

KRISHNA UNIVERSITY**Prof. K. Sobhan Babu****Registrar**

Rudravaram

Machilipatnam-521004.

Mobile: 7382483829

E-mail : registraraku@gmail.com



No. KRU/AAC/Reconstitution of the BOS for BCA (General) /2024

Dt: 30-09-2024

Sub: KU – Academic Audit – Reconstitution of the UG Board of Studies (BOS) in BCA (General) for the approval of the Under Graduation (UG) CBCS syllabus w.e.f 2023-24 – Reg.

Read: Note order's of the Vice-Chancellor, dt: 28-09-2024

~oOo~

ORDER:

In accordance with the provisions contained by the Krishna University act No:29 of 2008 of A.P. Govt., Hon'ble Vice-Chancellor is pleased to appoint the UG Board of Studies in the Faculty of B.Sc. Bachelor of Computer Applications (BCA) General with the following members.

UG BOARD OF STUDIES FOR B.Sc. BACHELOR OF COMPUTER APPLICATIONS (BCA) GENERAL		
S.No	Name of the Faculty	Designation
1	Dr.J. Saradha Lakshmi, Lecturer in Computer Science, SRR & CVR Government Degree College, Vijayawada Ph: 9246660822, E-Mail: jsl@srrcvr.ac.in	
2	Sri. K. Sridhar, B. Siddhartha College of Arts & Science, Vijayawada, Ph: 9849965036, E-mail: sridharkavuri@yahoo.com	
3	Sri. P.L.Ramesh, K.B.N College, Vijayawada, Ph: 9848850252, E-mail: rameshplus@yahoo.com	
4	T. Nagaprasadarao HOD in Computer Science .MCA.M.Tech, A.G&S.G. Siddhartha College of Arts & Science, Vuyyuru Ph: 98668 03938, E-Mail: t.nagaprasadarao@gmail.com.	
6	Y. Neelima (MCA) Sri Krishnaveni Degree Kalasala-194 Ph. 8341511341, E - Mail id: neelimayaram@gmail.com	 Industry Expert
7	Prof. Y.K. Sundara Krishna, Department of Computer Science, University College of Arts and Science, Krishna University, Machilipatnam Ph: 9440226744, E-Mail: yksk2010@gmail.com	University Nominee
8	One Student from SRR & CVR GDC(A), VIJAYAWADA	Student Member

The term of office of all the members including the chairperson of Board of studies is until further orders with effect from the date of the order or until the new boards of studies is constituted.

These orders shall come into force with immediate effect.

(BY ORDER)

Sd/-

REGISTRAR

Copy to:

The Chairperson & all members of UG Board of Studies of the above

PA's to Vice-Chancellor/Registrar KRU

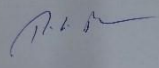
Accounts Section/ Establishment Section & File.

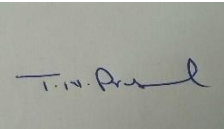
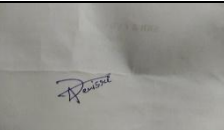
Krishna University: Machilipatnam
Board of Studies Meeting

Minutes of the meeting of the UG Board of Studies on the subject of
B.C.A. (HONORS) Computer Applications (Major) w.e.f., 2025-2026

The meeting of the UG Board of Studies in the subject of B.C.A. (HONORS) Computer Applications under a single Major/Minor Programme convened at 1.30 P.M., on 14-11-2025 through online mode with the members of the Board of Studies for Bachelor of Computer Applications (BCA) as mentioned in the proceedings No. KRU/AAC/Reconstitution of the BOS for BCA (General) /2024 dated 30.09.2024.

THE FOLLOWING MEMBERS ATTENDED THE ONLINE MEETING

UG BOARD OF STUDIES IN BACHELOR OF COMPUTER APPLICATIONS (BCA) GENERAL		
Name of the Faculty	Designation	
Dr.J. Saradha Lakshmi, Lecturer in Computer Science, SRR & CVR Government Degree College, Vijayawada Ph: 9246660822, E-Mail: jsl@srrcvr.ac.in	Chairperson	
Sri. K. Sridhar, B. Siddhartha College of Arts & Science, Vijayawada, Ph: 9849965036, E-mail: sridharkavuri@yahoo.com	Member	
Sri. P.L.Ramesh, K.B.N College, Vijayawada, Ph: 9848850252, E-mail: rameshplus@yahoo.com	Member	

T. Naga Prasad Rao HOD in Computer Science .MCA.M.Tech, A.G&S.G. Siddhartha College of Arts & Science, Vuyyuru Ph: 98668 03938, E-Mail: t.nagaprasadarao@gmail.com	Member	
Y. Neelima (MCA) Sri Krishnaveni Degree Kalasala-194 Ph. 8341511341, E - Mail id: neelimayaram@gmail.com	Member	
Prof. Y.K. Sundara Krishna, Department of Computer Science, University College of Arts and Science, Krishna University, Machilipatnam Ph: 9440226744, E-Mail: yksk2010@gmail.com	University Nominee	
One Student from SRR & CVR GDC(A), Vijayawada	Student Member	

Agenda:

Item 1: Approval of Syllabus for I, II, III, and IV semesters for the academic year 2025-2026 according to Curriculum Framework of B.A/B.Com/BBA (Honours) from the A.Y. 2025-26 for **BACHELOR OF COMPUTER APPLICATIONS (BCA) GENERAL Major and Minor**

Item 2: Approval of Blueprint and Model Question Papers and Question Bank.

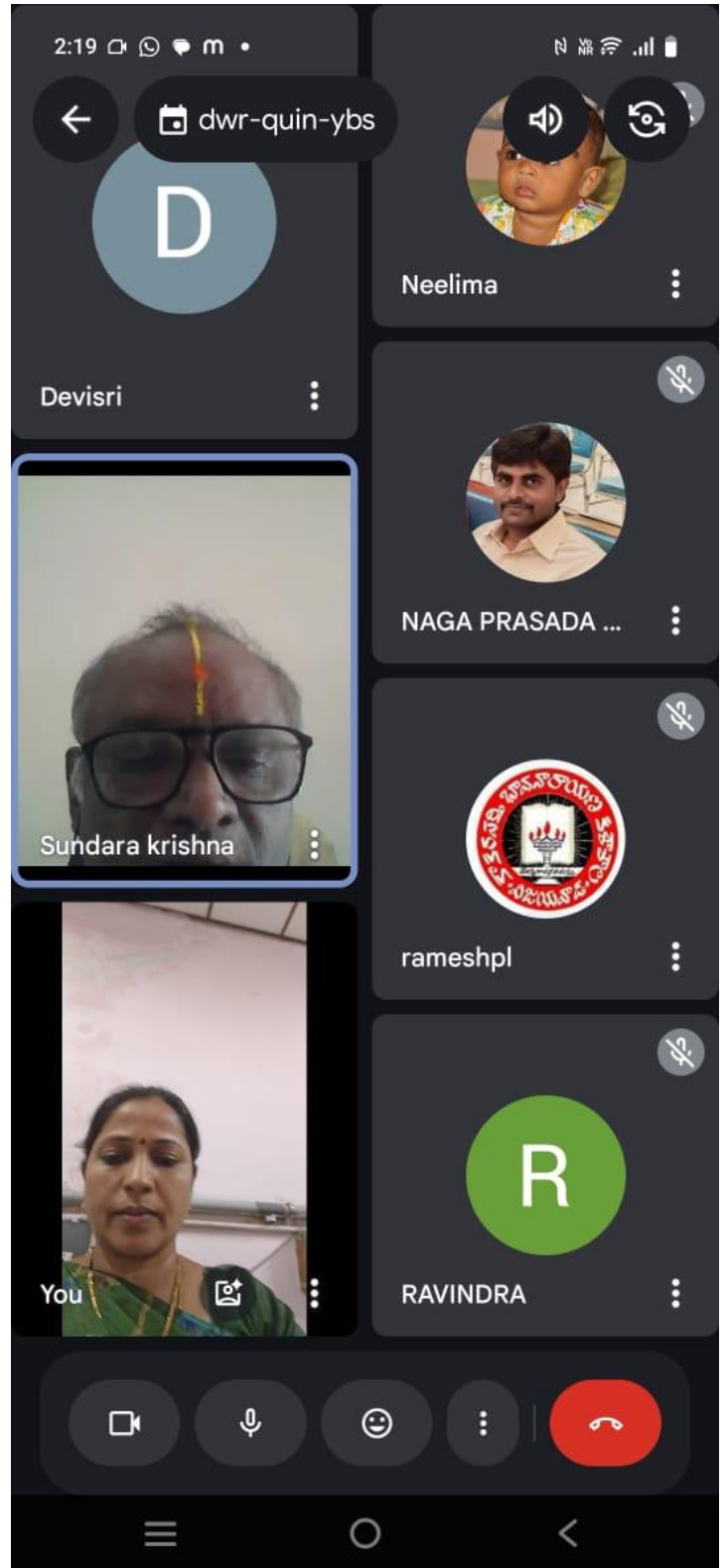
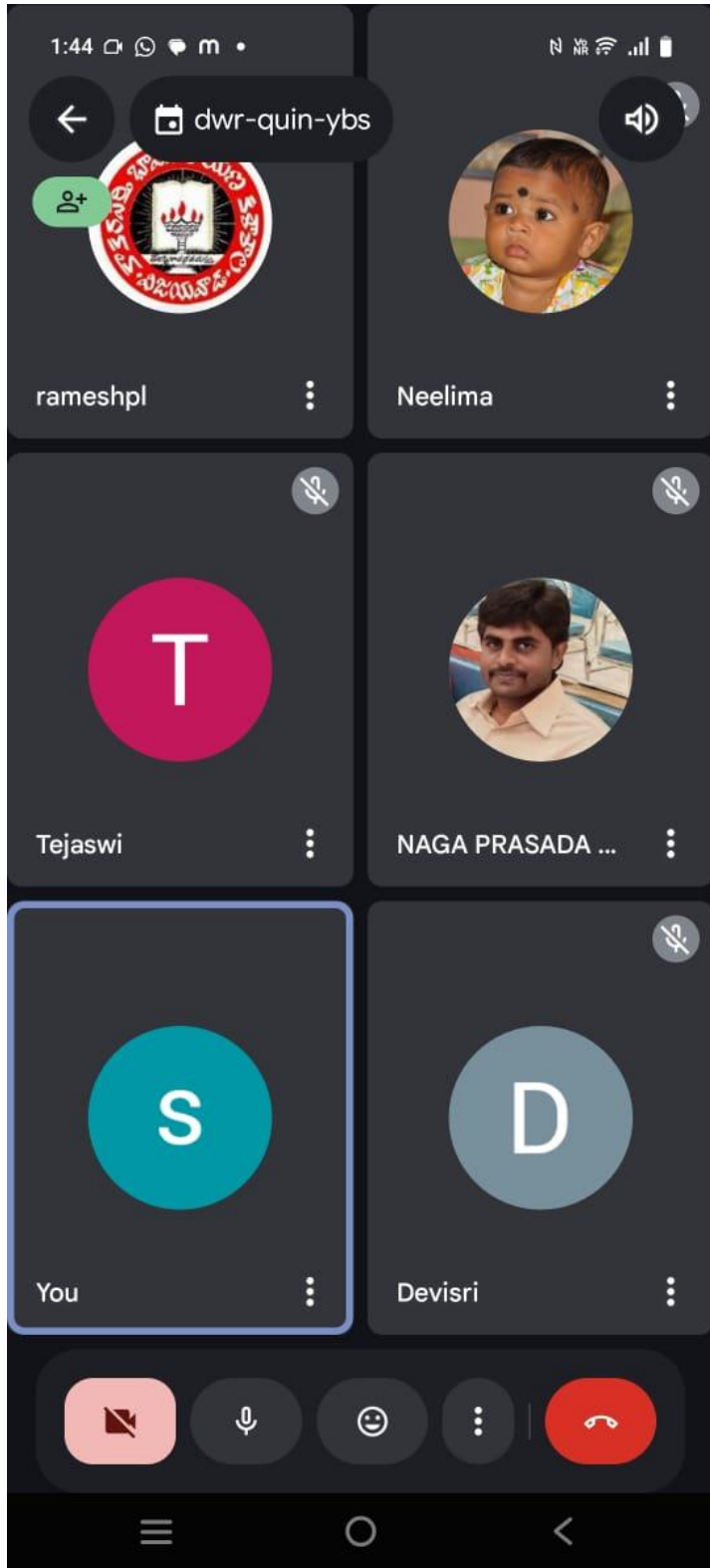
Item 3: Approval of Division of marks for Continuous Internal Assessment and Semester End Examination

Item 4: Any other related matters for the BCA general UG program, for I, II, III, and IV semesters with the approval of the Chair.

Resolutions:

The Board of Studies for Bachelor of Computer Applications (BCA) General of Krishna University members held an online meeting on 14-11-2025 AN and made the following resolutions unanimously.

1. The Board of Studies resolved to adopt and implement the syllabi for the core subjects as appended here under I, II, III & IV semesters only.
2. The Board of Studies unanimously approved 30 marks allocated for Continuous Internal Assessment in the colleges and 70 for Semester End Examination to be held by Krishna University.
3. The Board of Studies unanimously resolved the model question paper with its scheme of distribution of marks as laid down here.
4. The Board of Studies also came out with a model question paper as envisaged in the blueprint of the Question Paper attached herewith.



1:51 [status icons]

✓ ↶ ↷ A+ + [comment icon] ⋮

Presentation Tools: Using PowerPoint/Google Slides – slide design, animations, transitions.

Applications: Creating resumes, reports, brochures, and presentations.

Keyboard Shortcuts

Unit 4. Spreadsheet Basics:

Spreadsheet Concepts: Understanding rows, columns, cells in tools like MS Excel/Google Sheets.

Functions and Formulas: SUM, AVERAGE, IF, COUNT.

Charts and Graphs: Creating visual representation

Data Handling

Formulas

[B] [I] [U]

[grid icon]

1 2 3 4 5 6 7 8 9 0

q w e r t y u i o p

a s d f g h j k l

[home icon] z x c v b n m [delete icon]

?123 [emoji icon] [spacebar] [arrow down icon] [back arrow icon]

[floating window: S You, R RAVINDRA, +7]

2:13 [status icons]

← dwr-quin-ybs 🔊



Neelima ⋮

NAGA PRASADA ... ⋮ rameshpl ⋮

You're sharing your screen

[Stop]

[R] RAVINDRA

RAVINDRA Return to top

[stop icon] [microphone icon] [emoji icon] [three dots icon] [hang up icon]

[bottom navigation bar: hamburger, circle, back arrow]

Krishna University: Machilipatnam
B.C.A. (HONORS) Computer Applications (Major)
w.e.f. AY 2025-26
COURSE STRUCTURE

COURSE STRUCTURE (for Semester I to VI)

Year	Semester	Course	Title of the Course	No. of Hrs /Week	No. of Credits
I	I	1	Computer Fundamentals and Office Automation	3	3
			Computer Fundamentals and Office Automation-Practical	2	1
		2	Problem Solving Using C	3	3
			Problem Solving Using C-Practical	2	1
	II	3	Data Structures using C	3	3
			Data Structures using C-Practical	2	1
		4	Database Management System	3	3
			Database Management System-Practical	2	1
II	III	5	OOPS Through JAVA	3	3
			OOPS Through JAVA-Practical	2	1
		6	Computer Organisation	3	3
			Computer Organisation-Practical	2	1
		7	Probability and Statistics	4	4
	IV	8	Operating Systems	3	3
			Operating Systems-Practical	2	1
		9	Python Programming	3	3
			Python Programming-Practical	2	1
		10	Software Engineering	3	3
			Software Engineering-Practical	2	1
III	V	11	Data Mining	3	3
			Data Mining-Practical	2	1

Year	Semester	Course	Title of the Course	No. of Hrs /Week	No. of Credits
		12 A	Web Interface Design Technologies	3	3
			Web Interface Design Technologies-Practical	2	1
		OR			
		12 B	Computer Networks	3	3
			Computer Networks Lab-Practical	2	1
		13 A	Web Application Development using PHP & MySQL	3	3
			Web Application Development using PHP & MySQL-Practical	2	1
		OR			
		13 B	Cyber Security	3	3
			Cyber Security-Practical	2	1
	VI	14 A	Mobile Application Development	3	3
			Mobile Application Development-Practical	2	1
		OR			
		14 B	Cloud Fundamentals and Security	4	4
		15 A	MERN Stack	3	3
			MERN Stack-Practical	2	1
		OR			
		15 B	Digital Forensics	3	3
			Digital Forensics-Practical	2	1

Note: In the III Year (during the V and VI Semesters), students are required to select a pair of electives from one of the **Two** specified domains. **For example: if set ‘A’ is chosen, courses 12 to 15 to be chosen as 12 A, 13 A, 14 A and 15 A.** To ensure in-depth understanding and skill development in the chosen domain, students must continue with the same domain electives in both the V and VI Semesters.

SEMESTER-I

COURSE 1: COMPUTER FUNDAMENTALS AND OFFICE AUTOMATION

Theory

Credits: 3

3 hrs/week

Course Objectives

1. **Understand foundational computing concepts**, including number systems, the evolution of computers, block diagrams, and generational progress.
2. **Develop knowledge of computer architecture**, focusing on system organization and networking fundamentals.
3. **Acquire practical skills in document creation**, formatting, and digital presentations using word processing tools.
4. **Gain proficiency in spreadsheet operations**, such as data entry, formulas, functions, and charting techniques.
5. **Introduce data visualization and basic modelling principles**, fostering analytical thinking in structuring and interpreting data sets.

Course Outcomes

1. At the End of the Course, The Students will be able to **explain different number systems**, the historical evolution of computers, and identify key components in a block diagram.
2. Learners will demonstrate **basic blocks of a computer and fundamental networking knowledge**.
3. Learners will create professional-level documents and **design visually appealing presentations** using word processing software and presentation software.
4. Learners will manipulate data within spreadsheets, apply formulas, and **generate accurate summaries and visualizations**.
5. Learners will apply data modelling techniques to **analyze, organize, and represent data effectively** in various scenarios.

Unit 1. Number Systems, Evolution, Block Diagram and Generations:

Number Systems: Binary, Decimal, Octal, Hexadecimal; conversions between number systems.

