculus.
- Solve problems using permutations, combinations, and recurrence relations; manipulate relations and digraphs
- Understand graph theory concepts, Boolean algebras, and basic computability; explore semi-groups and groups.

Suggested Readings:

1. E. Goodaire , “Discrete Mathematics with Graph theory”, PHI,. ISBN--10: 0131679953 2) J. K. Sharma, “Discrete Mathematics”, McMillan, ISBN-9780230322301
2. J.P. Tremblay and R. Manohar, “Discrete Mathematical Structure with Applications to Computer Science”, TMH, ISBN- 0-07-463113-6
3. Bernard Kolman, Robert C. Busby and Sharon Ross, “Discrete Mathematical Structure”, PHI, ISBN- 978-81-203-3689-6
4. C. L. Liu, “Elements of Discrete Mathematics”, TMH

24BCAI-N-304	Data Analysis using Excel	3L:0T:0P	2 Credits
--------------	----------------------------------	----------	-----------

Maximum Marks:50
External Examination: 35
Internal Assessment: 15
Max. Time: 3 Hrs

Course Objectives

- Understand the fundamental elements of a dashboard, including charts, tables, and visuals, to effectively present data-driven insights.
- Develop skills in utilizing advanced data analysis tools to derive actionable insights and drive data-informed decision-making processes.
- Explore methods for linking Excel with databases, facilitating seamless data exchange between Excel and external data sources such as Access and SQL Server.

Note: There shall be nine questions in all. Question no. 1 shall be compulsory, consisting of eight short answer type questions covering the entire syllabus. Two questions will be asked from each unit. Students will have to attempt one question from each unit. Each question shall carry equal marks.

Unit I

Introduction to Excel for Data Analysis-Overview of Excel interface, Basics of spreadsheets, Data types and formats, Basic formulas and functions. Data Entry and Cleaning- Data entry techniques, importing data from various sources (CSV, Text, Web), Data cleaning techniques, removing duplicates, handling missing values.

Unit II

Basic Data Manipulation- Sorting and filtering data, using tables in Excel, Conditional formatting, Data validation, Advanced Formulas and Functions- Logical functions: IF, AND, OR; Lookup functions: VLOOKUP, HLOOKUP, INDEX, MATCH; Text functions: CONCATENATE, LEFT, RIGHT, MID, TRIM, Date and time functions

Unit III

Data Analysis Techniques-Descriptive statistics: mean, median, mode, standard deviation, Using the Analysis Toolpack, PivotTables and PivotCharts, Grouping and summarizing data
Data Visualization- Creating and formatting charts: bar, line, pie, scatter, Advanced charting techniques, Using Sparklines, Dynamic charts with interactive controls.

Unit IV

Introduction to Dashboards- Elements of a dashboard, designing interactive dashboards, Using form controls and slicers, Linking charts and tables for dynamic updates.
Advanced Data Analysis Tools- What-If analysis: Scenario Manager, Goal Seek, Data Tables, Solver add-in for optimization problems, Using Power Query for data extraction and transformation, Integrating Excel with Other Tools-Linking Excel with databases, Exporting and importing data with other software (e.g., Access, SQL Server), Using Excel with cloud services (e.g., Google Sheets, Power BI).

Course Outcomes

A student will be able to

- Design and create interactive dashboards that effectively communicate complex data insights to stakeholders.
- Apply techniques like What-If analysis, optimization, and data transformation tasks using advanced Excel tools.
- Demonstrate the ability to import, export, and manipulate data across different platforms, improving data interoperability and workflow efficiency.
- Apply their understanding of dashboard design, advanced data analysis, and data integration to solve complex business challenges effectively.

Suggested Readings:

1. "Information Dashboard Design: Displaying Data for At-a-Glance Monitoring" by Stephen Few
2. "Excel 2019 Bible" by Michael Alexander and Richard Kusleika
3. "Power Pivot and Power BI: The Excel User's Guide to DAX, Power Query, Power BI & Power Pivot in Excel 2010-2016" by Rob Collie and Avichal Singh
4. "SQL Server 2019 Integration Services Cookbook" by Christian Cote, MatijaLah, DejanSarka, and David Dye.

Java Programming Lab (Based on 24BCAI- N- 301)	0L:0T:2P	1 Credits
--	----------	-----------

Maximum Marks:30
External Examination: 20
Internal Assessment: 10

Lab 1: Basics

- Write a Java program that prints "Hello, World!" to the console.
- Create a Java program that calculates the factorial of a given number using a loop.
- Develop a Java program to print the Fibonacci series up to a given limit.

Lab 2: Interface

- WAP to find the area of rectangle and circle using Interface.

Lab 3: OOPS

Create a class Film with string objects which stores name, language and lead_actor and category (action/drama/fiction/comedy). Also include an integer data member that stores the duration of the film. Include parameterized constructor, default constructor and accessory functions to film class. Film objects can be initialized either using a constructor or accessor functions. Create a class FilmMain that includes a main function. In the main function create a vector object that stores the information about the film as objects. Use the suitable methods of vector class to iterate the vector object to display the following

- The English film(s) that has Arnold as its lead actor and that runs for shortest duration.
- The Tamil film(s) with Rajini as lead actor.
- All the comedy movies.

Lab 4: OOPS inheritance

Define an abstract class 'Themepark' and inherit 2 classes 'Queensland' and 'Wonderland' from the abstract class. In both the theme parks, the entrance fee for adults is Rs. 500 and for children it is Rs. 300. If a family buys 'n' adult tickets and 'm' children tickets, define a method in the abstract class to calculate the total cost. Also, declare an abstract method playGame() which must be redefined in the subclasses. In Queensland, there are a total of 30 games. Hence create a Boolean array named 'Games' of size 30 which initially stores false values for all the elements. If the player enters any game code that has already been played, a warning message should be displayed and the user should be asked for another choice. In Wonderla, there are a total of 40 different games. Thus create an integer array with 40 elements. Here, the games can be replayed, until the user wants to quit. Finally display the total count of games that were repeated and count of the games which were not played at all.