Evolution of Computers: History from early mechanical devices to modern-day systems.

Block Diagram of a Computer: Components like Input Unit, Output Unit, Memory, CPU (ALU + CU).

Generations of Computers: First to Fifth Generation – technologies, characteristics, examples.

Unit 2. Basic Organization and Network fundamentals:

Computer Organization: Functional components – Input/Output devices, Storage types, Memory hierarchy.

Types of Computers: Micro, Mini, Mainframe, and Supercomputers.

Networking Fundamentals: Definition, need for networks, types (LAN, WAN, MAN), topology (Star, Ring, Bus), IP Address, Domain Name, Web Browser, Email, WWW.

Unit 3. Word Processing and presentations:

Word Processing Basics: Using MS Word/Google Docs – formatting, styles, tables, mail merge.

Presentation Tools: Using PowerPoint/Google Slides – slide design, animations, transitions.

Applications: Creating resumes, reports, brochures, and presentations.

Keyboard Shortcuts

Unit 4. Spreadsheet Basics:

Spreadsheet Concepts: Understanding rows, columns, cells in tools like MS Excel/Google Sheets.

Functions and Formulas: SUM, AVERAGE, IF, COUNT.

Charts and Graphs: Creating visual representations

Data Handling: Sorting, filtering, conditional formatting.

Text Functions: LEFT, RIGHT, MID, LEN, TRIM, CONCAT, TEXTJOIN

Advanced Functions: Logical: IF, AND, OR, IFERROR, **Lookup:** VLOOKUP, HLOOKUP, XLOOKUP, INDEX, MATCH

Unit 5. Data Modelling:

Conditional Formatting: Custom rules, Color scales, Icon sets, Data bars

Data Analysis Tools: Pivot Tables and Pivot Charts, Data Validation (Drop-downs, Input Messages, Error Alerts), What-If Analysis: Goal Seek, Scenario Manager, Data Tables

Charts and Dashboards: Creating Interactive Dashboards, Using slicers with Pivot Tables, Combo Charts and Sparklines

Productivity Tips: Using Named Ranges, Freeze Panes, Split View

Textbooks:

1. **Fundamentals of Computers**, Reema Thareja, Oxford University Press, Second Edition
2. **Fundamentals of Computers**, V. Rajaraman – PHI Learning
3. **Introduction to Computers** by Peter Norton – McGraw Hill
4. **Microsoft Office 365 In Practice** by Randy Nordell – McGraw Hill Education

References:

1. **Excel 2021 Bible** by Michael Alexander, Richard Kusleika – Wiley
2. **Networking All-in-One For Dummies** by Doug Lowe – Wiley
3. **Microsoft Official Docs and Training:** <https://learn.microsoft.com>
4. **Google Workspace Learning Center:** <https://support.google.com/a/users/>

Activities:

Outcome: At the End of the Course, The Students will be able to **explain different number systems**, the historical evolution of computers, and identify key components in a block diagram.

Activity: Create a digital poster or infographic comparing number systems (binary, decimal, octal, hexadecimal) and illustrating the timeline of computer generations with key innovations.

Evaluation Method: Rubric-based assessment of the poster presentation on a 10-point scale focusing on:

- Accuracy of number system conversions
- Correct identification of block diagram components
- Visual organization and creativity

Outcome: Learners will demonstrate **basic blocks of a computer and fundamental networking knowledge**.

Activity: Design a concept map showing the internal architecture of a computer and types of networks (LAN, WAN, MAN), including devices and topologies.

Evaluation Method: Checklist-based peer review and instructor validation:

- Completeness of the map
- Correctness of networking concepts
- Use of appropriate terminology
- Logical flow and structure of the map

Outcome: Learners will create professional-level documents and **design visually appealing presentations** using word processing software and presentation software.

Activity: Prepare a formal report (e.g., project proposal) in a word processor and present it using a slide deck with transitions, embedded media, and design elements.

Evaluation Method: Performance-based evaluation using a 10-point scoring scale:

- Formatting and structure of the document
- Presentation aesthetics and clarity
- Communication skills during presentation

Outcome: Learners will manipulate data within spreadsheets, apply formulas, and **generate accurate summaries and visualizations.**

Activity: Analyze a dataset (e.g., student scores or sales data) using spreadsheet software. Apply formulas (SUM, AVERAGE, IF, VLOOKUP) and create relevant charts.

Evaluation Method: Practical test with a rubric:

- Correct use of formulas
- Accuracy of data summaries

Outcome: Learners will apply data modelling techniques to **analyze, organize, and represent data effectively** in various scenarios.

Activity: Prepare an interactive dashboard for a given data set using EXCEL.

Evaluation Method: Evaluation of the dashboard on a 10-point scoring scale:

- Presentation aesthetics and clarity
- Interactive Ness
- Communication skills during presentation

SEMESTER-I

COURSE 1: COMPUTER FUNDAMENTALS AND OFFICE AUTOMATION

Practical

Credits: 1

2 hrs/week

List of Experiments:

1. Demonstration of Assembling and Disassembling of Computer Systems.
2. Identify and prepare notes on the type of Network topology of your institution.
3. Prepare your resume in Word.
4. Using Word, write a letter to your higher official seeking 10-days leave.
5. Prepare a presentation that contains text, audio and video.
6. Using a spreadsheet, prepare your class Time Table.
7. Using a Spreadsheet, calculate the Gross and Net salary of employees (Min 5) considering all the allowances.
8. Generate the class-wise and subject-wise results for a class of 20 students. Also generate the highest and lowest marks in each subject.
9. Using IF, AND, OR, and IFERROR to Automate Grade Evaluation.
 - a. Create a table of student scores in different subjects.
 - b. Use IF to assign grades (A/B/C/Fail).
 - c. Use IFERROR to handle missing scores or invalid data.
10. Employee Database Search Using VLOOKUP, HLOOKUP, XLOOKUP, INDEX, and MATCH
 - a. Create a database of employees (Name, ID, Department, Salary).
 - b. Implement VLOOKUP to search by employee ID.
 - c. Use HLOOKUP to extract department heads by role.
 - d. Apply XLOOKUP for more flexible searches.
 - e. Use INDEX + MATCH as an alternative to VLOOKUP.
11. Sales Report Analysis Using Pivot Tables and Charts
 - a. Use a dataset of product sales (Product, Region, Date, Quantity, Revenue).
 - b. Create Pivot Tables to summarize data by region/product.
 - c. Insert Pivot Charts for visual analysis (e.g., bar, line).
 - d. Add slicers to make the dashboard interactive.
12. Designing a Data Entry Form with Drop-downs and Input Rules
 - a. Create a student registration form.
 - b. Add drop-down lists for course selection using Data Validation.
 - c. Add input messages to guide users.
 - d. Add error alerts for wrong entries.
13. Monthly Budget Planning using Goal Seek and Scenario Manager
 - a. Create a simple personal budget (income, expenses, savings).
 - b. Use Goal Seek to determine income needed to save a desired amount.
 - c. Use Scenario Manager to compare different budgeting scenarios (best/ worst/ realistic case).

- d. Create a one-variable Data Table to analyze how different expenses affect savings.

14. Dashboard Creation Using Combo Charts, Sparklines & Slicers

- a. Use existing sales or attendance data.
- b. Insert combo charts (e.g., column + line).
- c. Add sparklines to show trends.
- d. Use slicers with Pivot Tables to control dashboard elements.
- e. Finalize and format for interactivity.

KRISHNA UNIVERSITY: MACHILIPATNAM - 521004
B.C.A Honors Computer Applications [Major / Minor]
REVISED CBCS FRAMEWORK WITH EFFECT FROM 2025-2026 ADMITTED BATCH
Model Question Paper
COURSE 1: COMPUTER FUNDAMENTALS AND OFFICE AUTOMATION
SEMESTER: I

Time: 3 Hours

Max. Marks: 70

Section – A

Answer any FIVE questions. All questions carry equal marks.

(5 × 4 = 20 M)

1. Explain the different types of number systems with one example each.
2. Write short notes on the generations of computers and their characteristics.
3. Write short notes on Input and Output devices.
4. Define a computer network. Explain its need and importance.
5. What are the features of MS Word or Google Docs? Write any two applications.
6. Define a spreadsheet. What are rows, columns, and cells? Give examples.
7. Write short notes on Pivot Tables and their uses in data analysis.
8. What is conditional formatting? Mention different options available.

Section – B

Answer all questions. Each question carries equal marks.

(5 × 10 = 50 M)

9. a) Explain Binary, Decimal, Octal, and Hexadecimal number systems and show conversions between them with examples.

Or

b) Describe the evolution of computers from early mechanical devices to modern-day systems with examples.

10. a) Draw and explain the block diagram of a computer. Describe the functions of Input Unit, Output Unit, Memory, and CPU.

Or

b) Explain various types of networks (LAN, MAN, WAN) and network topologies (Star, Ring, Bus).

11. a) Explain the steps involved in creating and formatting a document in MS Word or Google Docs.

Or

b) What are presentation tools? Explain slide design, animations, transitions, and their importance in presentations.

12. a) Explain the concept of functions and formulas in spreadsheets with examples of SUM, AVERAGE, IF, and COUNT.

Or

b) Discuss different types of charts in MS Excel or Google Sheets with their applications in data analysis.

13. a) Explain Data Modelling using Pivot Charts and Dashboards. Give examples of their applications.

Or

b) Describe What-If Analysis tools – Goal Seek, Scenario Manager, and Data Tables – with examples.

KRISHNA UNIVERSITY :: MACHILIPATNAM - 521004

B.C.A Honours Computer Applications[Major / Minor]

REVISED CBCS FRAMEWORK WITH EFFECT FROM 2025-2026 ADMITTED BATCH

BLUEPRINT

COURSE 1: COMPUTER FUNDAMENTALS AND OFFICE AUTOMATION

Unit	S.A.Q Section :A (including choice) 4 Marks	E.Q Section: B (including choice) 10 Marks	Marks Allotted
I	2	2	28
II	2	2	28
III	1	2	24
IV	2	2	28
V	1	2	24
Total	8	10	132

SEMESTER-I

COURSE 2: PROBLEM SOLVING USING C

Theory

Credits: 3

3 hrs/week

Course Objectives:

1. Understand the fundamentals of computer programming, Apply structured problem-solving approaches using algorithms, flowcharts, and C programming constructs.
2. Develop efficient logic using decision-making, loop, and jump control statements.
3. Utilize derived data types like arrays and strings for modular program design.
4. Design and implement modular solutions using functions, recursive logic, pointer operations, and dynamic memory management.
5. Handle complex data structures including structures, unions, and text file operations.

Course Outcomes:

At the End of the Course, The Students will be able to:

1. Understand basic computing concepts, programming paradigms and write structured C programs.
2. Apply control flow statements to solve logical and repetitive tasks in C.
3. Implement arrays and string operations to manage and manipulate data efficiently.
4. Design modular code using functions, recursion, and appropriate parameter passing.
5. Utilize pointers and memory operations for effective data handling. Demonstrate competence in dynamic memory allocation and text file processing.

Unit 1. Introduction to computer programming:

Introduction, Types of software, Compiler and interpreter, Concepts of Machine level, Assembly level and high-level programming, Flowcharts and Algorithms, Fundamentals of C: History of C, Features of C, C Tokens-variables and keywords and identifiers, constants and Data types, Rules for constructing variable names, Operators, Structure of C program, Input /output statements in C-Formatted and Unformatted I/O

Unit 2. Control statements:

Decision making statements: if, if else, else if ladder, switch statements. Loop control statements: while loop, for loop and do-while loop. Jump Control statements: break, continue and goto.

Unit 3. Derived data types in C:

Arrays: One Dimensional arrays - Declaration, Initialization and Memory representation; Two Dimensional arrays -Declaration, Initialization and Memory representation. Strings: Declaring & Initializing string variables; String handling functions, Character handling functions

Unit 4. Functions:

Function Prototype, definition and calling. Return statement. Nesting of functions. Categories of functions. Recursion (Basic Concept only). Parameter Passing by address & by value. Local and Global variables. Storage classes: automatic, external, static and register.

Pointers: Pointer data type, Pointer declaration, initialization, accessing values using pointers. Pointer arithmetic, Pointers and arrays.

Unit 5. Dynamic Memory Management:

Introduction, Functions-malloc, calloc, realloc, free Structures: Basics of structure, structure members, accessing structure members, nested structures, array of structures, structure and functions, structures and pointers. Unions - Union definition; difference between Structures and Unions. Working with text files - modes: opening, reading, writing and closing text files.

Text Books:

1. Programming in ANSI C, E. Balagurusamy, Tata McGraw Hill, 6 th Edn,
2. Computer fundamentals and programming in C, Reema Theraja, Oxford University Press

Reference Books:

1. Let us C, Y Kanetkar, BPB publications
2. Head First C: A Brain-Friendly Guide, David Griffiths, Dawn Griffiths

Activities:

Outcome: Understand basic computing concepts, programming paradigms and write structured C programs.

Activity: Create a concept map of computing fundamentals and programming paradigms (procedural, structured, object-oriented). Then, they write a structured C program (e.g., a calculator or student grade system) using proper syntax, indentation, and modular design.

Evaluation Method: Rubric-based Code Review & Viva to check the

- The correctness of the concept map
- Correct use of structure (main + functions)
- Identification of paradigm used
- Code readability and documentation

Outcome: Apply control flow statements to solve logical and repetitive tasks in C.

Activity: Implement a program that solves a logic puzzle (e.g., number guessing game, pattern generation, or prime number finder) using if, switch, for, while, and do-while.

Evaluation Method: Automated Test Cases + Peer Review to check the

- Correct use of control statements
- Logical correctness of output

- Efficiency and edge case handling
- Peer feedback on clarity and logic

Outcome: Implement arrays and string operations to manage and manipulate data efficiently.

Activity: Build a program that stores and arranges student marks in ascending and descending order using arrays and performs string operations like concatenation, comparing, and formatting names.

Evaluation Method: Functional Demonstration + Code Walkthrough to check the

- Correct array and string usage
- Memory efficiency
- Handling of invalid inputs
- Explanation of sorting/searching logic

Activity:

- **Recursive Problem Solver**

Students write a modular program to solve a recursive problem (e.g., factorial, Fibonacci, or Tower of Hanoi) using functions with parameters and return values.

Evaluation Method:

- **Code Trace + Written Quiz**

- Correct function decomposition
- Proper parameter passing (by value/reference)
- Recursion depth and base case handling
- Quiz on tracing recursive calls

Outcome: Utilize pointers and memory operations for effective data handling. Demonstrate competence in dynamic memory allocation and text file processing.

Activity: Create a program that dynamically stores user input (e.g., survey responses) using pointers and writes/reads the data to/from a text file.

Evaluation Method: Memory Debugging + File I/O Assessment to check the

- Proper use of malloc, calloc, realloc, and free
- Pointer arithmetic and dereferencing
- File creation, reading, writing, and error handling
- Use of tools like Valgrind or manual memory trace (Optional for Unix flavours)

SEMESTER-I

COURSE 2: PROBLEM SOLVING USING C

Practical

Credits: 1

2 hrs/week

List of Experiments:

1. Write a program to check whether the given number is Armstrong or not.
2. Write a program to find the sum of individual digits of a positive integer.
3. Write a program to generate the first n terms of the Fibonacci sequence.
4. Write a program to find both the largest and smallest number in a list of integer values
5. Write a program to demonstrate change in parameter values while swapping two integer variables using Call by Value & Call by Address
6. Write a program to perform various string operations.
7. Write a program to search an element in a given list of values.
8. Write a program that uses functions to add two matrices.
9. Write a program to calculate factorial of given integer value using recursive functions
10. Write a program for multiplication of two N X N matrices.
11. Write a program to sort a given list of integers in ascending order.
12. Write a program to calculate the salaries of all employees using Employee (ID, Name, Designation, Basic Pay, DA, HRA, Gross Salary, Deduction, Net Salary) structure.
 - a. DA is 30 % of Basic Pay
 - b. HRA is 15% of Basic Pay
 - c. Deduction is 10% of (Basic Pay + DA)
 - d. Gross Salary = Basic Pay + DA+ HRA
 - e. Net Salary = Gross Salary - Deduction
13. Write a program to read / write the data from / to a file.
14. Write a program to reverse the contents of a file and store in another file.
15. Write a program to create Book (ISBN, Title, Author, Price, Pages, Publisher) structure and store book details in a file and perform the following operations
 - a. Add book details
 - b. Search a book details for a given ISBN and display book details, if available
 - c. Update a book details using ISBN
 - d. Delete book details for a given ISBN and display list of remaining Books

KRISHNA UNIVERSITY: MACHILIPATNAM - 521004

B.C.A Honors Computer Applications [Major / Minor]

REVISED CBCS FRAMEWORK WITH EFFECT FROM 2025-2026 ADMITTED BATCH

Model Question Paper

**COURSE 2: PROBLEM SOLVING USING C
SEMESTER-I**

Time: 3 Hours

Max. Marks: 70

Section – A

Answer any FIVE questions. All questions carry equal marks.

(5 × 4 = 20 M)

1. Define system software and application software with examples.
2. What are variables and constants in C? Give two examples for each.
3. Write short notes on the rules for naming variables in C.
4. Explain the use of the *break* and *continue* statements in C.
5. What are character arrays? Give an example.
6. Write a short note on recursion with an example.
7. What are storage classes in C?
8. List any four differences between structures and unions.

Section – B

Answer all questions. Each question carries equal marks.

(5 × 10 = 50 M)

9. a) Explain the difference between compiler, interpreter, and assembler.
Describe the concepts of machine-level, assembly-level, and high-level programming languages.
Or
b) What are flowcharts and algorithms? write an algorithm and flowchart to find the sum of first N natural numbers.
10. a) Explain the *if*, *if-else*, *else-if ladder*, and *switch* statements with examples.
Or
b) Explain different types of loops (*for*, *while*, *do-while*) with syntax and suitable examples.
11. a) Explain one-dimensional and two-dimensional arrays in C with declaration, initialization, and memory representation.
Or
b) Discuss various string handling functions in C such as *strlen()*, *strcpy()*, *strcmp()*, and *strcat()* with examples.
12. a) What is a pointer? Explain pointer declaration, initialization and relationship between pointers and arrays.
Or
b) Explain user-defined functions in C with examples.
13. a) Explain dynamic memory management in C using *malloc()*, *calloc()*, *realloc()*, and *free()* functions with syntax and examples.
Or
b) What are structures? Explain nested structures and array of structures with suitable examples.