Lab 5: Inheritance

Define a superclass Shape with an abstract method calculateArea(). Implement subclasses Circle and Rectangle that inherit from Shape and override the calculateArea() method to calculate the area specific to each shape. Create objects of both subclasses and demonstrate polymorphism.

Lab 6: Exception Handling

Read the Register Number and Mobile Number of a student. If the Register Number does not contain exactly 9 characters or if the Mobile Number does not contain exactly 10 characters, throw an IllegalArgumentException. If the Mobile Number contains any

character other than a digit, raise a `NumberFormatException`. If the Register Number contains any character other than digits and alphabets, throw a `NoSuchElementException`. If they are valid, print the message 'valid' else 'invalid'.

Lab 7: Package

Write a program to demonstrate the knowledge of students in working with user-defined packages and sub-packages. Eg., Within the package named 'primespackage', define a class `Primes` which includes a method `checkForPrime()` for checking if the given number is prime or not. Define another class named `TwinPrimes` outside of this package which will display all the pairs of prime numbers whose difference is 2. (Eg, within the range 1 to 10, all possible twin prime numbers are (3,5), (5,7)). The `TwinPrimes` class should make use of the `checkForPrime()` method in the `Primes` class.

Lab 8: File Handling

Write a program to demonstrate the knowledge of students in File handling. Eg., Define a class 'Donor' to store the below mentioned details of a blood donor. Name, age, Address, Contactnumber, bloodgroup, date of last donation Create 'n' objects of this class for all the regular donors at Vellore. Write these objects to a file. Read these objects from the file and display only those donors' details whose blood group is 'A+ve' and had not donated for the recent six months.

Lab 9: AWT

- Write AWT program in Java to find the sum, Multiplication and average of three numbers entered in three Text fields by clicking the corresponding Labeled Button. The result should be appearing in fourth text field.
- Write Java Program to generate Even numbers and Odd Numbers in TextField "T1 and T2 respectively" while pressing Button "Even" and "Odd".

Data Analysis using Excel Lab (Based on 24BCAI-N-304)	0L:0T:2P	1 Credits
--	----------	-----------

Maximum Marks:20
External Examination: 15
Internal Assessment: 5

Practical1: Data Entry and Data Cleaning

- **Task:** Create a sample dataset with 10 rows and 5 columns.
- **Steps:**
 1. Enter data into the spreadsheet.
 2. Use Excel's data cleaning tools to remove duplicates, handle missing values, and format data.

Practical 2: Sorting and Filtering Data

- **Task:** Sort and filter a dataset to analyze specific data subsets.
- **Steps:**
 1. Sort data by different columns.
 2. Apply filters to display specific data subsets based on criteria.

Practical 3: Using Formulas and Functions

- **Task:** Apply basic and conditional functions to a dataset.
- **Steps:**
 1. Use basic functions like SUM, AVERAGE, MIN, and MAX.
 2. Apply conditional functions such as IF, COUNTIF, and SUMIF.

Practical 4: Data Validation

- **Task:** Implement data validation rules to ensure accurate data entry.
- **Steps:**
 1. Create drop-down lists.
 2. Set validation rules for data entry to ensure data integrity.

Practical 5: Conditional Formatting

- **Task:** Use conditional formatting to highlight data based on specific criteria.
- **Steps:**
 1. Highlight cells based on specific criteria.
 2. Use color scales, data bars, and icon sets for visual representation.

Practical 6: Creating and Formatting Tables

- **Task:** Convert data ranges into tables and apply table styles.
- **Steps:**
 1. Convert a range of data into an Excel table.
 2. Apply different table styles and properties for better readability.

Practical 7: Charts and Graphs

- **Task:** Visualize data by creating various types of charts.
- **Steps:**
 1. Create bar, line, and pie charts.
 2. Customize chart elements such as titles, labels, and legends.

Practical 8: Pivot Tables

- **Task:** Summarize and analyze large datasets using pivot tables.
- **Steps:**
 1. Create pivot tables from large datasets.
 2. Analyze data by rearranging fields in the pivot table.

Practical 9: Pivot Charts

- **Task:** Create and customize pivot charts to visualize pivot table data.
- **Steps:**
 1. Create pivot charts from pivot tables.
 2. Customize pivot chart elements for clarity.

Practical 10: VLOOKUP and HLOOKUP Functions

- **Task:** Retrieve data using VLOOKUP and HLOOKUP functions.
- **Steps:**
 1. Use VLOOKUP to find data in vertical columns.
 2. Use HLOOKUP to find data in horizontal rows.

Practical 11: Text Functions

- **Task:** Clean and manipulate text data using text functions.
- **Steps:**
 1. Use functions like LEFT, RIGHT, MID, and CONCATENATE.
 2. Clean and format text data as required.

Practical 12: Date and Time Functions

- **Task:** Perform calculations with date and time data.
- **Steps:**
 1. Use functions like TODAY, NOW, DATE, and TIME.
 2. Calculate differences between dates and times.