KRISHNA UNIVERSITY :: MACHILIPATNAM - 521004

B.C.A Honours Computer Applications[Major / Minor]

REVISED CBCS FRAMEWORK WITH EFFECT FROM 2025-2026 ADMITTED BATCH

BLUEPRINT

COURSE 2: PROBLEM SOLVING USING C

Unit	S.A.Q Section :A (including choice) 4 Marks	E.Q Section: B (including choice) 10 Marks	Marks Allotted
I	2	2	28
II	2	2	28
III	1	2	24
IV	2	2	28
V	1	2	24
Total	8	10	132

SEMESTER-II

COURSE 3: DATA STRUCTURES USING C

Theory

Credits: 3

3 hrs/week

Course Objectives:

1. Understand the fundamentals of computer programming, Apply structured problem-solving approaches using algorithms, flowcharts, and C programming constructs.
2. Develop efficient logic using decision-making, loop, and jump control statements.
3. Utilize derived data types like arrays and strings for modular program design.
4. Design and implement modular solutions using functions, recursive logic, pointer operations, and dynamic memory management.
5. Handle complex data structures including structures, unions, and text file operations.

Course Outcomes:

At the End of the Course, The Students will be able to:

1. Understand basic computing concepts, programming paradigms and write structured C programs.
2. Apply control flow statements to solve logical and repetitive tasks in C.
3. Implement arrays and string operations to manage and manipulate data efficiently.
4. Design modular code using functions, recursion, and appropriate parameter passing.
5. Utilize pointers and memory operations for effective data handling. Demonstrate competence in dynamic memory allocation and text file processing.

Unit 1. Basic Concepts:

Algorithm: Definition and characteristics, Complexity analysis: Space Complexity, Time Complexity, Asymptotic Notations.

Introduction to Data structures: Definition, Types of Data structures, Abstract Data Types (ADT), Introduction to Linked Lists, Representation of linked lists in Memory, Comparison between Linked List and Array.

Unit 2. Linked Lists:

Types of Linked Lists - Singly Linked list, Doubly Linked list, Circularly Singly Linked list, Circularly Doubly Linked list; Implementation of Single Linked List ADT: Creating a List, Traversing a linked list, Searching in linked list, Insertion and deletion into linked list (At first Node, Specified Position, Last node).

Unit 3. Stacks and Queues:

Introduction to stack ADT, Implementation of stacks using array and Linked List, Application of stacks - Polish Notations - Converting Infix to Post Fix Notation - Evaluation of Post Fix Notation.

Queues: Introduction to Queue ADT, Implementation of Queues using array and Linked List, Application of Queues Types of Queues- Circular Queues, De-queues, Priority Queue, Heaps.

Unit 4. Searching and Sorting:

Linear or Sequential Search, Binary Search, Hashing and collision resolution.

Sorting: Selection Sort, Bubble Sort, Insertion Sort, Quick Sort and Merge Sort

Unit 5. Trees and Graphs:

Tree Terminology, Binary Tree Representation, Traversal techniques, Expression Tree, Binary Search Tree- Definition, Operations on a Binary Search Tree: Creation, Search, Insertion & deletion.

Graphs: Introduction to Graphs, Terminology, Representation (Adjacency Matrix, Adjacency List), Traversal of Graphs (DFS, BFS), Applications of Graphs, Concept of Shortest Path Problems, Concept of Minimum Cost Spanning Tree

Textbooks:

1. Data Structures Using C, Balagurusamy E. Tata McGraw Hill
2. Data Structures using C, Reema Thareja, Third Edition, Oxford University Press

Reference Books:

1. Data Structures, Lipschutz, Schaum's Outline Series, Tata Mcgraw-hill
2. Data Structures Using C, Ch. Vijay Kumar, Pen Press International

Activities:

Outcome: Explain algorithm characteristics, time and space complexity, and asymptotic notations with clarity

Activity: Create a comparative chart of algorithms with different notations related to time and space complexities.

Evaluation Method: Rubric-based assessment of the chart for correctness, clarity, and depth of explanation on a 10-point scale.

Outcome: Implement and analyze different types of linked lists, including insertion, deletion, and traversal operations

Activity: Code a menu-driven program in C to implement single linked lists with all basic operations.

Evaluation Method: Practical lab assessment with test cases and Viva-style questioning to explain pointer manipulation.

Outcome: Develop stack and queue data structures using arrays and linked lists, and apply them in expression evaluation

Activity: Build a program to convert infix expressions to postfix and evaluate them using stacks; Implement queues using both arrays and linked lists with enqueue/dequeue operations.

Evaluation Method: Code review and execution of programs for sample cases and evaluation based on correctness and efficiency.

Outcome: Apply efficient searching and sorting algorithms to solve computational problems and evaluate performance trade-offs

Activity: Implement and compare sorting algorithms (e.g., selection sort and bubble sort) and searching algorithms (e.g., Linear vs. Binary Search) on datasets of varying sizes. Record number of swaps and iterations for preparing a chart to assimilate the results.

Evaluation Method: Performance report with graphs and analysis. Oral presentation or peer review discussing trade-offs and algorithm selection rationale.

Outcome: Construct and traverse tree and graph structures, using them to solve problems like shortest path and spanning trees

Activity: Implement binary trees and graphs using adjacency lists/matrices.

Evaluation Method: Lab demo with sample inputs and visual output (e.g., tree traversal order, graph paths).

SEMESTER-II

COURSE 3: DATA STRUCTURES USING C

Practical

Credits: 1

2 hrs/week

List of Experiments

1. Write a program to read 'N' numbers of elements into an array and also perform the following operation on an array
 - a. Add an element at the beginning of an array
 - b. Insert an element at given index of array
 - c. Update an element using a values and index
 - d. Delete an existing element
2. Write a program to implement Single Linked List with insertion, deletion and traversal operations
3. Write a program to implement Doubly Linked List with insertion, deletion and traversal operations
4. Write a program to implement the Stack operations using Arrays and Linked Lists.
5. Write a program to convert a given infix expression to a postfix expression using stacks.
6. Write a program to implement the Queue operations using Arrays and Linked Lists.
7. Write a program to implement the Circular Queue operations using Arrays.
8. Write a program for Binary Search Tree Traversals
9. Write a program to search an item in a given list using the following Searching Algorithms
 - a. Linear Search
 - b. Binary Search.
10. Write a program for implementation of the following Sorting Algorithms
 - a. Bubble Sort
 - b. Insertion Sort
 - c. Quick Sort
 - d. Merge Sort

KRISHNA UNIVERSITY: MACHILIPATNAM - 521004

B.C.A Honors Computer Applications [Major / Minor]

REVISED CBCS FRAMEWORK WITH EFFECT FROM 2025-2026 ADMITTED BATCH

Model Question Paper

COURSE 3: DATA STRUCTURES USING C

Semester: III

Time: 3 Hours

Max. Marks: 70

Section – A

Answer any FIVE questions. All questions carry equal marks.

(5 × 4 = 20 M)

1. Define an algorithm and list its characteristics.
2. Compare linked lists and arrays.
3. Write a C code for creation of single linked list.
4. Explain how to traverse a linked list.
5. Explain applications of stacks.
6. Write short notes on circular queues and their applications.
7. Discuss linear search algorithm?
8. Write the concept of Minimum cost spanning tree.

Section – B

Answer all questions. Each question carries equal marks.

(5 × 10 = 50 M)

9. a) Explain the concept of algorithm complexity and discuss the types of asymptotic notations with examples.

Or

b) Define data structures. Explain different types of data structures with suitable examples.

10. a) Describe insertion and deletion operations in a singly linked list with algorithms.

Or

b) Describe the creation and traversal operations of a linked list.

11. a) What is a stack? Explain its implementation using arrays .

Or

b) Explain the conversion of infix expression to postfix expression using stacks with an example.

12. a) Explain quick sort sorting algorithm— with example.

Or

b) What is hashing? Explain hashing techniques with examples.

13. a) Explain binary tree terminology and discuss tree traversal techniques (Inorder, Preorder, Postorder) with examples.

Or

b) What is a graph? Explain graph traversal techniques (DFS, BFS).

KRISHNA UNIVERSITY :: MACHILIPATNAM - 521004

B.C.A Honours Computer Applications[Major / Minor]

REVISED CBCS FRAMEWORK WITH EFFECT FROM 2025-2026 ADMITTED BATCH

BLUEPRINT

COURSE 3: DATA STRUCTURES USING C

Unit	S.A.Q Section :A (including choice) 4 Marks	E.Q Section: B (including choice) 10 Marks	Marks Allotted
I	2	2	28
II	2	2	28
III	1	2	24
IV	2	2	28
V	1	2	24
Total	8	10	132

SEMESTER-II

COURSE 4: DATABASE MANAGEMENT SYSTEMS

Theory

Credits: 3

3 hrs/week

Course Objectives:

1. To understand the fundamentals of data, information, and the evolution from file-based systems to modern database management systems.
2. To develop the ability to design conceptual data models using Entity-Relationship (ER) and Enhanced ER diagrams.
3. To explore relational model principles, such as keys, integrity constraints, relational algebra and calculus, and normalization.
4. To perform data definition and manipulation using SQL commands including queries, joins, subqueries, views, and set operations.
5. To apply procedural logic using PL/SQL, incorporating control structures, functions, procedures, and database triggers.

Course Outcomes:

At the End of the Course, The Students will be able to:

1. **Describe** the fundamentals of data, database systems, and the differences between file-based and database approaches. **Compare and classify** various DBMS architectures, data models, and their components, including the three-schema architecture.
2. **Design** conceptual data models using Entity-Relationship and Enhanced ER diagrams, applying generalization, specialization, and constraints.
3. **Apply** relational model concepts, including Codd rules, relational algebra, relational calculus, and normalization techniques.
4. **Construct and execute** SQL queries for data definition, manipulation, aggregation, joining, and subqueries, including views and set operations.
5. **Develop** PL/SQL programs incorporating control structures, procedures, functions, and triggers to manage database behavior effectively.

Unit 1. Overview of Database Management System:

Introduction to data, information, database, database management systems, file-based system, Drawbacks of file-Based System, database approach, Classification of Database Management Systems, advantages of database approach, Various Data Models, Components of Database Management System, three schema architecture of data base, costs and risks of database approach.

Unit 2. Entity-Relationship Model:

Introduction, the building blocks of an entity relationship diagram, classification of entity sets, attribute classification, relationship degree, relationship classification, reducing ER diagram to tables, enhanced entity-relationship model (EER model), generalization and specialization, IS A relationship and attribute inheritance, multiple inheritance, constraints on specialization and generalization, advantages of ER modeling.

Unit 3. Relational Model:

Introduction, CODD Rules, relational data model, concept of key, relational integrity, relational algebra, relational algebra operations, advantages of relational algebra, limitations of relational algebra, Functional dependencies and normal forms.

Unit 4. Structured Query Language:

Introduction, Commands in SQL, Data Types in SQL, Data Definition Language, Selection Operation, Projection Operation, Aggregate functions, Data Manipulation Language, Table Modification Commands, Join Operation, Set Operations, View, Sub Query.

Unit 5. PL/SQL:

Introduction, Shortcomings of SQL, Structure of PL/SQL, PL/SQL Language Elements, Data Types, Operators Precedence, Control Structures, Steps to Create a PL/SQL, Program, Iterative Control, Procedures, Functions.

Textbooks:

1. Database System Concepts, Avi Silberschatz, Henry F. Korth, S. Sudarshan, Seventh Edition, McGraw-Hill
2. Database Management Systems by Raghu Ramakrishnan, McGrawhill

Reference Books:

1. Fundamentals of Database Systems, Elmasri Navathe Pearson Education
2. An Introduction to Database systems, C.J. Date, A.Kannan, S.Swami Nadhan, Pearson

Activities:

Outcome: Describe the fundamentals of data, database systems, and the differences between file-based and database approaches. Compare and classify various DBMS architectures, data models, and their components, including the three-schema architecture.

Activity: Create a comparative presentation or infographic illustrating:

- File-based vs. DBMS approaches
- Types of DBMS architectures (1-tier, 2-tier, 3-tier)
- Data models and the three-schema architecture

Evaluation Method: Rubric-based assessment of the presentation covering clarity, accuracy, and depth of comparison. Include a short quiz to test conceptual understanding.

Outcome: Design conceptual data models using Entity-Relationship and Enhanced ER diagrams, applying generalization, specialization, and constraints.

Activity: Model a university or hospital database using ER and Enhanced ER diagrams that shows:

- Entity sets, relationships
- Generalization/specialization
- Participation and cardinality constraints

Evaluation Method: Diagram submission with peer review and instructor feedback. Use a checklist to assess completeness, correctness, and notation usage.

Outcome: Apply relational model concepts, including CODD rules, relational algebra, relational calculus, and normalization techniques.

Activity: Normalize a given unstructured dataset up to 3NF. Then, write relational algebra expressions for sample queries.

Evaluation Method: Written assignment graded on:

- Correctness of normalization steps
- Accuracy of relational algebra expressions
- Short-answer questions on CODD rules and relational calculus

Outcome: Construct and execute SQL queries for data definition, manipulation, aggregation, joining, and subqueries, including views and set operations.

Activity:

- Create tables using DDL (CREATE TABLE) for Students, Courses, Enrollments, Faculty.
- Fetch aggregated data (e.g., average marks, total students).
- Perform joins (e.g., list students enrolled in each course).
- Execute subqueries (e.g., students scoring above average).
- Create and use views (e.g., StudentCourseSummary).
- Use set operations (UNION, INTERSECT) for combined results.

Evaluation Method:

- Appropriate use of data types, primary & foreign keys, normalization.
- Correct syntax and execution of table creation and data insertion.
- Use of GROUP BY, JOIN, and multi-table queries with accurate output.
- Correct implementation of UNION, INTERSECT, EXCEPT.
- Writing correlated/un-correlated subqueries and reusable views.

SEMESTER-II

COURSE 4: DATABASE MANAGEMENT SYSTEMS

Practical

Credits: 1

2 hrs/week

Experiment 1 : Database: Inventory Management

Table 1: Products

Structure:

Column Name	Data Type	Constraints
product_id	INT	PRIMARY KEY
product_name	VARCHAR(50)	NOT NULL
price	DECIMAL(10,2)	CHECK(price > 0)
stock_qty	INT	CHECK(stock_qty >= 0)

Sample Data:

product_id	product_name	price	stock_qty
1	Pen	10.00	100
2	Notebook	50.00	200
3	Stapler	120.00	50
4	Marker	25.00	80
5	File Folder	60.00	150

Table 2: Suppliers

Structure:

Column Name	Data Type	Constraints
supplier_id	INT	PRIMARY KEY
supplier_name	VARCHAR(50)	NOT NULL
contact_no	VARCHAR(20)	UNIQUE
product_id	INT	FOREIGN KEY REFERENCES Products(product_id)

Sample Data:

supplier_id	supplier_name	contact_no	product_id
101	StationeryMart	9876543210	1
102	PaperWorld	9876500000	2
103	OfficeSupplies	9876512345	3
104	MarkerHub	9876522222	4
105	FileDepot	9876533333	5

Section A: DDL (Data Definition Language)

1. Create a database called InventoryDB.
2. Create a table Products and table Suppliers with the specified columns and constraints:

Section B: DML (Data Manipulation Language)

4. Insert at least 5 rows into the Products table.
5. Insert at least 5 rows into the Suppliers table.
6. Update the stock quantity of product 'Pen' to 120.
7. Delete a supplier with a specific supplier_id.
8. Write a query to rename 'Notebook' to 'NoteBook A4'

Section C: DQL (SELECT Queries)

9. Display all records from the Products table.
10. Display only product_name and price of all products.
11. List all products that have a stock quantity less than 100.
12. Show all products between 20 and 100 price range.
13. Find all suppliers whose contact number starts with '98765'.
14. Find the average price of products.
15. Display the total number of products in the inventory.
16. Show the maximum and minimum stock quantities.
17. Count how many suppliers supply each product.
18. Show all products where price > 50 AND stock_qty > 100.
19. Show all products where price < 20 OR stock_qty < 80.
20. Display suppliers whose supplier_name contains the word 'Mart'
21. List all suppliers along with the product they supply (use INNER JOIN).
22. Display suppliers whose name starts with 'S'.
23. Find products whose name has exactly 5 characters
24. Find suppliers who supply products costing more than 100.

Experiment 2 : ONLINE BOOKSTORE DB

An online book store wants to implement a BOOKSTORE DB for managing their online transactions by using the following tables.

Authors Table

Column Name	Data Type	Constraints
author_id	INTEGER	PRIMARY KEY
first_name	VARCHAR	NOT NULL
last_name	VARCHAR	NOT NULL
nationality	VARCHAR	NULL allowed

Books Table

Column Name	Data Type	Constraints
book_id	INTEGER	PRIMARY KEY
Title	VARCHAR	NOT NULL
author_id	INTEGER	FOREIGN KEY REFERENCES Authors
publication_year	INTEGER	
Price	DECIMAL	

Customers Table

Column Name	Data Type	Constraints
customer_id	INTEGER	PRIMARY KEY
first_name	VARCHAR	NOT NULL
last_name	VARCHAR	NOT NULL
Email	VARCHAR	UNIQUE, NOT NULL
Address	VARCHAR	NOT NULL

Orders Table

Column Name	Data Type	Constraints
order_id	INTEGER	PRIMARY KEY
customer_id	INTEGER	FOREIGN KEY REFERENCES Customers
book_id	INTEGER	FOREIGN KEY REFERENCES Books
order_date	DATE	NOT NULL
quantity	INTEGER	NOT NULL

SAMPLE DATA SET for BOOKSTORE DB

Authors Table

author_id	first_name	last_name	nationality
1	Jane	Austen	British
2	George	Orwell	British
3	Gabriel	Garcia Marquez	Colombian
4	Toni	Morrison	American
5	Mark	Twain	American
6	Harper	Lee	American
7	Fyodor	Dostoevsky	Russian

Books Table

book_id	Title	author_id	publication_year	price
101	Pride and Prejudice	1	1813	12.99
102	1984	2	1949	9.50
103	One Hundred Years of Solitude	3	1967	15.00
104	Beloved	4	1987	11.25
105	Animal Farm	2	1945	8.75
106	Adventures of Huckleberry Finn	5	1884	10.50
107	To Kill a Mockingbird	6	1960	14.00

Customers Table

customer_id	first_name	last_name	Email	address
201	Alice	Smith	alice.s@example.com	12 Oak St, London
202	Bob	Johnson	bob.j@example.com	45 Pine Ave, Oxford
203	Charlie	Brown	charlie.b@example.com	78 Maple Rd, Bristol
204	Diana	Prince	diana.p@example.com	34 Queen St, York
205	Edward	Norton	edward.n@example.com	22 River Ln, Leeds
206	Fiona	Hall	fiona.h@example.com	56 Lake Dr, Bath
207	Greg	Miller	greg.m@example.com	89 Park Ave, Glasgow

Orders Table

order_id	customer_id	book_id	order_date	Quantity
301	201	101	2025-07-20	1
302	202	102	2025-07-21	2
303	201	105	2025-07-22	1
304	203	103	2025-07-23	1
305	204	106	2025-07-24	1
306	205	107	2025-07-25	3
307	206	104	2025-07-26	2

Section A: DDL (Schema Design & Constraints)

1. Write SQL statements to create all 4 tables (Authors, Books, Customers, Orders) with:
 - o Primary Keys
 - o Foreign Keys
 - o Appropriate data types
 - o NOT NULL constraints where necessary.