Scheme of Examination for BCA (Industry Integrated)**Semester: 4th (w.e.f. Session 2024-25)**

S r · N o ·	Course/ Paper Code	Title of the Course	Mode of Learnin g	Course Type	Credit			Contact Hours per Week			Examination Scheme				Total Mark s
					The ory	Practical / Seminar	Tot al	Theo ry	Practi cal /Semi nar	Tot al	End Semest er Exa m	Int ernal Asse ssme nt	Prac tical (Exte rnal)	Practic al (Intern al)	
1	24BCAI- N-401	Data Structure	CL/B L	CC	3	1	4	3	2	5	50	20	20	10	100
2	24BCAI- N-402	Linux and Shell Programming	CL/B L	CC	3	1	4	3	2	5	50	20	20	10	100
3	24BCAI-N-403	Back End Engineering	CL/B L	CC	3	1	4	3	2	5	50	20	20	10	100
4	24BCAI- N-404	Cloud Computing	CL/B L	MIC	3	..	3	3		3	50	20	..		70
5	24UN-ENG-AEC401	Vocabulary Building	CL/B L	AEC	2	..	2	2	..	2	35	15	· ·		50
	Total						17			20					420

CC - Core Course;**MIC: Minor course****AEC: Ability Enhancement Course**

24BCAI- N- 401	Data Structure	3L:0T:0P	3 Credits
----------------	----------------	----------	-----------

Maximum Marks:70
 External Examination: 50
 Internal Assessment: 20
 Max. Time: 3 Hrs

Course Objectives

- The Purpose of the course is to introduce the principles and paradigms of programming languages for design and implement the software intensive systems.

Note: *There shall be nine questions in all. Question no. 1 shall be compulsory, consisting of eight short answer type questions covering the entire syllabus. Two questions will be asked from each unit. Students will have to attempt one question from each unit. Each question shall carry equal marks.*

Unit I

Data Structure Definition, Data Type vs. Data Structure, Classification of Data Structures, Data Structure Operations, Applications of Data Structures.

Algorithm Specifications: Performance Analysis and Measurement (Time and Space Analysis of Algorithms- Average, Best and Worst Case Analysis).

Arrays: Introduction, Linear Arrays, Representation of Linear Array in Memory, Two Dimensional and Multidimensional Arrays, Sparse Matrix and its Representation, Operations on Array: Algorithm for Traversal, Selection, Insertion, Deletion and its implementation.

Unit II

String Handling: Storage of Strings, Operations on Strings viz.,Length, Concatenation, Substring, Insertion, Deletion, Replacement,Pattern Matching

Linked List: Introduction, Array vs. linked list, Representation of linked lists in Memory, Traversing a Linked List, Insertion, Deletion,Searching into a Linked list, Type of Linked List.

Unit III

Stack: Array Representation of Stack, Linked List Representation of Stack, Algorithms for Push and Pop, Application of Stack: Polish Notation, Postfix Evaluation Algorithms, Infix to Postfix Conversion, Infix to Prefix Conversion, Recursion.

Introduction to Queues: Simple Queue, Double Ended Queue, Circular Queue, Priority Queue, Representation of Queues as Linked List and Array, Applications of Queue. Algorithm on Insertion and Deletion in Simple Queue and Circular Queue. Priority Queues.

Unit IV

Tree: Definitions and Concepts, Representation of Binary Tree, Binary Tree Traversal (Inorder, postorder, preorder), Binary Search Trees – Definition, Operations viz., searching, insertions and deletion; Searching and Sorting Techniques, Sorting Techniques: Bubble sort, Merge sort, Selection sort, Quick sort, Insertion Sort. Searching Techniques: Sequential Searching, Binary Searching.

Course Outcomes

A student will be able to:

- Understand and analyze the basic concepts of complexity of an algorithm.
- Describe how arrays, linked lists, stacks, queues, trees, and graphs are represented in memory, used by the algorithms and their common applications.
- Design & Analyze the searching and sorting algorithms.
- Implement the Graphs and also perform various operations on this data structure.
- Understand & Implement the various concepts of stacks and queues

Suggested Readings:

1. Lipschutz, “Data Structures” Schaum’s Outline Series, Tata McGraw-hill Education (India) Pvt. Ltd.
2. Thareja, “Data Structure Using C” Oxford Higher Education.
3. Aaron M. Tenenbaum, YedidyahLangsam and Moshe J. Augenstein, “Data Structures Using C and C++”, PHI Learning Private Limited, Delhi India
4. Horowitz and Sahani, “Fundamentals of Data Structures”, Galgotia Publications Pvt Ltd Delhi India.

24BCAI- N- 402	Linux and Shell Programming	3L:0T:0P	3 Credits
----------------	------------------------------------	----------	-----------

Maximum Marks:70
 External Examination: 50
 Internal Assessment: 20
 Max. Time: 3 Hrs

Course Objectives

- The Purpose of the course is to introduce the Unix/Linux operating system and shell programming basics and knowledge of Unix/Linux tools for file handling.

Note: There shall be nine questions in all. Question no. 1 shall be compulsory, consisting of eight short answer type questions covering the entire syllabus. Two questions will be asked from each unit. Students will have to attempt one question from each unit. Each question shall carry equal marks.

Unit I

Introduction to Linux: Linux distributions, Overview of Linux operating system, Linux architecture, Features of Linux, Accessing Linux system, Starting and shutting down system, Logging in and Logging out, Comparison of Linux with other operating systems.

Unit II

Commands in Linux: General-Purpose commands, File oriented commands, directory oriented commands, Communication-oriented commands, process oriented commands, etc. Regular expressions & Filters in Linux: Simple filters viz. more, wc, diff, sort, uniq, grep; Introducing regular expressions.

Unit III

Linux file system: Linux files, inodes and structure and file system, file system components, standard file system, file system types. Processes in Linux: Starting and Stopping Processes, Initialization Processes, Mechanism of process creation, Job control in linux using at, batch, cron& time.

Unit IV

Shell Programming: vi editor, shell variables, I/O in shell, control structures, loops, subprograms, creating & executing shell scripts in linux.

Course Outcomes

A student will be able to:

- Understand the Unix/Linux operating system, installation procedures, basic utilities command and basics of shell programming.
- Organize & handle the files or records through various Unix/Linux tools by using regular expression.
- Analyze C/C++ environment, Makefile, libraries, linking, debugging methodologies and apply the advantages of Python programming.
- Assess the processes and manage the networks securely in Linux. CO5: Analyze process manipulation and learning password management for user privileges in Linux.

Suggested Readings:

1. John Goerzen: Linux Programming Bible, IDG Books, New Delhi, 2000 Neil
2. Matthew, Richard Stones: Beginning Linux Programming, Wrox-Shroff, 2007

24BCAI-N-403	Back End Engineering	3L:0T:0P	3 Credits
--------------	-----------------------------	----------	-----------

Maximum Marks:70
 External Examination: 50
 Internal Assessment: 20
 Max. Time: 3 Hrs

Course Objectives

- The Purpose of the course is to introduce the concepts of Backend-Engineering.

Note: *There shall be nine questions in all. Question no. 1 shall be compulsory, consisting of eight short answer type questions covering the entire syllabus. Two questions will be asked from each unit. Students will have to attempt one question from each unit. Each question shall carry equal marks.*

Unit I

Client Server Architecture: Introduction to client Server Architecture, Understanding Server and how request is handled, Understanding Nodejs, advantages and disadvantages, Folder structure and basic Node server and understanding modules

Creating basic node server, handling request, creating endpoints, Modules, import modules Handling static pages with file stream, Handling exception, callback and other methods of endpoints,

Unit II

Express and Routing: Introduction to Express- Request lifecycle in express, how request travels in express, blocking vs non-blocking code, Handling static file.

Routing and different type like Static, Dynamic.

Unit III

Middlewares and Client Server Management: Intro to Middlewares, Middleware lifecycle, serve static files, Body parser, Introduction to Multer Disk storage and File Filter handling in Multer Client state management Introduction, Client state management using session module, Different ways of managing session, Node style async error handling.

Authentication: Authentication, Authorization, Handling Exceptions during authorization And authentication

SSR - How to use EJS to template your Node.js application, Create the EJS Partials, Pass Data to Views and Partials

Unit IV

Database: Introduction to databases, SQL and No SQL Databases, MongoDB Shell and MongoDB Compass Installation, MongoDB Shell and commands, Applying CRUD operations on database and creating Pages using EJS

Course Outcomes

A student will be able to:

- Understand Backend technologies with Nodejs and MongoDB.
- Understand and Create Nodejs server, handling request, creating endpoints.

- Understand and applying Session Management and Routing.
- Use of NoSQL database MongoDB and applying CRUD operations on MongoDB.
- Apply advanced Excel tools and features, such as PivotTables and Power Query, to handle complex data analysis tasks

Suggested Readings:

1. Eloquent JavaScript by MarijnHaverbeke, No Starch Press, 3rd edition (2018).
2. Mastering Node.js by Sandro Pasquali, Packt Publishing Ltd, Second Edition
3. MongoDB Basics by Peter Membrey, David Hows and EelcoPlugge, Apress, 1st Edition.

24BCAI- N-404	Cloud Computing	3L:0T:0P	3 Credits
---------------	-----------------	----------	-----------

Maximum Marks:70
External Examination: 50
Internal Assessment: 20
Max. Time: 3 Hrs

Course Objectives

- The Purpose of the course is to understand the evolution of computer networks using the layered network architecture

Note: There shall be nine questions in all. Question no. 1 shall be compulsory, consisting of eight short answer type questions covering the entire syllabus. Two questions will be asked from each unit. Students will have to attempt one question from each unit. Each question shall carry equal marks.

Unit I

Cloud Computing – Definition of Cloud – Evolution of Cloud Computing – Underlying Principles of Parallel and Distributed, History of Cloud Computing - Cloud Architecture - Types of Clouds - Business models around Clouds – Major Players in Cloud Computing Issues in Clouds - Eucalyptus - Nimbus - Open Nebula, CloudSim.

Unit II

Types of Cloud services: Software as a Service Platform as a Service –Infrastructure as a Service - Database as a Service - Monitoring as a Service –Communication as services. Service providers- Google, Amazon, Microsoft Azure, IBM, Sales force, S3 Storage.

Case Study: Creating EC2 Instances of Windows and Linux in AWS, Creating and Accessing Relational Database Services in AWS.

Unit III

Email Communication over the Cloud - CRM Management – Project Management-Event Management - Task Management – Calendar - Schedules - Word Processing – Presentation – Spreadsheet - Databases – Desktop - Social Networks and Groupware.

Case Study: CRM Management: Setting up a Salesforce Account and customizing dashboards & reports in CRM Tools

Unit IV

Need for Virtualization – Pros and cons of Virtualization – Virtualization and Cloud - Types of Virtualizations –System VM, Process VM, Virtual Machine monitor – Virtual machine properties - Interpretation and binary translation, HLL VM - supervisors – Xen, KVM, VMware, Virtual Box, Hyper-V.

Case Study: Installing and configuring Hyper-V/ KVM for installation of a Guest Operating System.

Security in Clouds: Cloud security challenges – Software as a Service Security, Common Standards: The Open Cloud Consortium – The Distributed Management Task Force – Standards for Application Developers Standards for Messaging – Standards for Security, End user access to cloud computing, Mobile Internet devices and the cloud.

Course Outcomes

A student will be able to:

- Illustrate the evolution and issues in Cloud Computing.
- Demonstrate the different services of Cloud.
- Analyze different services of cloud in industry prospective.
- Show the concept of virtualization in Cloud computing.
- Illustrate the core issues of cloud computing such as resource management and security.