2. Alter the Books table to add a constraint that price must be greater than 0.
3. Add a new column phone_number to the Customers table (VARCHAR(15)) and ensure it is unique.
4. Drop the phone_number column from the Customers table.

Section B: DML (Data Manipulation)

5. Insert at least 7 records for each table (use sample dataset above).
6. Update the price of the book titled *Animal Farm* by increasing it by 10%.
7. Delete all orders made before 2025-07-21.
8. Change the nationality of Gabriel Garcia Marquez to “Latino-American”.

Section C: SELECT Queries (Data Querying)

9. List all books published between 1900 and 2000.
10. Find all customers whose email contains “example.com”.
11. Retrieve books whose price is between 10 and 15 and published before 1950.
12. Show authors who are either ‘British’ or ‘American’.
13. Find books that have a price less than 10 or are published after 1980.
14. Display all orders placed after 2025-07-22.
15. List all books written by author with author_id = 2.
16. Find customers whose last name starts with B.
17. Show all books with a price NOT between 9 and 13.
18. Display books whose publication_year is in (1813, 1945, 1987).
19. Find authors whose nationality is NOT ‘British’.
20. List customers whose address contains the word Park.
21. Show all books sorted by price in descending order.
22. List authors in alphabetical order by last_name.
23. Display orders sorted by order_date (latest first).

Use of Date Functions

24. Show all orders placed in July 2025.
25. Show all orders with an estimated delivery date (5 days after order date).
26. Show customers who placed an order on a weekend.
27. Calculate how many days have passed since the last order was placed.

Aggregate Functions (COUNT, SUM, AVG, MIN, MAX)

28. Count the total number of books in the database.
29. Find the average price of all books.
30. Show the highest-priced book.

31. Count how many orders each customer has placed.
32. Calculate the total sales (price × quantity) for each customer.

GROUP BY and HAVING

33. Count how many books are written by each author.
34. Group orders by customer_id and display total quantity ordered.
35. Show customers who have ordered more than 2 books in total (use HAVING).
36. Find the total number of books sold per author (GROUP BY author).

Experiment 3: EMPLOYEE DB

An enterprise wants to automate its employee management process by implementing an Employee Database. The goal is to replace manual record-keeping with a centralized system that stores employee, department, and project details. Use the following table structures and data set to implement Employee DB.

EmployeeDB – Table Structures

1. Departments Table

Column	Type	Constraints
dept_id	INT	PRIMARY KEY
dept_name	VARCHAR	UNIQUE, NOT NULL
location	VARCHAR	NOT NULL

2. Employees Table

Column	Type	Constraints
emp_id	INT	PRIMARY KEY
first_name	VARCHAR	NOT NULL
last_name	VARCHAR	NOT NULL
email	VARCHAR	UNIQUE, NOT NULL
phone	VARCHAR	CHECK (phone LIKE '--____')
hire_date	DATE	NOT NULL
job_title	VARCHAR	NOT NULL
salary	DECIMAL	CHECK (salary > 0)

dept_id	INT	FOREIGN KEY REFERENCES Departments(dept_id)
manager_id	INT	FOREIGN KEY REFERENCES Employees(emp_id) (self-referential)

3. Projects Table

Column	Type	Constraints
project_id	INT	PRIMARY KEY
project_name	VARCHAR	NOT NULL
start_date	DATE	NOT NULL
end_date	DATE	NULL
dept_id	INT	FOREIGN KEY REFERENCES Departments(dept_id)

4. Employee_Project Table (Many-to-Many)

Column	Type	Constraints
emp_id	INT	FOREIGN KEY REFERENCES Employees(emp_id), PRIMARY KEY(emp_id, project_id)
project_id	INT	FOREIGN KEY REFERENCES Projects(project_id)
hours_allocated	INT	CHECK (hours_allocated > 0)

Sample Data Set

Departments Table

dept_id	dept_name	Location
1	HR	New York
2	IT	San Francisco
3	Finance	Chicago
4	Marketing	Boston
5	Operations	Seattle

6	Legal	Washington D.C.
7	Sales	Dallas
8	R&D	Austin
9	Procurement	Denver
10	Customer Care	Miami

2. Employees Table

emp_id	first_name	last_name	Email	phone	hire_date	job_title	salary	dept_id	manager_id
101	Alice	Johnson	alice.j@corp.com	123-456-7890	2020-03-15	HR Manager	75000	1	NULL
102	Bob	Smith	bob.s@corp.com	234-567-8901	2019-05-20	IT Analyst	65000	2	104
103	Charlie	Brown	charlie.b@corp.com	345-678-9012	2021-01-10	Finance Executive	58000	3	106
104	Diana	Prince	diana.p@corp.com	456-789-0123	2018-07-12	IT Manager	90000	2	NULL
105	Ethan	Hunt	ethan.h@corp.com	567-890-1234	2022-02-25	Marketing Lead	62000	4	NULL
106	Fiona	Hall	fiona.h@corp.com	678-901-2345	2017-11-01	Finance Manager	85000	3	NULL
107	Greg	Miles	greg.m@corp.com	789-012-3456	2023-04-15	IT Support	45000	2	104
108	Hannah	White	hannah.w@corp.com	890-123-	2021-	HR	50000	1	101

			orp.com	4567	09-05	Executive			
109	Ian	Scott	ian.s@corp.com	901-234-5678	2020-11-20	Operations Analyst	56000	5	NULL
110	Julia	Adams	julia.a@corp.com	012-345-6789	2019-12-18	Legal Advisor	70000	6	NULL

3. Projects Table

project_id	project_name	start_date	end_date	dept_id
201	Payroll System	2023-01-01	NULL	3
202	Website Upgrade	2023-02-10	NULL	2
203	Recruitment Drive	2023-03-05	NULL	1
204	Ad Campaign	2023-05-20	NULL	4
205	New CRM Tool	2023-04-15	NULL	7
206	Compliance Portal	2023-06-10	NULL	6
207	Inventory System	2023-07-01	NULL	5
208	AI Research	2023-08-05	NULL	8
209	Customer Feedback	2023-09-10	NULL	10
210	Procurement System	2023-10-01	NULL	9

4. Employee_Project Table

emp_id	project_id	hours_allocated
102	202	120
104	202	80
103	201	100

106	201	150
101	203	50
105	204	70
107	202	60
109	207	90
110	206	110
108	203	40

Section A: DDL (Schema Creation & Modification)

1. Write SQL statements to create the above tables with the specified constraints
2. Alter the Employees table to add a column bonus DECIMAL(8,2) with default value 0.
3. Drop the column bonus from Employees.

Section B: DML (Insert, Update, Delete)

4. Insert at least 10 rows into Departments, Employees, Projects, and Employee_Project.(use the above data set)
5. Try inserting an employee with a negative salary (should fail due to CHECK constraint).
6. Update the salary of the employee with emp_id = 103 by 15%.
7. Delete an employee record who has resigned (choose any emp_id).
8. Increase all employees' salaries in the IT department by 5%.
9. Change the department of an employee to "Research".(should fail due to FK constraint)

Section C: DQL (Select Queries)

10. List all employees and their details.
11. Show all employees in the "HR" department.
12. Find employees with salaries between 50,000 and 80,000.
13. Retrieve employees hired after 2020.

14. Show employees who are in either the IT or Finance department.
15. Find employees whose email ends with "@corp.com".
16. List all employees with salary > 60,000 AND located in "New York".
17. Display employees in descending order of salary.
18. Count the number of employees in each department.
19. Show the average salary of employees department-wise.
20. Display departments where the average salary is greater than 70,000.
21. Find the number of employees in each project.
22. Display departments with more than 3 employees.
23. Show the sum of all salaries department-wise.
24. List all distinct department IDs from the Employees table.
25. Show employee names with the year they were hired.
26. Show employees grouped by the year of hire.
27. List employees hired in the last 90 days.
28. List the no of years of experience of all the employees

Section D: Joins

29. List all employees with their department names (INNER JOIN).
30. Display all departments along with employees, including those departments without employees (LEFT JOIN).
31. Show employees and the projects they are working on (JOIN 3 tables: Employees, Employee_Project, Projects).
32. List projects along with total hours allocated by employees.
33. Write a query to find employees who are working on more than one project.
34. Show all projects handled by the 'Finance' department.

Section E: PL/SQL Programming

1. Write a procedure GetEmpInfo that takes emp_id as input and displays name, salary, and department.
2. Write a PL/SQL block that checks if an employee's salary is above 50,000. If yes, print "High Salary" ;Otherwise print "Standard Salary".

3. Write a PL/SQL program to display the top 10 rows in the Emp table based on their job and salary
4. Write a stored procedure GiveBonus that takes department ID and a designation as input, along with a bonus amount, and updates the salary of all employees in that department who have the specified designation by adding the bonus amount to their current salary.
5. Create a trigger to prevent inserting employees with a salary less than 30,000.
6. Create a trigger to avoid any transactions(insert, update, delete) on EMP table on Saturday & Sunday.

KRISHNA UNIVERSITY: MACHILIPATNAM - 521004

B.C.A Honors Computer Applications [Major / Minor]

REVISED CBCS FRAMEWORK WITH EFFECT FROM 2025-2026 ADMITTED BATCH

Model Question Paper

COURSE 4: DATABASE MANAGEMENT SYSTEMS

Semester: II

Time: 3 Hours

Max. Marks: 70

Section – A

Answer any FIVE questions. All questions carry equal marks.

(5 × 4 = 20 M)

1. Define data, information, and database with examples.
2. Write short notes on drawbacks of file-based systems.
3. What are the advantages of the database approach?
4. Define an entity and attribute. Give suitable examples.
5. What is a key? List different types of keys used in a relational model.
6. Write short notes on relational algebra.
7. What are aggregate functions in SQL? Give any four examples.
8. Write short notes on functions in PL/SQL.

Section – B

Answer all questions. Each question carries equal marks.

(5 × 10 = 50 M)

9. a) Explain the components of a Database Management System (DBMS) and its three-schema architecture with a neat diagram.

Or

b) What are data models? Explain various data models.

10. a) Explain the building blocks of an Entity Relationship Diagram (ERD) and the classification of entity sets and attributes with examples.

Or

b) What is an Enhanced ER Model (EER)? Explain generalization, specialization with examples.

11. a) Explain Codd's rules for relational databases. Discuss relational integrity constraints and their importance.

Or

b) Explain Normalization concept? Explain 1NF, 2NF, and 3NF with suitable examples.

12. a) Explain different types of SQL commands with examples — DDL and DML.

Or

b) Discuss SQL operations — selection, projection and joins with examples.

13. a) What is PL/SQL? Explain the structure of a PL/SQL program with example.

Or

b) Write about PL/SQL language elements and data types. Explain the steps to create and execute a PL/SQL program.

KRISHNA UNIVERSITY :: MACHILIPATNAM - 521004

B.C.A Honours Computer Applications[Major / Minor]

REVISED CBCS FRAMEWORK WITH EFFECT FROM 2025-2026 ADMITTED BATCH

BLUEPRINT

COURSE 4: DATABASE MANAGEMENT SYSTEMS

Unit	S.A.Q Section :A (including choice) 4 Marks	E.Q Section: B (including choice) 10 Marks	Marks Allotted
I	2	2	28
II	2	2	28
III	1	2	24
IV	2	2	28
V	1	2	24
Total	8	10	132

SEMESTER-III

COURSE 5: OOPS THROUGH JAVA

Theory

Credits: 3

3 hrs/week

Course Objectives

1. Introduce core OOP principles and contrast procedural and object-oriented paradigms within the Java ecosystem.
2. Equip learners with foundational and advanced Java syntax including variables, control statements, arrays, strings, and classes.
3. Enable the use of inheritance, polymorphism, interfaces, and exception handling to create maintainable and reusable code.
4. Train students in concurrent programming and stream-based I/O operations including file management and serialization.
5. Empower learners to design, build, and manage GUI programs using Swing components, layout managers, and event handling techniques.

Course Outcomes

At the End of the Course, The Students will be able to:

1. Apply OOP principles such as encapsulation, inheritance, and polymorphism in Java applications.
2. Write, compile, and debug Java code using control statements, arrays, classes, and methods effectively.
3. Construct modular code leveraging interfaces, abstract classes, and package hierarchies.
4. Manage thread lifecycles, synchronization, and I/O streams for file handling and console interaction.
5. Design user interfaces using Swing and handle keyboard/mouse input through event-driven programming.

Unit 1. OOPs Concepts and Java Programming:

Introduction to Object-Oriented concepts, procedural and object-oriented programming paradigm

Java programming: An Overview of Java, Java Environment, Data types, Variables, constants, scope and life time of variables, operators, type conversion and casting, Accepting Input from the Keyboard, Reading Input with Java.util.Scanner Class, Displaying Output, Displaying Formatted Output with String.format(), Control Statements

Unit 2. Arrays and OOP Constructs:

Arrays, Command Line Arguments, Strings-String Class Methods

Classes & Objects: Creating Classes, declaring objects, Methods, parameter passing, static fields and methods, Constructors, and 'this' keyword, overloading methods and access Inheritance:

Inheritance hierarchies, super and subclasses, member access rules, 'super' keyword, preventing inheritance: final classes and methods, the object class and its methods; Polymorphism: Dynamic binding, method overriding, abstract classes and methods;

Unit 3. Interfaces, Packages & Exception Handling:

Interfaces VS Abstract classes, defining an interface, implement interfaces, accessing implementations through interface references, extending interface;

Packages: Defining, creating and accessing a package, importing packages.

Exception Handling: Benefits of exception handling, the classification of exceptions, exception hierarchy, checked exceptions and unchecked exceptions, usage of try, catch, throw, throws and finally, rethrowing exceptions, exception specification, built in exceptions, creating own exception sub classes.

Unit 4. Multithreading & Stream based I/O:

Differences between multiple processes and multiple threads, thread states, thread life cycle, creating threads, interrupting threads, thread priorities, synchronizing threads, inter thread communication.

Stream based I/O (java.io) – The Stream classes-Byte streams and Character streams, Reading console Input and Writing Console Output, File class, Reading and writing Files, The Console class, Serialization

Unit 5. GUI Programming with Swing:

Introduction, MVC architecture, components, containers. Understanding Layout Managers - Flow Layout, Border Layout, Grid Layout, Card Layout, GridBag Layout. Event Handling- The Delegation event model- Events, Event sources, Event Listeners, Event classes, Handling mouse and keyboard events, Adapter classes.

Text Books:

1. Java The complete reference, Herbert Schildt, 9th edition, McGraw Hill.
2. Programming in Java, S. Malhotra, S. Chudhary, 2nd edition, Oxford Univ. Press.

Reference Books:

1. Programming with JAVA - A Primer, E Balaguruswamy, 3rd Edition, McGraw Hill
2. Head First Java: A Brain-Friendly Guide , Katty Sierra, Bert Bates, 2nd Edition, O'Reilly

Activities:

Outcome: Apply OOP principles such as encapsulation, inheritance, and polymorphism in Java applications

Activity: Develop a class hierarchy for a zoo management system using inheritance and polymorphism (e.g., Animal → Mammal → Dog). Implement encapsulation through private fields and public getters/setters.

Evaluation Method: Code review and oral explanation focusing on class relationships, method overriding, and encapsulation practices.

Outcome: Write, compile, and debug Java code using control statements, arrays, classes, and methods effectively

Activity: Create a console-based student grade calculator using loops, conditionals, arrays, and modular methods.

Evaluation Method: Practical test with debugging tasks and output validation across multiple input scenarios.

Outcome: Construct modular code leveraging interfaces, abstract classes, and package hierarchies

Activity:

Design a payment processing system with abstract classes for Payment, interfaces for Taxable, and organize classes into packages (e.g., com.billing, com.tax).

Evaluation Method:

Project submission assessed for modularity, interface implementation, abstraction usage, and package structure.

Outcome: Manage thread lifecycles, synchronization, and I/O streams for file handling and console interaction

Activity: Build a multithreaded logger that reads input from the console and writes to a file using synchronized threads and buffered streams.

Evaluation Method: Lab demonstration with thread state tracing and file output verification under concurrent input.

Outcome: Design user interfaces using Swing and handle keyboard/mouse input through event-driven programming

Activity: Create a GUI-based quiz application using Swing components (JFrame, JButton, JTextField) with event listeners for mouse clicks and key presses.

Evaluation Method: Live demo and rubric-based assessment of UI responsiveness, event handling accuracy, and layout design.