Suggested Readings:

1. Haley Beard, “Best Practices for Managing and Measuring Processes for On-demand Computing, Applications and Data Centers in the Cloud with SLAs”, Emereo Pty Limited, July 2008.
2. G. J. Popek, R.P. Goldberg, “Formal requirements for virtualizable third generation Architectures, Communications of the ACM”, No.7 Vol.17, July 1974
3. David E.Y. Sarna, “Implementing and Developing Cloud Application”, CRC press 2011. 2. Lee Badger, Tim Grance, Robert Patt-Corner, Jeff Voas, NIST, Draft cloud computing synopsis and recommendation, May 2011.

Data Structure Lab (Based on 24BCAI- N- 401	0L:0T:2P	1 Credits
--	----------	-----------

Maximum Marks:30
External Examination: 20
Internal Assessment: 10

1. Write a program to find the 2nd maximum element in an array.
2. Given a list of integers, write a program to find out the number that has the highest frequency. If there are more than one such numbers, output the smaller one.
3. Given two arrays of the same size, write a program to find out if given arrays are equal or not.
Note: Two arrays are said to be equal if both of them contain the same set of elements, although arrangements (or permutation) of elements may be different.
4. Given a string, write a program to test the characters for their uniqueness. If all the characters occur at most 1 time in the string, then print "YES", otherwise if some character occurs at least twice in the string print "NO".
5. Write a program to count the numbers of words in a string.
A word is defined as text separated by space(' ') or multiple spaces.
6. Write a program to find the smallest and the largest element of a given linked list and move the smallest node to the front and move the largest node to the end of the list.
7. Write a program to check whether it contains a loop or not. If there is a loop return the number of nodes in the loop, otherwise return 0.
8. Write a program to rotate a given doubly linked list counterclockwise by k nodes.
9. Write a program to insert in a sorted circular linked list.
10. Write a program to convert the infix to postfix expression.
11. Write a program to implement a queue using a linked list.
12. Given a binary tree, write a program to print all paths from root node to leaf nodes with their respective lengths and total number of paths in it.
13. Write a program to print a binary tree with level order traversal.
14. Write a program to find the kth smallest element in the binary search tree.
15. Given a sorted array of n elements, and an integer k. Write a program to find whether there exists any two elements in the array that sums up to the given value k. If such a pair is found, then print "Found", else print "Not Found".

Linux and Shell Programming Lab (Based on 24BCAI- N-402)	0L:0T:2P	1 Credits
--	----------	-----------

Experiment	Experiment
1	Installation of Red Hat Linux using VMware Workstation or Oracle Virtual Box.
2	Demonstrate general purpose commands in Linux- cal, who, date, echo, bc, tty.
3	Demonstrate Directory and File Management in Linux- mkdir, rmdir, cd, pwd, ls, cat, rm, mv, cp.
4	Demonstrate to handle File and Directory permissions through: ls -l, ls -ld, chmod.
5	Show use of vi-editor and its various options.
6	Demonstrate the working of all process handling commands: ps, fg, bg, nice, renice, top etc.
7	Copy /etc/passwd file from your Linux system and demonstrate filters in Linux: more, less, diff, sort, Grep and sed.
8	Using crontab schedule below tasks in your Linux Server: · Run a command at 15:00 on every day from Monday through Friday · Run a script every day, every hour, on the hour, from 8 AM through 4 PM · Run the a script at 9:15pm, on the 1st and 15th of every month · Set custom HOME, PATH, SHELL and MAILTO variables and run a command every minute
9	Use at command to schedule: · To schedule a job for the coming Monday at a time twenty minutes later than the current time · To schedule a job to run at 1:45 on August 12, 2024 · schedule a job to run at 3 PM four days from the current time · schedule a job to shut down the system at 4:30 PM today

10	WAP that accepts user name and reports if a user is logged in or not. If logged in, display a greeting for him.
11	Create a shell script that checks if a file exists in the current directory or not.
12	WAP that takes a filename as input and checks if it is executable, if not make it executable
13	Create a Menu driven shell script which manages users and display following menu 1. create a new user 2. set passwords for new user or update passwords for existing users 3. Lock the user password 4. locate the home directory for the user and create its backup file with timestamp 5. View all the authentic users and their id
14	Write a shell script that calculates the sum of integers from 1 to N using a loop.
15	Create a script that searches for a specific word in a file and counts its occurrences.

Back End Engineering LAB (Based on 24BCAI-N-403)	0L:0T:2P	1 Credits
---	----------	-----------

1. Nodejs Installation and Intro to Modules [user defined and built-in]
2. Creating a basic HTTP Server and returning different types of content in Node.js
3. Handling Filestream
4. Creating a Node server with Express
5. Handling Static Pages with Filestream
6. Handling Exception
7. Example to implement callback and other methods of endpoints
8. Express: creating server using Express
9. Static Routing using express
10. Dynamic Routing using express
11. static and body parser middleware
12. Client state management using session module
13. Handling Exceptions during auth. and authentication
14. Multer: uploading single file
15. Multer: uploading multiple file
16. EJS Server side rendering
17. Create the EJS Partial and Pass Data to Views and Partial
18. MongoDB command-line Interface creating database, collection, different operations on Database
19. Connecting mongodb with nodejs application using sign up pages
20. Applying CRUD operations on database
21. Login Management using Mongoose with ATLAS
22. CRUD Operations using Mongoose with validation
23. Authentication using JSON using Web Tokens

Scheme of Examination for BCA (Industry Integrated)**Semester: 5th (w.e.f. Session 2024-25)**

S r · N o ·	Course/ Paper Code	Title of the Course	Mode of Learnin g	Course Type	Credit			Contact Hours per Week			Examination Scheme			Pract ical (Inte rnal)	Total Mark s
					Th eor y	Practical / Seminar	Tot al	Theo ry	Practi cal /Semi nar	Tot al	(End Semest er Exa m)	Int ernal Asse ssme nt	Prac tical (Exte rnal)		
1	24BCAI- N-501	Programming in Python	CL/BL	CC	3	1	4	3	2	5	50	20	20	10	100
2	24BCAI-N-502	Computer Network	CL/BL	CC	4	-	4	4	..	4	70	30			100
3	24BCAI-N-503	Artificial Intelligence	CL/BL	CC	4	-	4	4	..	4	70	30			100
4	24BCAI-N-504	Web Security	CL	MIC	3	-	3	3	..	3	50	20			70
5	24BCAI- N - SEC -501	Internship	CL	SEC	..	4	4	-.		..	70	30			100
	Total						19			16					470

CC - Core Course; SEC-Skill Enhancement CoursMIC-Minor course

24BCAI- N- 501/	Programming in Python	3L:0T:0P	3 Credits
-----------------	----------------------------------	----------	-----------

Maximum Marks:70
External Examination: 50
Internal Assessment: 20
Max. Time: 3 Hrs

Course Objectives

- To understand why Python is a useful scripting language for developers.
- To learn how to design and program Python applications.
- To learn how to use lists, tuples, and dictionaries in Python programs.
- To learn how to use exception handling in Python applications for error handling.

Note: There shall be nine questions in all. Question no. 1 shall be compulsory, consisting of eight short answer type questions covering the entire syllabus. Two questions will be asked from each unit. Students will have to attempt one question from each unit. Each question shall carry equal marks.

Unit I

Introduction to Python: Python variables, Python basic Operators, Understanding python blocks. Python Data Types, Declaring and using Numeric data types: int, float etc. Python Program Flow Control Conditional blocks: if, else and else if, Simple for loops in python, For loop using ranges, string, list and dictionaries. Use of while loops in python, Loop manipulation using pass, continue, break and else. Programming using Python conditional and loop blocks.

Unit II

Python Complex data types: Using string data type and string operations, Defining list and list slicing, Use of Tuple data type. String, List and Dictionary, Manipulations Building blocks of python programs, string manipulation methods, List manipulation. Dictionary manipulation, Programming using string, list and dictionary in-built functions. Python Functions, Organizing python codes using functions.

Unit III

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

Unit IV

Python packages: Simple programs using the built-in functions of packages matplotlib, numpy, pandas etc. GUI Programming: Tkinter introduction, Tkinter and Python Programming, Tk Widgets, Tkinter examples. Python programming with IDE.

Course Outcomes

A student will be able to:

- Study fundamental concepts of Python.
- Understand the concept of packages and file handling in Python programming.
- Understand and apply concepts related to Graphical User Interface (GUI) programming in Python.

Suggested Readings:

1. Wesley J. Chun, “Core Python Applications Programming”, 3rd Edition , Pearson Education, 2016
2. 2. Lambert, Fundamentals of Python: First Programs with MindTap, 2nd 1st edition , Cengage Learning publication
3. 3. Charles Dierbach, “Introduction to Computer Science using Python”, Wiley, 2015
4. 4. Jeeva Jose &P.SojanLal, “Introduction to Computing and Problem Solving with PYTHON”, Khanna Publishers, New Delhi,
5. 2016
6. 5. Downey, A. et al., “How to think like a Computer Scientist: Learning with Python”, John Wiley, 2015
7. 6. Mark Lutz, “Learning Python”, 5th edition, Orelly Publication, 2013, ISBN 978-1449355739
8. 7. John Zelle, “Python Programming: An Introduction to Computer Science”, Second edition, Course Technology Cengage
9. Learning Publications, 2013, ISBN 978- 1590282410

24BCAI-N-502	Computer Network	4L:0T:0P	4 Credits
--------------	------------------	----------	-----------

Maximum Marks:100
External Examination: 70
Internal Assessment: 30
Max. Time: 3 Hrs

Course Objectives

- The Purpose of the course is to understand the evolution of computer networks using the layered network architecture

Note: *There shall be nine questions in all. Question no. 1 shall be compulsory, consisting of eight short answer type questions covering the entire syllabus. Two questions will be asked from each unit. Students will have to attempt one question from each unit. Each question shall carry equal marks.*

Unit I

Introduction to Computer Networks : Data Communication System and its components, Types of computer networks, Wireless and Wired networks, Network topologies, protocols, interfaces and services, ISO- OSI reference model, TCP/IP architecture.

Physical Layer: Concept of Analog & Digital Signal, Bandwidth, Transmission Multiplexing : Frequency Division, Time Division, Wavelength Division, FDMA, TDMA, CDMA, Transmission Media: Twisted pair, Coaxial cable, Fiber optics, Wireless transmission (radio, microwave, infrared), Switching: Circuit Switching, Message Switching ,Packet Switching.

Unit II

Data link layer: Error Control, Types of errors, framing(character and bit stuffing), error detection & correction methods; Flow control; Protocols: Stop & wait ARQ, sliding window protocols, HDLC;

Medium access sub layer: Point to point protocol, FDDI, token bus, token ring; Reservation, polling, Multiple access protocols: Pure ALOHA, Slotted ALOHA, CSMA, CSMA/CD,, Traditional Ethernet, fast Ethernet, Network devices-repeaters, hubs, switches, Bridges, Router, Gateway

Unit III

Network layer: Addressing : Internet address, sub-netting; Routing techniques, static vs. dynamic routing , routing table, DHCP,

Routing algorithms: shortest path algorithm, flooding, distance vector routing, link state routing; Protocols: ARP, RARP, IP, IPV6;

Unit IV

Transport & Application Layer: Process to process delivery; UDP; TCP, Congestion control algorithm: bucket algorithmS, Quality of service: techniques to improve QoS.