SEMESTER-III

COURSE 5: OOPS THROUGH JAVA

Practical

Credits: 1

2 hrs/week

List of Experiments

1. Write a Java program to print Fibonacci series.
2. Write a Java program to calculate multiplication of 2 matrices.
3. Write a Java program for sorting a given list of names in ascending order.
4. Create a class Rectangle. The class has attributes length and width. It should have methods that calculate the perimeter and area of the rectangle. It should have readAttributes() method to read length and width from the user.
5. Write a Java program that implements method overloading.
6. Write a Java program to implement various types of inheritance
 - i. Single
 - ii. Multi-Level
 - iii. Hierarchical
 - iv. Hybrid
7. Write a java program to implement runtime polymorphism.
8. Write a Java program which accepts withdrawal amount from the user and throws an exception In Sufficient Funds when withdrawal amount is more than available amount.
9. Write a Java program to create three threads and that displays good morning, for every one second, hello for every 2 seconds and welcome for every 3 seconds by using extending Thread class.
10. Write a Java program that creates three threads. The first thread displays OOPS, the second thread displays Through and the third thread displays JAVA by using Runnable interface.
11. Write a Java program that displays the number of characters, lines and words in a text file.
12. Implement a Java program for handling mouse events when the mouse entered, exited, clicked, pressed, released, dragged and moved in the client area.
13. Implement a Java program for handling key events when the key board is pressed, released, typed.
14. Write a Java swing program that reads two numbers from two separate text fields and displays the sum of two numbers in the third text field when the button add is pressed.
15. Write a Java program to design student registration form using Swing Controls. The form must have the following fields and button SAVE
Form Fields are: Name, RNO, Mailid, Gender, Branch, Address.

KRISHNA UNIVERSITY: MACHILIPATNAM - 521004
B.C.A Honors Computer Applications [Major / Minor]

REVISED CBCS FRAMEWORK WITH EFFECT FROM 2025-2026 ADMITTED BATCH

COURSE 5: OOPS THROUGH JAVA

Time: 3 hrs.

Max marks: 70

SECTION-A

Answer any **FIVE** out of the following

5 x 4 M= 20M

1. Explain type conversion.
2. What are the operators available in JAVA?
3. Explain command line arguments.
4. What is the use of 'this' keyword?
5. Explain Exception Handling.
6. How many states are there in Thread life cycle?
7. Explain Serialization in I/O stream?
8. Briefly explain MVC architecture?

SECTION-B

Answer ALL Questions

5x10M= 50M

9. Explain OOPs concepts.
OR
10. Explain Control flow statements in JAVA.
11. What is Inheritance? Explain inheritance Hierarchy.
OR
12. Explain Method Overloading and Method Overriding.
13. What is an Interface? How to implement and extend interfaces?
OR
14. How to create and access a package in JAVA?
15. What is a Thread? Explain priorities and synchronization in Threads.
OR
16. How to read and write a file in JAVA? Write an example.
17. Explain Components and Containers Swings?
OR
18. Explain Event Handling in GUI programming.

KRISHNA UNIVERSITY :: MACHILIPATNAM - 521004

B.C.A Honours Computer Applications[Major / Minor]

REVISED CBCS FRAMEWORK WITH EFFECT FROM 2025-2026 ADMITTED BATCH

BLUEPRINT

COURSE 5: OOPS THROUGH JAVA

Unit	S.A.Q Section :A (including choice) 4 Marks	E.Q Section: B (including choice) 10 Marks	Marks Allotted
I	2	2	28
II	2	2	28
III	1	2	24
IV	2	2	28
V	1	2	24
Total	8	10	132

SEMESTER-III

COURSE 6: COMPUTER ORGANIZATION

Theory

Credits: 3

3 hrs/week

Course Objectives

1. To introduce foundational concepts of register transfer language and micro-operations, enabling understanding of basic computer organization.
2. To examine CPU architecture and the control unit's design through hardwired and microprogrammed approaches.
3. To explore various memory organization strategies, including hierarchy, cache mapping, and associative techniques.
4. To understand input-output systems and data transfer mechanisms using I/O interfaces and DMA methods.
5. To analyze arithmetic algorithms and the principles of parallel and pipelined processing.

Course Outcomes

At the End of the Course, The Students will be able to:

1. Interpret register-level operations and perform arithmetic, logic, and shift micro-operations within a basic computer framework.
2. Illustrate CPU functionality, addressing modes, and control unit design with both hardware-based and microprogramming techniques.
3. Categorize types of memory and explain memory hierarchy, access methods, and mapping techniques.
4. Describe I/O organization, including asynchronous transfer modes, DMA, and interrupt-driven processing.
5. Apply arithmetic algorithms and demonstrate understanding of parallel and pipelined processing models.

Unit 1. Register Transfer Language and Micro Operations:

Introduction- Functional units, computer registers, register transfer language, register transfer, bus and memory transfers, arithmetic, logic and shift micro-operations, arithmetic logic shift unit. Basic Computer Organization and Design: Instruction codes, instruction cycle. Register reference instructions, Memory – reference instructions, input – output and interrupt.

Unit 2. CPU and Micro Programmed Control:

Central Processing unit: Introduction, instruction formats, addressing modes. Control memory, address sequencing, design of control unit - hard wired control, micro programmed control.

Unit 3. Memory Organization:

Memory hierarchy, main memory, auxiliary memory, associative memory, cache Memory and mappings.

Unit 4. Input-Output Organization:

Peripheral Devices, input-output interface, asynchronous data transfer, modes of transfer-programmed I/O, priority interrupt, direct memory access, Input – Output Processor (IOP).

Unit 5. Computer Arithmetic:

Data representation- fixed point, floating point, addition and subtraction, multiplication and division algorithms.

Text Books:

1. Computer Systems Architecture, M. Moris Mano, 3rd edition, Pearson/ PHI
2. Computer Organization and Architecture, William Stallings, 8th edition, Pearson/PHI

Reference Books:

1. Computer Organisation and Architecture, V. Rajaraman, T. Radha Krishnan, PHI
2. Computer Organization and Architecture Hamacher, Vranesic, Zaky, 5th edition, McGraw Hill

Activities:

Outcome: Interpret register-level operations and perform arithmetic, logic, and shift micro-operations within a basic computer framework.

Activity: Use a simulation tool (e.g., Logisim or Digital Works) to design a simple 4-bit ALU that performs:

- Addition, subtraction
- AND, OR
- Logical and arithmetic shifts

Evaluation Method: Demonstration-based assessment where students explain each operation using test inputs. Include a short worksheet with expected vs. actual outputs.

Outcome: Illustrate CPU functionality, addressing modes, and control unit design with both hardware-based and microprogramming techniques.

Activity: Create a flowchart or block diagram showing:

- CPU components (ALU, registers, control unit)
- Addressing modes (immediate, direct, indirect, etc.)
- Control unit types (hardwired vs. microprogrammed)

Evaluation**Method:**

Oral presentation or peer-reviewed poster session. Use a rubric to assess clarity, completeness, and correct identification of modes and control logic on a 10-point scale.

Outcome: Categorize types of memory and explain memory hierarchy, access methods, and mapping techniques.

Activity: Build a memory hierarchy chart using coloured cards or digital tools. Include:

- Registers, cache, RAM, secondary storage
- Access methods (direct, associative, etc.)
- Mapping techniques (direct, associative, set-associative)

Evaluation Method: Quiz with matching and short-answer questions. Include a scenario-based question where students choose the best memory type for a given task.

Outcome: Describe I/O organization, including asynchronous transfer modes, DMA, and interrupt-driven processing.

Activity: Role-play simulation: Assign students' roles (CPU, DMA controller, I/O device) and simulate data transfer using cards or tokens to represent data and control signals.

Evaluation Method: Reflective worksheet where students describe the flow of control and data in each mode. Include a diagram labelling key signals and steps.

Outcome: Apply arithmetic algorithms and demonstrate floating point arithmetic operations.

Activity: Use a spreadsheet or visual tool to simulate floating point arithmetic operations.

Evaluation Method: Submission of simulation results with explanation. Evaluate the report on a 10-point scale.

SEMESTER-III

COURSE 6: COMPUTER ORGANIZATION

Practical

Credits: 1

2 hrs/week

List of Experiments:

1. Simulate Register Transfer Using Logisim
2. Build a microprogrammed control unit using Logisim or VHDL (GHDL).
3. Simple Register Transfers and Arithmetic Operations using EMU8086 Simulator
4. Simulate a Simple Instruction Cycle in Emu8086
5. Demonstrate Addressing Modes (Immediate, Register, Direct, Indexed) in Emu8086
6. Implement Subroutines Using CALL and RET in Emu8086
7. Simulate DMA Transfer in Ripes or Logisim
8. Simulate Booth's Algorithm for multiplication and restoring division in Python/C or Logisim
9. Write assembly language code for $A+B*(C-D)$ using various instruction formats in any open-source assembler.
10. Write assembly language code for $A+B*C$ using various addressing modes in any open-source assembler.

KRISHNA UNIVERSITY:MACHILIPATNAM - 521004
B.C.A Honors Computer Applications [Major / Minor]

REVISED CBCS FRAMEWORK WITH EFFECT FROM 2025-2026 ADMITTED BATCH

COURSE 6: COMPUTER ORGANIZATION

Time: 3 hrs.

Max marks: 70

SECTION-A

Answer any **FIVE** out of the following

5 x 4 M= 20M

1. What is Micro operation? Give an example.
2. Explain bus transfer?
3. Explain design of control unit.
4. What is address sequencing? Explain
5. Explain Memory.
6. Explain I/O Interface.
7. Explain asynchronous data transfer.
8. Explain Fixed and Floating-point data representations.

SECTION-B

Answer ALL Questions

5x10M= 50M

9. Explain different functional units of a digital computer with neat sketch.
OR
10. Explain arithmetic micro-operations with example.
11. Explain instruction formats.
OR
12. Explain addressing modes.
13. Draw a neat block diagram of memory hierarchy in a computer system?
OR
14. Explain the mapping techniques used for cache mapping?
15. Write a short note on Peripheral devices.
OR
16. Explain different modes of transfer.
17. Explain Booth's multiplication algorithm?
OR
18. Explain Restoring division algorithm.

KRISHNA UNIVERSITY :: MACHILIPATNAM - 521004

B.C.A Honours Computer Applications[Major / Minor]

REVISED CBCS FRAMEWORK WITH EFFECT FROM 2025-2026 ADMITTED BATCH

BLUEPRINT

COURSE 6: COMPUTER ORGANIZATION

Unit	S.A.Q Section :A (including choice) 4 Marks	E.Q Section: B (including choice) 10 Marks	Marks Allotted
I	2	2	28
II	2	2	28
III	1	2	24
IV	2	2	28
V	1	2	24
Total	8	10	132

SEMESTER-III

COURSE 7: PROBABILITY AND STATISTICS

Theory

Credits: 4

4 hrs/week

Course Objectives:

1. To introduce the foundational concepts of statistics including data types, graphical representation, and basic statistical measures.
2. To develop the ability to analyze relationships between variables using correlation, regression, and probability theory.
3. To enable students to apply probability distributions and hypothesis testing for decision-making and statistical inference.

Course Outcomes (COs):

After successful completion of this course, students will be able to:

1. Organize and summarize statistical data using frequency tables, charts, and descriptive measures.
2. Calculate and interpret measures of central tendency and dispersion for given data.
3. Apply correlation and linear regression techniques to examine relationships between variables.
4. Solve real-life problems using probability rules and standard probability distributions.
5. Conduct hypothesis testing using z-test, t-test, and chi-square test for statistical inference.

Unit 1. Basic concepts of Statistics:

Qualitative and quantitative data, classification of data, construction of frequency distribution, diagrammatic representation of data.

Measures of Central Tendency: Arithmetic mean, median and mode-their properties

Measures of Dispersion: Range, mean deviation, quartile deviation, variance and standard deviation.

Unit 2. Correlation & Regression:

Correlation: Definition, scatter diagram, types of correlation, measures—Karl Pearson's correlation coefficient and Spearman's rank correlation coefficient.

Regression: Linear regression-fitting by least square method and interpretation.

Unit 3. Probability & Random Variables:

Concepts of probability: Experiment and sample space, events and operations with events, probability of an event, basic probability rules, applications of probability rules, conditional probability.

Random Variables: Discrete and continuous random variable, probability distribution of a random variable, probability mass function, probability density function, expectation and variance of a random variable.

Unit 4. Probability Distributions:

Standard Probability Distributions: Binomial probability distribution, Poisson probability distribution, Normal probability distribution.

Sampling Distribution: Concept of Population and Sample, parameter and statistic, sampling distribution of sample mean and sample proportion.

Unit 5. Statistical Inference Techniques:

Statistical Inference: Estimation and Hypothesis Testing (only concept).

Hypothesis Testing for a Single Population: Concept of a hypothesis testing, tests involving a population mean and population proportion (z test and t test). **Chi square test for independence of attributes and goodness of fit.**

Text Books

1. Manish Sharma, Amit Gupta, The Practice of Business Statistics, Khanna Book Publishing Company, 2010 (AICTE Recommended Textbook)
2. Das N. G., Statistical Methods, Combined Edition, Tata McGraw Hill, 2010.
3. Ross Sheldon M., Introduction to Probability and Statistics for Engineers and Scientists, 6th Edition, Elsevier, 2021.
4. Miller Irwin and Miller Marylees, Mathematical Statistics with Applications, Seventh Edition, Pearson Education, 2005

Reference Books

5. Pal Nabendu and Sarkar Sahadeb, Statistics: Concepts and Applications, Second Edition, PHI, 2013
6. Montgomery Douglas and Runger George C., Applied Statistics and Probability for Engineers, Wiley, 2016.
7. Reena Garg, Engineering Mathematics, Khanna Publishing House, 2024.

Web Resources

1. <https://nptel.ac.in/courses/111106112>
2. <https://nptel.ac.in/courses/111105041>

Outcome: Organize and summarize statistical data using frequency tables, charts, and descriptive measures.

Activity: Students will collect real or simulated data (such as student marks or preferences), organize it into a frequency table, and represent it visually using bar charts, pie charts, or histograms.

Evaluation Method: The teacher will evaluate the activity based on accuracy of data classification, correctness of charts, and clarity of presentation.

Outcome: Calculate and interpret measures of central tendency and dispersion for given data.

Classroom Activity: Students will be given a worksheet with datasets in class and asked to compute the mean, median, mode, range, variance, and standard deviation. *Evaluation Method:* The worksheet will be reviewed for correctness of formulas used, accuracy of calculations, and proper interpretation of the results.

Outcome: Apply correlation and linear regression techniques to examine relationships between variables.

Classroom Activity: Working in pairs, students will gather small bivariate datasets (e.g., height and weight of classmates), plot scatter diagrams, calculate the correlation coefficient, and derive the regression equation.

Evaluation Method: The teacher will assess the submission based on accuracy of plotting, correct use of formulas, and the interpretation of results. Peer collaboration will also be observed.

Outcome: Solve real-life problems using probability rules and standard probability distributions.

Classroom Activity: Students will conduct simple random experiments in class—like rolling dice, flipping coins, or drawing colored balls from a bag—to understand probability. They will then apply theoretical probability concepts to solve related questions. *Evaluation Method:* Students will be assessed based on their understanding and application of probability rules, logical reasoning, and active participation in the group task.

Outcome: Conduct hypothesis testing using z-test, t-test, and chi-square test for statistical inference.

Classroom Activity: Students will be given a case scenario or dataset (e.g., a small sample of test scores) and asked to choose the correct hypothesis test, carry it out, and interpret the result. This will be followed by a short classroom discussion or group explanation. *Evaluation Method:* Students will be evaluated based on the accuracy of hypothesis formulation, selection of the appropriate test, computation steps, and clarity in explaining their conclusions.

KRISHNA UNIVERSITY:MACHILIPATNAM - 521004

B.C.A Honors Computer Applications [Major / Minor]

REVISED CBCS FRAMEWORK WITH EFFECT FROM 2025-2026 ADMITTED BATCH

COURSE 7: PROBABILITY AND STATISTICS

Time: 3 hrs.

Max marks: 70

SECTION-A

Answer any **FIVE** out of the following

5 x 4 M= 20M

1. Define qualitative and quantitative data with examples.
2. Explain classification of data and types.
3. Define Arithmetic Mean and mention its properties.
4. What is scatter diagram?
5. Define correlation and types of correlation.
6. Explain sample space and events.
7. Define discrete and continuous random variables.
8. What is Binomial distribution?

SECTION – B

Answer **ALL** questions

5 × 10M = 50 M

9. Construct a frequency distribution table and explain diagrammatic representation of data.

OR

10. Explain Measures of Central Tendency – Mean, Median and Mode with properties.
11. Explain Karl Pearson's coefficient of correlation with example.

OR

12. Explain Spearman's Rank Correlation with suitable example.
13. Define probability and explain basic probability rules with applications.

OR

14. Explain probability distribution, PMF, PDF, expectation and variance.
15. Explain Normal distribution and its properties.

OR

16. Explain sampling distribution of sample mean and sample proportion.
17. Explain hypothesis testing for population mean using z-test.

OR

18. Explain Chi-square test for independence and goodness of fit.

KRISHNA UNIVERSITY :: MACHILIPATNAM - 521004

B.C.A Honours Computer Applications[Major / Minor]

REVISED CBCS FRAMEWORK WITH EFFECT FROM 2025-2026 ADMITTED BATCH

BLUEPRINT

COURSE 7: PROBABILITY AND STATISTICS

Unit	S.A.Q Section :A (including choice) 4 Marks	E.Q Section: B (including choice) 10 Marks	Marks Allotted
I	2	2	28
II	2	2	28
III	1	2	24
IV	2	2	28
V	1	2	24
Total	8	10	132

SEMESTER-IV

COURSE 8: OPERATING SYSTEMS

Theory

Credits: 3

3 hrs/week

Course Objectives:

1. **Understand the evolution and core functions** of operating systems, including resource management and system types.
2. **Analyze process and thread management**, focusing on system calls, kernel modes, scheduling algorithms, and threading models.
3. **Evaluate process synchronization and deadlock handling**, using classical problems and inter-process communication methods.
4. **Apply memory management techniques**, including paging, segmentation, and virtual memory implementation.
5. **Examine file, I/O, and device management strategies**, along with basic OS-level security features.

Course Outcomes:

At the End of the Course, The Students will be able to:

1. Explain the foundational principles and evolution of operating systems, including resource abstraction and core management functions.
2. Classify and compare different types of operating systems, such as multiprogramming, batch, time-sharing, real-time, and personal device-based systems.
3. Analyze process and thread management techniques, including processor modes, system calls, kernel functions, and scheduling algorithms.
4. Evaluate process synchronization and deadlock handling approaches, applying classical concurrency solutions and inter-process communication methods.
5. Apply memory, file, and I/O management strategies, incorporating allocation techniques, virtual memory models, disk scheduling, and OS-level security features.

Unit 1. Operating System Fundamentals:

Operating System Definition, History and Evolution of OS, Basic OS functions, Types of Operating Systems– Multiprogramming Systems, Batch Systems, Time Sharing Systems; Operating Systems for Personal Computers, Workstations and Hand-held Devices, Process Control & Real time Systems.

Unit 2. Process & Thread:

Processor and User Modes, Kernels, System Calls and System Programs, System View of the Process and Resources, Process Abstraction, Process Hierarchy, Threads, Threading Issues, Thread Libraries; Process Scheduling- Non-Preemptive and Preemptive Scheduling Algorithms.

Unit 3. Process Management:

Concurrent and Dependent Processes, Critical Section, Semaphores, Methods for Inter process Communication; Process Synchronization, Classical Process Synchronization Problems: Producer-Consumer, Reader-Writer.

Deadlock, Deadlock Characterization, Necessary and Sufficient Conditions for Deadlock, Deadlock Handling Approaches: Deadlock Prevention, Deadlock Avoidance and Deadlock Detection and Recovery.

Unit 4. Memory Management:

Physical and Virtual Address Space; Memory Allocation Strategies–Fixed and -Variable Partitions, Paging, Segmentation, Virtual Memory.

Unit 5. File and I/O Management:

Directory Structure, File Operations, File Allocation Methods, Device Management, Pipes, Buffer, Shared Memory, Disk Scheduling algorithms.

Text Books:

1. Operating System Principles, Abraham Silberschatz, Peter Baer Galvin and GregGagne, 7th Edition, Wiley India Edition.
2. Operating Systems: Internals and Design Principles, Stallings , Pearson edition

Reference Books:

1. Operating Systems Design and Implementation, Andrew S. Tanenbaum, 3rd Edition, Pearson
2. A Text Book of Operating Systems, Singh, Kaur and Gupta, Khanna Publishers

Activities:

Outcome: Explain Foundational Principles and Evolution of Operating Systems

Activity: Timeline Creation & Concept Mapping - Students will collaboratively build a visual timeline showing the evolution of operating systems from early batch systems to modern distributed and mobile OS. They will also create a concept map illustrating core management functions (e.g., process, memory, file, device management).

Evaluation Method: Rubric-based assessment of timeline and concept map:

- Historical accuracy
- Inclusion of core OS functions
- Presentation and teamwork

Outcome: Classify and Compare Types of Operating Systems

Activity: Comparative Analysis Table & Case Study Discussion - Students will research and fill out a comparison table for OS types (multiprogramming, batch, time-sharing, real-time, personal device-based), focusing on architecture, scheduling, responsiveness, and use cases. Followed by a group discussion using real-world examples (e.g., RTOS in pacemakers vs. Android in smartphones).

Evaluation Method: Students will be evaluated on a 10-point scale based on

- Individual submission of comparison table
- Group participation score in discussion

Outcome: Analyze Process and Thread Management Techniques

Activity: Simulation & Code Walkthrough - Students will simulate process scheduling using tools or pseudocode (e.g., Round Robin, Priority Scheduling). They will also analyze thread creation and system calls using a simple multithreaded program in Java.

Evaluation Method: Students will be assessed on a 10-point scale based on

- Scheduling algorithm simulation results
- Explanation of processor modes and system calls
- Code annotations for kernel functions

Outcome: Evaluate Process Synchronization and Deadlock Handling

Activity: Role-play & Coding Challenge - Students will role-play classical problems (e.g., Dining Philosophers, Producer-Consumer) to understand synchronization. Then, they'll implement semaphore or monitor-based solutions in code and simulate deadlock detection or avoidance.

Evaluation Method: Students will be evaluated for 10 marks based on

- Code review for correctness and use of synchronization primitives
- Peer evaluation of role-play clarity and engagement
- Written test with deadlock scenarios and solution strategies

Outcome: Apply Memory, File, and I/O Management Strategies

Activity: Interactive Lab & Design Task - Students will use OS simulators to experiment with memory allocation (paging, segmentation), disk scheduling (FCFS, SSTF, SCAN), and file system operations.

Evaluation Method: Students will be evaluated for 10 points based on

- Correct use of allocation techniques
- Disk scheduling output analysis
- Virtual memory behaviour observation

SEMESTER-IV

COURSE 8: OPERATING SYSTEMS

Practical

Credits: 1

2 hrs/week

List of Experiments:

1. Illustrate the LINUX commands
 - a) pwd
 - b) mkdir
 - c) rmdir
 - d) grep
 - e) chmod
 - f) ls
 - g) rm
 - h) cp
2. Write a program to calculate average waiting time and turn around time of each process using the following CPU Scheduling algorithm for the given process schedules.
 - a) FCFS
 - b) SJF
 - c) Priority
 - d) Round Robin
3. Simulate MVT and MFT memory management technique
4. Write a program for Bankers Algorithm for Dead Lock Avoidance
5. Implement Bankers Algorithm Dead Lock Prevention.
6. Write a program to simulate Producer-Consumer problem.
7. Simulate all Page replacement algorithms.
 - a) FIFO
 - b) LRU
 - c) LFU
 - d) Optimal
8. Simulate Paging Techniques of memory management
9. Simulate the following disk scheduling algorithms
 - a) FCFS
 - b) SSTF
 - c) SCAN
 - d) CSCAN

KRISHNA UNIVERSITY:MACHILIPATNAM - 521004

B.C.A Honors Computer Applications [Major / Minor]

REVISED CBCS FRAMEWORK WITH EFFECT FROM 2025-2026 ADMITTED BATCH

COURSE 8: OPERATING SYSTEMS

Time: 3 hrs.

Max marks: 70

SECTION-A

Answer any **FIVE** out of the following

5 x 4 M= 20M

1. Define OS. Explain the evolution of Operating Systems.
2. Differentiate Batch systems and Time-sharing systems.
3. Distinguish between Processor mode and User mode.
4. What are Thread Libraries? Explain briefly.
5. Define Deadlock and list its necessary conditions.
6. Define Physical and Virtual Address Space.
7. What is Critical Section?
8. Discuss Directory structure .

SECTION – B

Answer **ALL** questions

5 × 10M = 50 M

9. Explain Operating System functions and types of Operating Systems.
OR
10. Discuss Operating Systems for Personal Computers, Workstations and Hand-held devices with process control features.
11. Explain System Calls, Kernels and System Programs with system view of processes and resources.
OR
12. Explain Pre-emptive and Non-Pre-emptive Scheduling Algorithms with suitable examples.
13. Explain Process Synchronization mechanisms and Semaphores with Inter Process Communication methods.
OR
14. Explain Classical Synchronization Problems – Producer-Consumer and Reader-Writer problems.
15. Explain Memory Allocation strategies (Fixed and Variable Partitions).
OR
16. Explain Paging and Segmentation.
17. Explain File Operations and File Allocation Methods
OR
18. Explain Disk Scheduling algorithms in detail.

KRISHNA UNIVERSITY:: MACHILIPATNAM - 521004

B.C.A Honours Computer Applications [Major / Minor]

REVISED CBCS FRAMEWORK WITH EFFECT FROM 2025-2026 ADMITTED BATCH

BLUEPRINT

COURSE 8: OPERATING SYSTEMS

Unit	S.A.Q Section: A (including choice) 4 Marks	E.Q Section: B (including choice) 10 Marks	Marks Allotted
I	2	2	28
II	2	2	28
III	1	2	24
IV	2	2	28
V	1	2	24
Total	8	10	132

SEMESTER-IV

COURSE 9: PYTHON PROGRAMMING

Theory

Credits: 3

3 hrs/week

Course Objectives

1. **Introduce the foundational concepts** of Python programming including its syntax, IDEs, and control structures.
2. **Develop proficiency in modular programming** using functions, lambda expressions, recursion, and Python's built-in modules and packages.
3. **Explore core data structures** like strings, lists, tuples, and dictionaries for effective data manipulation.
4. **Teach exception handling mechanisms** and the use of regular expressions for pattern matching and text processing.
5. **Enable students to interact with files and databases** using Python to build real-world applications involving persistent storage and data retrieval.

Course Outcomes

At the end of the Course, the students will be able to

1. **Write and execute structured Python programs** using variables, expressions, and flow control statements.
2. **Implement modular code** leveraging functions, argument types, recursion, and reusable libraries.
3. **Manipulate and organize data efficiently** using Python's string operations and complex data structures.
4. **Handle runtime errors and apply regular expressions** for robust and flexible program behaviour.
5. **Perform file operations and connect to databases** through Python scripts to store, retrieve, and manage data effectively.

Unit 1. Basics of Python Programming:

Features of python, history of python, Python IDEs, Writing and Executing Python Program, literal constants, variables and identifiers, Data types, input operation- comments, Reserved words, Indentation, Operators and Expressions: Expressions in python, Operations in Strings, Other Data types, Type conversion, Decision control Statements, Iterative Statements, Nested loops, break, Continue, Pass Statements, else statement used with loops.

Unit 2. Strings and Collections:

Strings: Built-in String Methods and Functions

Lists: Access values in List, Updating values in Lists, Nested lists, Basic list operations, List Methods.

Tuple: Creating, Accessing, Updating and Deleting Elements in a tuple, Nested tuples.

Dictionaries: Creating a dictionary, Accessing values, Modifying an Entry, Deleting items, Built-in Dictionary Functions and Methods

Unit 3. Functions and Modules:

Function Definition, Function Call, Variable Scope and lifetime, The return statement, Required Arguments, Keyword Arguments, Default Arguments and Variable Length Arguments, Lambda Functions, Recursive Functions.

Importing Libraries, Modules, Packages in python, Standard library modules- Globals(), Locals(), and Reload(), Function Redefinition.

Unit 4. Exception Handling:

Errors and Exceptions, Handling Exceptions, Multiple Except blocks, Multiple Exceptions in a single block, Except Block without Exception, The else clause, Built-in and user-defined Exceptions, The finally block, Re-raising Exception, Assertions in python

Unit 5. File Handling & Database Connectivity:

Types of files in Python, Opening and Closing files, Reading and Writing files: write() and writelines() methods- append() method, read() and readlines() methods, Splitting words, File Positions.

Database connectivity using Python, Executing SQL commands using python, Assimilating SQL command results using python.

Textbooks:

1. Python Programming using Problem Solving Approach Reema Thareja Oxford University Press 2020
2. Exploring Python, Budd T A, McGraw-Hill Education, 1st Edition, 2011.

Reference Book:

1. Python: The Complete Reference, Martin C. Brown, Mc Graw Hill, 2018
2. Fundamentals of Python, Kenneth A. Lambert. (2019), First Programs, 2nd Edition, CENGAGE Publication.

Activities:

Outcome: Write and execute structured Python programs using variables, expressions, and flow control statements.

Activity: Create a calculator program that uses variables, arithmetic expressions, and flow control (if-else) to perform basic operations (add, subtract, multiply, divide) based on user input.

Evaluation Method: Code walkthrough and output validation. Use a checklist to assess on a 10-point scale to check the:

- Correct use of variables and expressions
- Proper flow control logic
- Accurate results for different inputs

Outcome: Implement modular code leveraging functions, argument types, recursion, and reusable libraries.

Activity: Build a factorial calculator using both iterative and recursive functions. Include parameterized functions and import the math library for comparison.

Evaluation Method: Code review and oral explanation. Assess on 10-point scale based on:

- Function structure and argument usage
- Recursion logic
- Use of reusable libraries

Outcome: Manipulate and organize data efficiently using Python's string operations and complex data structures.

Activity: Develop a contact manager that stores names and phone numbers using dictionaries and lists. Include string formatting and search functionality.

Evaluation Method: Practical demo with test cases. Evaluate based on:

- Use of string methods (split, join, format)
- Data structure selection and manipulation
- Search and retrieval accuracy

Outcome: Handle runtime errors and apply regular expressions for robust and flexible program behaviour.

Activity: Create a form validator that checks email and phone number formats using regular expressions. Include try-except blocks to handle invalid inputs.

Evaluation Method: Scenario-based testing. Assess based on:

- Regex accuracy for pattern matching
- Robust error handling
- Program stability with edge cases

Outcome: Perform file operations and connect to databases through Python scripts to store, retrieve, and manage data effectively.

Activity: Build a student grade logger that reads from a CSV file, stores data in a SQLite database, and allows querying by student name.

Evaluation Method: Lab test with sample data. Evaluate on a 10-point scale:

- File read/write operations
- Database connection and query execution
- Data integrity and retrieval accuracy

SEMESTER-IV

COURSE 9: PYTHON PROGRAMMING

Practical

Credits: 1

2 hrs/week

List of Experiments

1. Display a welcome message using print().
2. Declare and use identifiers belonging to strings, integers, floats, and booleans.
3. Accept user input (name, age, height, student status) and display each value with its type using type().
4. Perform operations like .upper(), .find(), .replace() on strings.
5. Write a program to reverse the string, count vowels and words.
6. Write a program for slicing, sorting, and list comprehension.
7. Program to apply list methods: append(), extend(), insert(), remove(), pop(), sort().
8. Create tuples to store student (name, age, course) data and perform
 - a. Accessing elements using indexing and slicing.
 - b. Demonstrate immutability by attempting to modify a tuple.
 - c. Create and navigate nested tuples.
9. Create a dictionary with student roll number as keys and names/marks as values.
 - a. Accessing, adding, updating, and deleting key-value pairs.
 - b. Iterating through keys, values, and items.
10. Write a program to demonstrate variable length arguments.
11. Write a program to illustrate lambda and recursive functions.
12. Write a program to demonstrate globals(), locals(), and reload() functions.
13. Demonstrate exception handling and assertions in Python.
14. Write a Python program to copy the contents of one file into another in reverse order.
15. Working with Datasets using Numpy and Pandas
16. Write a program to connect to the database and retrieve the required information using SQL commands.

KRISHNA UNIVERSITY:MACHILIPATNAM - 521004

B.C.A Honors Computer Applications [Major / Minor]

REVISED CBCS FRAMEWORK WITH EFFECT FROM 2025-2026 ADMITTED BATCH

COURSE 9: PYTHON PROGRAMMING

Time: 3 hrs.

Max marks: 70

SECTION-A

Answer any **FIVE** out of the following

5 x 4 M= 20M

1. List the important features of Python and explain indentation rules.
2. Explain break and continue with suitable syntax.
3. Write a note on built-in string methods with examples.
4. Differentiate Lists and Tuples with examples.
5. Define lambda functions and give an example.
6. What are Python modules and packages?
7. Explain user-defined exceptions.
8. Define file opening modes and explain file positioning.

SECTION – B

Answer **ALL** questions

5 × 10 = 50 M

9. Explain Python data types, operators and expressions.

OR

10.Explain iterative statements, nested loops and the use of break, continue, pass and else with loops using suitable programs.

11.Explain Lists and their operations including nested lists and list methods with examples.

OR

12.Explain Dictionaries in Python – creation, accessing values, modifying entries, deleting items and built-in methods.

13.Explain function definition, function call, variable scope and lifetime with different types of arguments.

OR

14.Explain importing libraries, modules and packages in Python. Discuss standard library utilities like globals(), locals(), reload() and function redefinition.

15. Explain exception handling in Python with multiple except blocks, else clause, finally block .

OR

16. Explain built-in and user-defined exceptions, re-raising exceptions and handling multiple exceptions in a single block with examples.

17. Explain reading and writing files in Python using write(), writelines(), append(), read() and readlines() with examples.

OR

18. Explain database connectivity using Python and illustrate executing SQL commands and retrieving results through Python programs.

KRISHNA UNIVERSITY:: MACHILIPATNAM - 521004

B.C.A Honours Computer Applications [Major / Minor]

REVISED CBCS FRAMEWORK WITH EFFECT FROM 2025-2026 ADMITTED BATCH

BLUEPRINT

COURSE 9: PYTHON PROGRAMMING

Unit	S.A.Q Section: A (including choice) 4 Marks	E.Q Section: B (including choice) 10 Marks	Marks Allotted
I	2	2	28
II	2	2	28
III	2	2	28
IV	1	2	24
V	1	2	24
Total	8	10	132

SEMESTER-IV

COURSE 10: SOFTWARE ENGINEERING

Theory

Credits: 3

3 hrs/week

Course Objectives :

1. Understand the fundamental principles of software engineering, including software development life cycle models and their practical applications.
2. Analyze and document software requirements through feasibility studies and specification techniques.
3. Apply software design principles and development standards for building reliable and maintainable systems.
4. Evaluate software testing methodologies, including design and execution of test cases for various testing levels.
5. Examine software maintenance strategies, types, and metrics to sustain software performance over time.

Course Outcomes

At the End of the Course, The Students will be able to:

1. Compare and contrast software development life cycle models such as Waterfall, Spiral, and Agile, and explain their appropriate use cases.
2. Conduct requirement analysis and distinguish between functional and non-functional requirements to develop a Software Requirements Specification (SRS) document.
3. Utilize design principles like modularity, cohesion, and coupling; implement Data Flow Diagrams (DFDs), structure charts, and follow coding standards and version control practices.
4. Design and perform different types of software tests—white-box, black-box, unit, integration, system—and differentiate between manual and automated testing approaches.
5. Categorize software maintenance types and propose strategies based on software maintenance metrics for effective long-term software sustainability.

Unit 1: Software Engineering Foundations and Requirements Engineering:

Definition of Software engineering, Software life cycle models: Waterfall, prototyping, Evolutionary, Spiral and Agile models. Comparison among development life cycles.

Unit 2: Requirement Analysis and Specification

Feasibility study, Requirements gathering, Functional and Non-functional requirements, Requirements analysis and specification, design of software requirement specification (SRS).

Unit 3: Software Design Principles and Development Practices

Introduction to software design, modularity, cohesion, coupling and layering, functional design, and solution design, use of DFD and Structure chart in software design, UML in software design, user interface design. Software Development Basics, Coding standards, Version Control and Code review techniques.

Unit 4: Software Testing

Fundamentals of testing (verification and validation), White-box, and black-box testing, unit testing, integration testing, system testing, acceptance testing (alpha testing and beta testing), test scenarios and test case design, automation testing and manual testing.

Unit 5: Software Maintenance

Introduction to software maintenance, corrective maintenance, perfective maintenance, adaptive maintenance, preventive maintenance, challenges in software maintenance, metrics related to software maintenance.