Application layer: DNS; SMTP, SNMP, FTP, HTTP & WWW; Firewalls, Bluetooth, Email, S/MIME IMAP

Network Security: Cryptography, user authentication, security protocols in internet, public key encryption algorithm, digital signatures

Course Outcomes

A student will be able to:

- Describe the ISO-OSI reference model and TCP/IP architecture, and demonstrate an understanding of the physical layer, including analog and digital signals, bandwidth, transmission impairments, multiplexing techniques, and different transmission media.
- Analyze the data link layer and medium access sub layer protocols and techniques for error control, flow control, framing, and multiple access, as well as knowledge of various network devices..
- Comprehend and implement network layer concepts including addressing, routing techniques, routing algorithms, protocols, and standards
- Understand the functions of the Transport layer, including process-to-process delivery, UDP, TCP, and congestion control algorithms
- Analyze and evaluate various security protocols used in the internet to ensure data confidentiality, integrity, and availability.

Suggested Readings:

1. Larry L.Peterson, Peter S. Davie, “Computer Networks”, Elsevier, Fifth Edition, 2012.
2. William Stallings, “Data and Computer Communication”, Eighth Edition, Pearson Education, 2007
3. Behrouz A. Forouzan, “Data communication and Networking”, Tata McGraw Hill, Fourth Edition, 2011.
4. Computer Networks, 4th Edition, Pearson Education by Andrew S. Tanenbau

24BCAI-N-503	Artificial Intelligence	4L:0T:0P	4 Credits
--------------	-------------------------	----------	-----------

Maximum Marks:100
External Examination: 70
Internal Assessment: 30
Max. Time: 3 Hrs

Course Objectives

- The Purpose of the course is to introduce the principles and paradigms of programming languages for design and implement the software intensive systems.

Note: There shall be nine questions in all. Question no. 1 shall be compulsory, consisting of eight short answer type questions covering the entire syllabus. Two questions will be asked from each unit. Students will have to attempt one question from each unit. Each question shall carry equal marks.

Unit I

Introduction to artificial intelligence, Historical development and foundation areas of artificial intelligence,

Tasks and application areas of artificial intelligence.

Introduction, types and structure of intelligent agents, Computer Vision, Natural language processing.

Unit II

Introduction, Problem solving by searching, Searching for solutions, Uniformed searching techniques, Informed searching techniques, Local search algorithms, Adversarial search methods, Search techniques used in games, AlphaBeta pruning. Propositional logic, Predicate logic, First order logic, Inference in first order logic, Clause form conversion, Resolution.

Unit III

Chaining- concept, forward chaining and backward chaining, Utility theory and Probabilistic reasoning, Hidden Markov model, Bayesian networks. Introduction, types and application areas, Statistical learning methods- Regression Decision trees, Naïve Bayes models, Learning with hidden data- concept and EM algorithm, Reinforcement learning.

Unit IV

Introduction and design principles, Statistical pattern recognition, Parameter estimation methods - Principle component analysis and Linear discrimination analysis, Classification techniques - Nearest neighbour rule and Bayes classifier, K-means clustering, Support vector machine.

Course Outcomes

A student will be able to:

- Write the meaning of intelligence and study various intelligent agents.
- Apply AI searching algorithms in different problem domains.
- Analyze various models for knowledge representations.
- Demonstrate the basic concepts of machine learning to analyze and implement widely used learning methods and algorithms.
- Outline the knowledge of pattern recognition.

Suggested Readings:

1. Khemani D., “A First Course in Artificial Intelligence”, McGraw Hill.
2. Winston P. H., “Artificial Intelligence”, Pearson Education.
3. Thornton C. and Boulay B., “Artificial Intelligence- Strategies, Applications and Models through Search”, New Age International Publishers.
4. Russell S. and Norvig P., “Artificial Intelligence – A Modern Approach”, Pearson Education.
5. Rich E. and Knight K., “Artificial Intelligence”, McGraw Hill Publications.
6. Charnik E. and McDermott D., “Introduction to Artificial Intelligence”, Pearson Education.

24BCAI-N504	Web Security	3L:0T:0P	3 Credits
-------------	--------------	----------	-----------

Maximum Marks:70
External Examination: 50
Internal Assessment: 20
Max. Time: 3 Hrs

Course Objectives

- Gain a comprehensive understanding of web security principles, including the importance of confidentiality, integrity, and availability.
- Explore secure authentication mechanisms and session management best practices to mitigate common vulnerabilities like session hijacking and fixation.
- Learn about common web vulnerabilities such as Cross-Site Scripting (CSS), Cross-Site Request Forgery (CSRF), and SQL Injection.
- Learn about security testing methodologies, tools, and techniques for vulnerability assessment and penetration testing.

Note: There shall be nine questions in all. Question no. 1 shall be compulsory, consisting of eight short answer type questions covering the entire syllabus. Two questions will be asked from each unit. Students will have to attempt one question from each unit. Each question shall carry equal marks.

Unit I

Introduction to Web Security, Importance of web security, Basic security concepts: confidentiality, integrity, availability. Overview of web application architecture, Common security terminology

Types of attackers and their motives, Common attack vectors, Case studies of recent web security breaches

Authentication and Session Management, Secure authentication mechanisms, Password policies and storage techniques, multi-factor authentication, Session management best practices, Common vulnerabilities (e.g., session hijacking, fixation)

Unit II

Cross-Site Scripting (XSS)-Types of XSS: reflected, stored, DOM-based, Detection and exploitation techniques, Preventive measures: input validation, output encoding

Cross-Site Request Forgery (CSRF)-Understanding CSRF attacks, Defense mechanisms: anti-CSRF tokens, Same Site cookies.