Textbooks:

1. Software Engineering: A Practitioner's Approach, Roger Pressman, McGraw Hill, 6th Edition
2. Software Engineering, Sommerville, Addison Wesley, 7th edition.

Reference Books:

1. Fundamentals of Software Engineering, Mall Rajib, PHI, Fifth Edition,
2. Fundamentals of Software Engineering, Hitesh, BPB Publications

Activities:

Outcome: Compare and contrast software development life cycle models such as Waterfall, Spiral, and Agile, and explain their appropriate use cases.

Activity: Create a comparison chart in groups showing key features, pros/cons, and use cases of Waterfall, Spiral, and Agile models. Include a real-world example for each.

Evaluation Method: Use a rubric to assess on a 10-point scale to check:

- Accuracy of model descriptions
- Clarity of comparison
- Relevance of examples
- Presentation skills (if shared in class)

Outcome: Conduct requirement analysis and distinguish between functional and non-functional requirements to develop a Software Requirements Specification (SRS) document.

Activity: Analyze a simple system (e.g., online library or food ordering app). Identify 5 functional and 3 non-functional requirements. Draft a mini-SRS document using a template.

Evaluation Method: Checklist-based review on a 10-point scale:

- Correct classification of requirements
- Completeness of SRS sections
- Clarity and formatting
- Use of standard terminology

Outcome: Utilize design principles like modularity, cohesion, and coupling; implement Data Flow Diagrams (DFDs), structure charts, and follow coding standards and version control practices.

Activity: Design a basic login system using DFD (Level 0 and Level 1) and a structure chart. Highlight modules and discuss cohesion and coupling in pairs.

Evaluation Method: Peer review and instructor feedback on a 10-point scale:

- Correct use of DFD symbols
- Logical flow and modularity
- Explanation of cohesion/coupling
- Neatness and labelling

Outcome: Design and perform different types of software tests—white-box, black-box, unit, integration, system—and differentiate between manual and automated testing approaches.

Activity: Write simple test cases for a calculator app (e.g., addition, division by zero). Perform manual unit testing and simulate black-box and white-box testing.

Evaluation Method: Observation and worksheet to assess on a 10-point scale:

- Correct identification of test types
- Clear test case steps and expected output
- Execution and result recording
- Understanding of manual vs automated testing

Outcome: Categorize software maintenance types and propose strategies based on software maintenance metrics for effective long-term software sustainability.

Activity: Categorize sample maintenance tasks (e.g., fixing a bug, adding a feature) into corrective, adaptive, perfective, or preventive. Discuss sustainability strategies in small groups.

Evaluation Method: Group activity score:

- Correct classification of maintenance types
- Participation in discussion
- Suggestions for sustainability (e.g., documentation, modular design)
- Use of basic metrics (e.g., number of bugs fixed)

SEMESTER-IV

COURSE 10: SOFTWARE ENGINEERING

Practical

Credits: 1

2 hrs/week

Select domain of interest (e.g. College Management System) and identify multi-tier software applications to work on (e.g. Online Fee Collection). Analyze, design and develop this application:

1. Develop an SRS document. Also develop risk management and project plan (Gantt chart).
2. Understanding of System modeling: Data modeling using ER – Diagram-Draw an ER Diagram which also includes generalization, specialization and aggregation of specified problem statements.
3. Understanding of System modeling: Functional modeling: DFD Context and draw it
4. Understanding of System modeling: UML and draw it.
5. Develop a sample calculator program and perform Black-box Testing:
 - a. Identify test scenarios without viewing the internal code.
 - b. Write test cases for valid and invalid inputs.
 - c. Execute the test cases and record outcomes.
6. Develop a sample login authentication page and perform White--box Testing:
 - a. Analyze the code for all possible paths, conditions, and loops.
 - b. Apply statement coverage and branch coverage techniques.
 - c. Test internal functions with known inputs.
7. For the above login authentication page, perform the following:
 - a. Simulate the following maintenance activities:
 - **Corrective Maintenance:** Fix an existing bug (e.g., wrong output, crash, miscalculation).
 - **Perfective Maintenance:** Add a new user-requested feature (e.g., sorting, filter, or improved UI).
 - **Adaptive Maintenance:** Modify the code to adapt to a new platform or library version.
 - **Preventive Maintenance:** Refactor the code to improve readability, performance, or security.
 - b. Document each change with:
 - Problem description
 - Type of maintenance
 - Before and after screenshots
 - Change summary

KRISHNA UNIVERSITY:MACHILIPATNAM - 521004

B.C.A Honors Computer Applications [Major / Minor]

REVISED CBCS FRAMEWORK WITH EFFECT FROM 2025-2026 ADMITTED BATCH

COURSE 10: SOFTWARE ENGINEERING

Time: 3 hrs.

Max marks: 70

SECTION-A

Answer any **FIVE** out of the following

5 x 4 M= 20M

1. Define Software Engineering and mention advantages and limitations of waterfall model.
2. List the phases of Software Development Life Cycle (SDLC).
3. Define functional and non-functional requirements with examples.
4. What is Software Requirement Specification (SRS)? Mention its purpose.
5. Define cohesion and coupling.
6. What is UML? State its role in software design.
7. Define verification and validation in software testing.
8. Discuss metrics related to software maintenance.

SECTION – B

Answer **ALL** questions

5 × 10 = 50 M

9. Explain major software life cycle models such as Agile and Spiral models with comparison.

OR

10. Explain Prototyping development model and discuss advantages and limitations of it.
11. Explain feasibility study and requirements gathering techniques.

OR

12. Explain the process of Requirement Analysis and Specification with neat steps and examples.
13. Explain software design principles including modularity, layering and user interface design guidelines.

OR

14. Explain the role of Data Flow Diagrams (DFD) and Structure Charts in software design with neat diagrams.
15. Explain different levels of testing such as unit testing, integration testing, system testing and acceptance testing with examples.

OR

16. Describe test scenarios and test case design with automation vs manual testing.
17. Explain types of software maintenance – corrective, adaptive, preventive and perfective maintenance with examples.

OR

18. Discuss challenges in software maintenance and explain software maintenance metrics used for performance evaluation.

KRISHNA UNIVERSITY:: MACHILIPATNAM - 521004

B.C.A Honours Computer Applications [Major / Minor]

REVISED CBCS FRAMEWORK WITH EFFECT FROM 2025-2026 ADMITTED BATCH

BLUEPRINT

COURSE 10: SOFTWARE ENGINEERING

Unit	S.A.Q Section: A (including choice) 4 Marks	E.Q Section: B (including choice) 10 Marks	Marks Allotted
I	2	2	28
II	2	2	28
III	2	2	28
IV	1	2	24
V	1	2	24
Total	8	10	132



ANDHRA PRADESH STATE COUNCIL OF HIGHER EDUCATION

Model Syllabus for Computer Applications (Minor) in consonance with Curriculum framework w.e.f. AY 2025-26

COURSE STRUCTURE

Year	Semester	Course	Title of the Course	No. of Hrs /Week	No. of Credits
II	III	1	Computer Fundamentals and Office Automation	3	3
			Computer Fundamentals and Office Automation-Practical	2	1
	IV	2	Database Management System	3	3
			Database Management-Practical	2	1
III	V	3	Introduction to Web Programming	3	3
			Introduction to Web Programming-Practical	2	1
		4	Fundamentals of Python Programming	3	3
			Fundamentals of Python Programming-Practical	2	1
	VI	5	Cyber Security	3	3
			Cyber Security-Practical	2	1
		6	Data Visualization using Tableau & Power BI	3	3
			Data Visualization using Tableau & Power BI-Practical	2	1

SEMESTER-III

COURSE 1: COMPUTER FUNDAMENTALS AND OFFICE AUTOMATION

Theory

Credits: 3

3 hrs/week

Course Objectives:

1. **Understand foundational computing concepts** including number systems, evolution of computers, and architectural components.
2. **Explore basic computer organization and network fundamentals**, recognizing device functions, system types, and internet components.
3. **Demonstrate proficiency in word processing and presentation tools**, applying formatting techniques and design elements for professional outputs.
4. **Develop competency in spreadsheet operations**, employing formulas, charts, and data-handling techniques.
5. **Apply advanced data modeling and productivity features** to analyze and visualize data efficiently using modern tools.

Course Outcomes:

Learners will be able to:

1. **Convert between binary, decimal, octal, and hexadecimal systems**, and explain computer evolution and generations with examples.
2. Learners will demonstrate **basic blocks of a computer and fundamental networking knowledge**.
3. Create professional-level documents and **design visually appealing presentations** using word processing software and presentation software.
4. Manipulate data within spreadsheets, apply formulas, and **generate accurate summaries and visualizations**.
5. Apply data modelling techniques to **analyze, organize, and represent data effectively** in various scenarios.

Unit-I: Number Systems, Evolution , Block Diagram and Generations

Number Systems: Binary, Decimal, Octal, Hexadecimal; conversions between number systems.

Evolution of Computers: History from early mechanical devices to modern-day systems.

Block Diagram of a Computer: Input Unit, Central Processing Unit, Memory Unit, Output Unit.

Generations of Computers: First to Fifth Generation – Technologies, Characteristics, Examples.

Unit-II Basic Organization and Network Fundamentals

Computer Organization: Functional components: Input/Output devices, Storage types, Memory Hierarchy.

Types of Computers: Micro, Mini, Mainframe, and Supercomputers.

Networking Fundamentals: Definition, Need for Networks, **Key Components:** Nodes, Links, Protocols, Networking Devices. **Types of Computer Networks :** LAN, WAN, MAN.

Network Topologies: Bus, Ring, Star, Mesh..

Internet Basics: History, IP Address, URL, WWW, Web browsers, Search engines, E-mail, Internet Security.

Unit-III Word Processing and Presentations

Word Processing Basics: Definition, Using Microsoft Word / Google Docs. Templates for resumes, letters, reports. **Basic text editing and formatting** - Typing and editing text, Font styles, sizes, colors, and effects, Paragraph alignment, indentation, and spacing, Bullets, numbering, and text highlighting, Templates for resumes, letters and reports. **Working with Tables and Graphics** - Inserting and formatting tables, Adding images, shapes, icons, and SmartArt, Text wrapping and positioning graphics.

Document Layout and Design - Page setup, Headers, footers, and page numbering, Section breaks and columns, Applying themes and styles. **Advanced Features** - Spell check and grammar tools, Thesaurus, and Mail merge. **References and Citations** Footnotes, endnotes, and captions, Bibliography and citation tools, Table of contents and index creation.

Presentation Tools: Using PowerPoint/Google Slides – Creating, Saving and Opening presentations, Adding, deleting, and rearranging slides, Slide layouts and design themes, Using templates and master slides, Slide sorter and outline view, Applying transitions and Animations, Design and Layout.

Applications: Creating resumes, Reports, Brochures, and Presentations.

Unit-IV Spreadsheet Basics

Spreadsheet Concepts: Understanding rows, columns, cells in tools like MS Excel/Google Sheets, Workbook, Worksheet, **Cell referencing-** Relative, Absolute, Mixed.

Functions and Formulae: Mathematical, Statistical, Logical, Text, Date and Time, Financial.

Lookup and Reference : VLOOKUP, HLOOKUP, XLOOKUP, INDEX, MATCH

Visual representations: Creating a chart, common chart types, Column Chart, Bar Chart, Line Chart, Pie Chart, Scatter Chart, Histogram.

Data Handling: Sorting data, Filtering data, Grouping Data, **Conditional formatting:** Data Bars, Color Scales, Icon Sets, Custom Formulas.

Unit-V Data Modelling

Data Analysis Tools: Pivot Tables and Pivot Charts, Data Validation (Drop-downs, Input Messages, Error Alerts), **What-If Analysis:** Goal Seek, Scenario Manager, Data Tables

Charts and Dashboards: Creating Interactive Dashboards, Using slicers with Pivot Tables, Combo Charts and Sparklines.

Productivity Tips: Using Named Ranges, Freeze Panes, Split View.

Text Books:

1. **Thareja, R.** (Second Edition). *Fundamentals of Computers*. Oxford University Press.
2. **Rajaraman, V.** (n.d.). *Fundamentals of Computers*. PHI Learning.
3. **Norton, P.** (2017). *Introduction to Computers* (7th ed.). McGraw Hill Education.
4. **Nordell, R., Stewart, K., Easton, A., Graves, P. R., & Triad Interactive, Inc.** (2022). *Microsoft Office 365: In Practice* (1st ed.). New York: McGraw Hill Education.

References Books:

1. **Alexander, M., & Kusleika, R.** (2022). *Microsoft Excel 365 Bible* (2nd ed.). Wiley.
2. **Lowe, D.** (2021). *Networking All-in-One For Dummies* (8th ed.). Wiley.
3. **Microsoft Official Docs and Training:** <https://learn.microsoft.com>
4. **Google Workspace Learning Center:** <https://support.google.com/a/users/>

Activities:

Unit 1: Number Systems & Computer Evolution

Outcome: At the End of the Course, The Students will be able to **explain different number systems**, the historical evolution of computers, and identify key components in a block diagram.

Activity: Create a digital poster or infographic comparing number systems (binary, decimal, octal, hexadecimal) and illustrating the timeline of computer generations with key innovations.

Evaluation Method: Rubric-based assessment of the poster presentation on a 10-point scale focusing on:

- Accuracy of number system conversions
- Correct identification of block diagram components
- Visual organization and creativity

Unit 2: Computer Architecture & Networking Basics

Outcome: Learners will demonstrate **basic blocks of a computer and fundamental networking knowledge**.

Activity: Design a concept map showing the internal architecture of a computer and types of networks (LAN, WAN, MAN), including devices and topologies.

Evaluation Method: Checklist-based peer review and instructor validation:

- Completeness of the map
- Correctness of networking concepts
- Use of appropriate terminology
- Logical flow and structure of the map

Unit 3: Word Processing & Presentation Design

Outcome: Learners will create professional-level documents and **design visually appealing presentations** using word processing software and presentation software.

Activity: Prepare a formal report (e.g., project proposal) in a word processor and present it using a slide deck with transitions, embedded media, and design elements.

Evaluation Method: Performance-based evaluation using a 10-point scoring scale:

- Formatting and structure of the document
- Presentation aesthetics and clarity
- Communication skills during presentation

Unit 4: Spreadsheet Analysis & Visualization

Outcome: Learners will manipulate data within spreadsheets, apply formulas, and **generate accurate summaries and visualizations**.

Activity: Analyze a dataset (e.g., student scores or sales data) using spreadsheet software. Apply formulas (SUM, AVERAGE, IF, VLOOKUP) and create relevant charts.

Evaluation Method: Practical test with a rubric:

- Correct use of formulas
- Accuracy of data summaries

Unit 5: Data Analysis and Visualization:

Outcome: Learners will apply data modelling techniques to **analyze, organize, and represent data effectively** in various scenarios.

Activity: Prepare an interactive dashboard for a given data set using EXCEL.

Evaluation Method: Evaluation of the dashboard on a 10-point scoring scale:

- Presentation aesthetics and clarity
- Interactiveness
- Communication skills during presentation

SEMESTER-III

COURSE 1: COMPUTER FUNDAMENTALS AND OFFICE AUTOMATION

Practical

Credits: 1

2 hrs/week

List of Experiments:

1. Demonstration of Assembling and Desassembling of Computer Systems.
2. Identify and prepare notes on the type of Network topology of your institution.
3. Prepare your resume in Word by using the Resume template.
4. Using Word, write a letter to your higher official seeking 10-days leave.
5. Create a multi-page academic report and format it using headers and footers. The header will include the document title and author name, while the footer will contain page numbers and the date.
6. Prepare a formal invitation letter and use Mail Merge to personalize it for a list of recipients.
7. Prepare a report that includes: A table summarizing sales data, A graphic (image or chart) illustrating product performance with the proper formatting and alignment of both elements
8. Prepare a document and add Citations, Footnotes, and Bibliography in Word.
9. Create a PowerPoint Presentation on the Role of AI in Business Decision-Making.
10. Using a spreadsheet, prepare your class Time Table.
11. Using a Spreadsheet, calculate the Gross and Net salary of employees (Min 5) considering all the allowances.
12. Generate the class-wise and subject-wise results for a class of 20 students. Also generate the highest and lowest marks in each subject.
13. Using IF, AND, OR, and IFERROR to Automate Grade Evaluation.
 - a. Create a table of student scores in different subjects.
 - b. Use IF to assign grades (A/B/C/Fail).
 - c. Use IFERROR to handle missing scores or invalid data.
14. Consider the problem of preparing a stationary order for the month of March. The item description, quantity and cost per item are available. The total cost per item is to be calculated and the final cost per item involves a sales tax of 2% over the total cost. The gross total and the net total are to be displayed.

Sl. No.	Description	Quantity	Cost Per Item
1	Notepad	202	2.85
2	Writing Pad	86	3.95
3	Ball point pen (Blue)	520	2.50
4	Cello-tape	75	2.95

5	A4 Refill pad	90	5.95
6	Pencils	603	0.50
7	Crayons	80	3.85
8	Stapler	30	9.95
9	Hole punch	25	14.95
10	Ring Binder	45	10.95

15. You are given the order details of a company in the below table.

Order Id	Product	Unit price	Quantity	Discount	Revenue	Tax (2% for each order)	Net Income
11250	A	8	10	0%	?	?	?
11251	B	20.8	1	0%	?	?	?
11252	C	7.7	16	25%	?	?	?
11253	D	15.6	50	0%	?	?	?
11254	E	39.4	15	25%	?	?	?
				Total	?		?

- Calculate the revenue and tax on the revenue for each product.
- Calculate the net income of each product.
- Calculate the total revenue of all products.
- Calculate the total net income of all products.

16. Create an Excel sheet with the following fields in the Sales table.

i) Month ii) Item iii) Quantity iv) Price v) Commission

Use Data Validation criteria for:

- Quantity and Price should be whole numbers
- Commission @3.5% of Price should be allowed only two decimals.
- Price should accept 5000 and above values only.

17. Consider the problem of finding the total and average marks of five subject marks for five students. Calculate the Maximum mark, minimum mark, mean, median, Standard deviation and Variance for each subject.