SQL Injection-Basics of SQL injection, Types of SQL injection: in-band, inferential, out-of-band, Detection techniques and tools, Mitigation strategies

.

Unit III

Secure Coding Practices- Principles of secure coding, Common coding flaws and how to avoid them using secure libraries and frameworks, Code reviews and static analysis tools

Security Testing and Vulnerability Assessment, Types of security testing: black-box, white-box, gray-box

Tools for security testing (e.g., Burp Suite, OWASP ZAP), Conducting penetration tests, Reporting and remediating vulnerabilities.

Unit IV

HTTPS and Secure Communication- Importance of HTTPS, How SSL/TLS works, Certificate authorities and chain of trust, Implementing HTTPS in web applications, Common pitfalls and best practices, Content Security Policy (CSP) and Security Headers, Web Application Firewalls (WAF) and Security Tools, Machine learning and AI in web security.

Course Outcomes

A student will be able to

- Develop a solid understanding of web security fundamentals, including confidentiality, integrity, and availability, and their importance in web application development.
- Understand best practices for password policies, storage techniques, and multi-factor authentication, ensuring robust security measures in authentication processes.
- Develop skills in identifying and remediating vulnerabilities through effective coding practices, testing methodologies, and security tools.
- Apply secure coding practices and conduct security testing using tools like Burp Suite and OWASP ZAP to identify and address vulnerabilities in web applications.

Suggested Readings:

1. "Web Application Security, A Beginner's Guide" by Bryan Sullivan and Vincent Liu.
2. "The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws" by Dafydd Stuttard and Marcus Pinto
3. "Computer Security: Principles and Practice" by William Stallings and Lawrie Brown.

Python Programming Lab (Based on 24BCAI- N- 501)	0L:0T:2P	1 Credits
---	----------	-----------

Maximum Marks:30
External Examination: 20
Internal Assessment: 10

Lab 1: Basics of python

- Write a Python program to check if a given year is a leap year or not.
- Write a Python program to generate the Fibonacci series up to n terms.

Lab 2: Basics

- Declare a variable num_complex and assign it a complex number. Perform complex arithmetic operations like addition, subtraction, multiplication, and division with num_complex.
- Write a Python program to implement the Tower of Hanoi problem using recursion.
- Implement a Python program to generate Pascal's triangle up to a given number of rows.

Lab 3: String

- Write a Python function count_vowels that takes a string as input and returns the count of vowels (both uppercase and lowercase) present in the string.
- Write a Python function reverse_words that takes a sentence as input and returns the sentence with each word reversed.
- Write a Python function find_max_min that takes a list of numbers as input and returns a tuple containing the maximum and minimum numbers in the list.
- Write a Python function remove_duplicates that takes a list as input and returns a new list with duplicate elements removed while preserving the original order.

Lab 4: Dictionaries

- Write a Python function **merge_dicts** that takes two dictionaries as input and returns a new dictionary containing the merged key-value pairs from both dictionaries. If there are common keys, prioritize the values from the second dictionary.
- Write a Python function **flatten_dict** that takes a nested dictionary as input and returns a flattened dictionary where each key is a tuple representing the nested structure and the value is the corresponding value from the original dictionary.

Lab 5: Strings

- Write a Python program that takes a sentence from the user, counts the frequency of each word, and prints the word-frequency pairs sorted by frequency in descending order.
- Write a Python program that takes a list of strings as input, converts each string to uppercase, removes any duplicates, and prints the modified list.

Lab 6: Files

- Write a Python program that reads the first 50 characters from a text file named "data.txt" and prints them to the console.
- Write a Python program that reads the contents of a text file named "numbers.txt" containing integers separated by spaces, converts them to a list of integers, and prints the sum of the numbers.
- Modify the program to read the contents of the file line by line and print the sum of each line.

Lab 7: Files

- Write a Python program that takes a list of strings as input and writes each string to a new line in a text file named "strings.txt".
- Modify the program to write the list of strings as a single line separated by commas in the "strings.txt" file.

Lab 8: Matplotlib

- Write a Python program using Matplotlib to plot a simple line graph showing the trend of temperature over a week. Use lists to represent days and corresponding temperatures.
- Extend the program to save the plot as a PNG file named "temperature_plot.png".

Lab 9: Pandas

- Write a Python program using Pandas to read a CSV file named "data.csv" containing information about students (e.g., name, age, grade) and display the first 5 rows of the DataFrame.
- Modify the program to calculate the average age of the students and print it.

Lab 10: Tkinter Examples

- Write a Python program using Tkinter to create a calculator application with buttons for digits, arithmetic operations, and a display for the result.
- Write another Python program using Tkinter to create a simple login form with entry widgets for username and password, and a button to submit the credentials.

Lab 11: Tkinter

- Write a Python program using Tkinter to create a simple window with a label widget displaying the text "Hello, Tkinter!".
- Add a button widget labeled "Click Me" to the window.

Scheme of Examination for BCA (Industry Integrated)Semester: 6th (w.e.f. Session 2024-25)

S r · N o ·	Course/ Paper Code	Title of the Course	Mode of Learnin g	Cours e Typ e	Credit			Contact Hours per Week			Examination Scheme			Total Mark s
					Theo ry	Practical / Seminar	Tot al	Theo ry	Practi cal /Semi nar	Tot al	Major Test (End Semest er Exa m)	Internal Assessme nt	Practical (External)	
1	24BCAI-N-601	Internship		CC	..	20	20	..	..	..	400	50		450
	Total						20							450

CC - Core Course

Note: Students will have to submit Internship Report after successful completion of Internship.