Roll. No.	Name	Accounting	Income Tax	Business Law	Total	Average
100	Ramesh	85	75	60	?	?
101	Mahesh	100	78	85	?	?
102	Suresh	65	72	70	?	?
103	Ravi	90	80	85	?	?
104	Raju	80	76	90	?	?

18. The following are the details of Expenditure. Draw a Pie diagram with appropriate Formatting options, including Percentages and chart headings.

Product	Sales
Food	10000
Rent	5000
Clothing	1000
Fee	4000

19. The following are the marks obtained by the students of B. Com. In three subjects.

Roll. No.	Name	Accounts	IT	Economics
2001	Ramesh	65	85	75
2009	Mahesh	88	75	60
2004	Suresh	67	84	35
2002	Ravi	42	85	74
2007	Raju	88	89	90

1. Sort the above table on Roll. No.
2. Using Conditional formatting List out students who scored
 - a. Less than 55 in Accounts
 - b. More than 75 in IT
 - c. Between 60 and 75 in Economics.

20. Prepare Pivot Table for the given data:

Department	Employee Name	Salary
HR	Ramesh	20,000
Finance	Mahesh	18,500
IT	Suresh	17,500
HR	Ravi	13,000
Finance	Raju	15,000
IT	Balu	10,000

21. *Employee Database Search Using VLOOKUP, HLOOKUP, XLOOKUP, INDEX, and MATCH*

- a. Create a database of employees (Name, ID, Department, Salary).
- b. Implement VLOOKUP to search by employee ID.
- c. Use HLOOKUP to extract department heads by role.
- d. Apply XLOOKUP for more flexible searches.
- e. Use INDEX + MATCH as an alternative to VLOOKUP.

22. Sales Report Analysis Using Pivot Tables and Charts

- a. Use a dataset of product sales (Product, Region, Date, Quantity, Revenue).
- b. Create Pivot Tables to summarize data by region/product.
- c. Insert Pivot Charts for visual analysis (e.g., bar, line).
- d. Add slicers to make the dashboard interactive.

23. Designing a Data Entry Form with Drop-downs and Input Rules

- e. Create a student registration form.
- f. Add drop-down lists for course selection using Data Validation.

- g. Add input messages to guide users.
- h. Add error alerts for wrong entries.

24. Monthly Budget Planning using Goal Seek and Scenario Manager

- i. Create a simple personal budget (income, expenses, savings).
- j. Use Goal Seek to determine income needed to save a desired amount.
- k. Use Scenario Manager to compare different budgeting scenarios (best/ worst/ realistic case).
- l. Create a one-variable Data Table to analyze how different expenses affect savings.

25. Consider the monthly sales report

Month	Region	Product	Units Sold	Unit Price (₹)	Total Sales (₹)
Jan-25	North	Laptop	120	50,000	60,00,000
Jan-25	South	Tablet	80	30,000	24,00,000
Feb-25	North	Laptop	150	50,000	75,00,000
Feb-25	South	Tablet	90	30,000	27,00,000
Mar-25	North	Laptop	100	50,000	50,00,000
Mar-25	South	Tablet	110	30,000	33,00,000

- 1. Create PivotTables
- 2. Add Slicers
- 3. Create Combo Chart
- 4. Insert Sparklines
- 5. Assemble Dashboard

SEMESTER-III
KRISHNA UNIVERSITY :: MACHILIPATNAM - 521004
Computer Applications (Minor)
COURSE 1
MODEL QUESTION PAPER
Computer Fundamentals and Office Automation

Time: 3 hrs.

Max Marks: 70

SECTION-A

Answer any FIVE of the following questions

5 x 4 M = 20 M

1. Convert $(110101)_2$ to its decimal equivalent and $(58)_{10}$ to its binary equivalent.
2. Explain the block diagram of a computer with a neat diagram.
3. Discuss router and gateway.
4. Write about memory hierarchy.
5. What is mail merge? Explain its use in word processing.
6. Explain the difference between relative, absolute and mixed cell referencing in a spreadsheet.
7. What is conditional formatting? Explain with an example.
8. What is a Pivot Table? Explain its uses in data analysis.

SECTION-B

Answer ALL Questions

5 x 10 M = 50 M

9. Explain the evolution of computers through various generations, with examples.

OR

10. Explain the different number systems and demonstrate conversions among binary, octal, decimal, and hexadecimal.

11. Explain types of networks and its topologies in detail with neat diagrams,

OR

12. Explain Internet basics - IP address, URL, WWW, web browsers.

13. Explain the features of basic text editing and formatting using MS Word.

OR

14. Explain the features of presentation software used to create professional presentations - slides, themes, transitions, and animations.

15. Explain the various categories of spreadsheet functions and formulae with examples.

OR

16. Explain data handling techniques in spreadsheets.

17. Explain data analysis tools in spreadsheets - Pivot Tables and Data Validation

OR

18. Explain how interactive dashboards are created using charts, slicers in spreadsheets.

KRISHNA UNIVERSITY: MACHILIPATNAM - 521004

Computer Applications (Minor)

BLUEPRINT

Course 1

Computer Fundamentals and Office Automation

Unit	S.A.Q Section: A (including choice) 4 Marks	E.Q Section: B (including choice) 10 Marks	Marks Allotted
I	2	2	28
II	2	2	28
III	1	2	24
IV	2	2	28
V	1	2	24
Total	8	10	132

SEMESTER-IV

COURSE 2: DATABASE MANAGEMENT SYSTEMS

Theory

Credits: 3

3 hrs/week

Course Objectives:

1. Understand the fundamental concepts of data, databases, and the architecture of Database Management Systems (DBMS). Analyze and design database schemas using Entity-Relationship (E-R) and Extended E-R models.
2. Apply relational database principles, including normalization and integrity constraints, to ensure efficient schema design.
3. Develop SQL queries for data definition, manipulation, and control, including advanced operations like joins and nested queries.
4. Implement PL/SQL programming constructs and manage database transactions with a focus on ACID properties.

Course Outcomes:

The Learners will be able to:

1. Differentiate between file systems and DBMS, and explain the evolution, architecture, and components of modern database systems.
2. Design conceptual data models using E-R and Extended E-R diagrams, including specialization, generalization, and aggregation.
3. Normalize relational schemas up to BCNF and apply relational algebra operations to query and manipulate data.
4. Write efficient SQL queries using DDL, DML, DCL, and TCL commands, and utilize functions, joins, and views for data analysis.
5. Develop PL/SQL blocks with control structures, procedures, and triggers, and manage transactions ensuring atomicity, consistency, isolation, and durability.

Unit-I: Overview of Database Management System

Introduction: Data, Information, Database, Database Management System, Database System Applications, File Systems versus DBMS, Advantages of DBMS.

Evolution of Database Systems, Data Models, Data Abstraction, Database Architecture, Centralized and Client/Server Architectures for DBMSs, Database Users, Database Administrator, Components of DBMS, DBMS Vendors.

Unit-II: Database Design and the E-R Model

Overview of the Database Design Process, **The Entity-Relationship Model** : Entity Sets, Classification of Entity Sets, Attributes, Types of Attributes, Relationship, Relationship Set, Degree of a relationship set, Relationship Classification, Mapping cardinalities.

Extended Entity Relationship Model : Specialization, Generalization, Constraints, Attribute Inheritance and ISA relationship, Aggregation.

Case Study: Hospital Management System

Unit-III : Relational Database Design

Relational Model: Introduction to Relational Model, Concepts of Relation, Tuple, Attribute, Instance, Domain. Keys (Super key, Candidate Key, Primary Key, Foreign Key), Constraints (Domain constraints, Key constraints, Integrity constraints), selection, projection operations, Codd's rule set for relational databases.

Normalization: Purpose of Normalization/ Schema refinement, Functional dependencies, Normal Forms : 1NF, 2NF, 3NF and BCNF, Denormalization.

Case Study: College Student Enrollment System

Unit-IV : Introduction to SQL

Structured Query Language (SQL) : Overview of the SQL, Data types, Operators, Basic structure of SQL Query,

Commands: Data Definition Language (DDL), Data Manipulation Language (DML), Data Control Language (DCL), Transaction Control Language (TCL).

Functions: Numeric, String, Date Functions. Set operators, Aggregate functions, Nested queries, Joins, GROUP By Clause, ORDER By Clause, views.

Case Study : Retail Store Database Management (Design and query a relational database for a retail store that manages products, customers, orders, and employees. Apply SQL concepts to extract insights, maintain data integrity, and support decision-making.)

Unit-V Advanced SQL and Transaction Management

PL/SQL: Structure of PL/SQL block, Control Structures, Procedures, Functions, Exception handling, Cursors, Triggers.

Transactions : Transaction concept, Simple transaction model, State diagram of a transaction, ACID properties: Atomicity, Consistency, Isolation, Durability.

Textbooks:

1. Database System Concepts, Avi Silberschatz, Henry F. Korth, S. Sudarshan, Seventh Edition, McGraw-Hill
2. Database Management Systems by Raghu Ramakrishnan, McGrawhill

Reference Books:

1. Fundamentals of Database Systems, Elmasri Navathe Pearson Education
2. An Introduction to Database systems, C.J. Date, A.Kannan, S.Swami Nadhan, Pearson

Activities:

Outcome: Describe the fundamentals of data, database systems, and the differences between file-based and database approaches. Compare and classify various DBMS architectures, data models, and their components, including the three-schema architecture.

Activity: Create a comparative presentation or infographic illustrating:

- File-based vs. DBMS approaches
- Types of DBMS architectures (1-tier, 2-tier, 3-tier)
- Data models and the three-schema architecture

Evaluation Method: Rubric-based assessment of the presentation covering clarity, accuracy, and depth of comparison. Include a short quiz to test conceptual understanding.

Outcome: Design conceptual data models using Entity-Relationship and Enhanced ER diagrams, applying generalization, specialization, and constraints.

Activity: Model a university or hospital database using ER and Enhanced ER diagrams that shows:

- Entity sets, relationships
- Generalization/specialization
- Participation and cardinality constraints

Evaluation Method: Diagram submission with peer review and instructor feedback. Use a checklist to assess completeness, correctness, and notation usage.

Outcome: Apply relational model concepts, including CODD rules, relational algebra, relational calculus, and normalization techniques.

Activity: Normalize a given unstructured dataset up to 3NF. Then, write relational algebra expressions for sample queries.

Evaluation Method: Written assignment graded on:

- Correctness of normalization steps
- Accuracy of relational algebra expressions
- Short-answer questions on CODD rules and relational calculus

Outcome: Construct and execute SQL queries for data definition, manipulation, aggregation, joining, and subqueries, including views and set operations.

Activity: Implement a mini-project (e.g., Library or Inventory DB) using SQL. Include:

- Table creation (DDL)
- Data manipulation (DML)
- Aggregation, joins, subqueries, views, and set operations

Evaluation Method: Lab-based practical test with query execution and output validation. Include a viva to explain logic and optimization.

Outcome: Develop PL/SQL programs incorporating control structures, procedures and functions to manage database behaviour effectively.

Activity: Build a PL/SQL-based payroll or student grading system using:

- Procedures and functions
- Control structures (IF, LOOP)
- Triggers for automated updates

Evaluation Method: Code review and demonstration. Evaluate based on:

- Syntax correctness
- Logical flow

SEMESTER-IV

COURSE 2: DATABASE MANAGEMENT SYSTEMS

Practical

Credits: 1

2 hrs/week

Experiment 1 : Database: Inventory Management

Table 1: Products

Structure:

Column Name	Data Type	Constraints
product_id	INT	PRIMARY KEY
product_name	VARCHAR(50)	NOT NULL
price	DECIMAL(10,2)	CHECK(price > 0)
stock_qty	INT	CHECK(stock_qty >= 0)

Sample Data:

product_id	product_name	price	stock_qty
1	Pen	10.00	100
2	Notebook	50.00	200
3	Stapler	120.00	50
4	Marker	25.00	80
5	File Folder	60.00	150

Table 2: Suppliers

Structure:

Column Name	Data Type	Constraints
supplier_id	INT	PRIMARY KEY
supplier_name	VARCHAR(50)	NOT NULL
contact_no	VARCHAR(20)	UNIQUE

product_id	INT	FOREIGN KEY REFERENCES Products(product_id)
------------	-----	---

Sample Data:

supplier_id	supplier_name	contact_no	product_id
101	StationeryMart	9876543210	1
102	PaperWorld	9876500000	2
103	OfficeSupplies	9876512345	3
104	MarkerHub	9876522222	4
105	FileDepot	9876533333	5

Section A: DDL (Data Definition Language)

1. Create a database called InventoryDB.
2. Create a table Products and table Suppliers with the specified columns and constraints:

Section B: DML (Data Manipulation Language)

4. Insert at least 5 rows into the Products table.
5. Insert at least 5 rows into the Suppliers table.
6. Update the stock quantity of product 'Pen' to 120.
7. Delete a supplier with a specific supplier_id.
8. Write a query to rename 'Notebook' to 'NoteBook A4'

Section C: DQL (SELECT Queries)

9. Display all records from the Products table.
10. Display only product_name and price of all products.
11. List all products that have a stock quantity less than 100.
12. Show all products between 20 and 100 price range.
13. Find all suppliers whose contact number starts with '98765'.
14. Find the average price of products.
15. Display the total number of products in the inventory.
16. Show the maximum and minimum stock quantities.

17. Count how many suppliers supply each product.
18. Show all products where price > 50 AND stock_qty > 100.
19. Show all products where price < 20 OR stock_qty < 80.
20. Display suppliers whose supplier_name contains the word 'Mart'
21. List all suppliers along with the product they supply (use INNER JOIN).
22. Display suppliers whose name starts with 'S'.
23. Find products whose name has exactly 5 characters
24. Find suppliers who supply products costing more than 100.

Experiment 2 : ONLINE BOOKSTORE DB

An online book store wants to implement a **BOOKSTORE DB** for managing their online transactions by using the following tables.

Authors Table

Column Name	Data Type	Constraints
author_id	INTEGER	PRIMARY KEY
first_name	VARCHAR	NOT NULL
last_name	VARCHAR	NOT NULL
nationality	VARCHAR	NULL allowed

Books Table

Column Name	Data Type	Constraints
book_id	INTEGER	PRIMARY KEY
Title	VARCHAR	NOT NULL
author_id	INTEGER	FOREIGN KEY REFERENCES Authors
publication_year	INTEGER	
Price	DECIMAL	

Customers Table

Column Name	Data Type	Constraints
customer_id	INTEGER	PRIMARY KEY
first_name	VARCHAR	NOT NULL
last_name	VARCHAR	NOT NULL
Email	VARCHAR	UNIQUE, NOT NULL
Address	VARCHAR	NOT NULL

Orders Table

Column Name	Data Type	Constraints
order_id	INTEGER	PRIMARY KEY
customer_id	INTEGER	FOREIGN KEY REFERENCES Customers
book_id	INTEGER	FOREIGN KEY REFERENCES Books
order_date	DATE	NOT NULL
quantity	INTEGER	NOT NULL

SAMPLE DATA SET for BOOKSTORE DB

Authors Table

author_id	first_name	last_name	nationality
1	Jane	Austen	British
2	George	Orwell	British
3	Gabriel	Garcia Marquez	Colombian
4	Toni	Morrison	American
5	Mark	Twain	American

6	Harper	Lee	American
7	Fyodor	Dostoevsky	Russian

Books Table

book_id	Title	author_id	publication_year	price
101	Pride and Prejudice	1	1813	12.99
102	1984	2	1949	9.50
103	One Hundred Years of Solitude	3	1967	15.00
104	Beloved	4	1987	11.25
105	Animal Farm	2	1945	8.75
106	Adventures of Huckleberry Finn	5	1884	10.50
107	To Kill a Mockingbird	6	1960	14.00

Customers Table

customer_id	first_name	last_name	Email	address
201	Alice	Smith	alice.s@example.com	12 Oak St, London
202	Bob	Johnson	bob.j@example.com	45 Pine Ave, Oxford
203	Charlie	Brown	charlie.b@example.com	78 Maple Rd, Bristol
204	Diana	Prince	diana.p@example.com	34 Queen St, York
205	Edward	Norton	edward.n@example.com	22 River Ln, Leeds
206	Fiona	Hall	fiona.h@example.com	56 Lake Dr, Bath
207	Greg	Miller	greg.m@example.com	89 Park Ave, Glasgow

Orders Table

order_id	customer_id	book_id	order_date	Quantity
301	201	101	2025-07-20	1
302	202	102	2025-07-21	2
303	201	105	2025-07-22	1
304	203	103	2025-07-23	1
305	204	106	2025-07-24	1
306	205	107	2025-07-25	3
307	206	104	2025-07-26	2

Section A: DDL (Schema Design & Constraints)

1. Write SQL statements to create all 4 tables (Authors, Books, Customers, Orders) with:
 - o Primary Keys
 - o Foreign Keys
 - o Appropriate data types
 - o NOT NULL constraints where necessary.
2. Alter the Books table to add a constraint that price must be greater than 0.
3. Add a new column phone_number to the Customers table (VARCHAR(15)) and ensure it is unique.
4. Drop the phone_number column from the Customers table.

Section B: DML (Data Manipulation)

5. Insert at least 7 records for each table (use sample dataset above).
6. Update the price of the book titled *Animal Farm* by increasing it by 10%.
7. Delete all orders made before 2025-07-21.
8. Change the nationality of Gabriel Garcia Marquez to “Latino-American”.

Section C: SELECT Queries (Data Querying)

9. List all books published between 1900 and 2000.
10. Find all customers whose email contains “example.com”.
11. Retrieve books whose price is between 10 and 15 and published before 1950.

12. Show authors who are either 'British' or 'American'.
13. Find books that have a price less than 10 or are published after 1980.
14. Display all orders placed after 2025-07-22.
15. List all books written by author with author_id = 2.
16. Find customers whose last name starts with B.
17. Show all books with a price NOT between 9 and 13.
18. Display books whose publication_year is in (1813, 1945, 1987).
19. Find authors whose nationality is NOT 'British'.
20. List customers whose address contains the word Park.
21. Show all books sorted by price in descending order.
22. List authors in alphabetical order by last_name.
23. Display orders sorted by order_date (latest first).

Use of Date Functions

24. Show all orders placed in July 2025.
25. Show all orders with an estimated delivery date (5 days after order date).
26. Show customers who placed an order on a weekend.
27. Calculate how many days have passed since the last order was placed.

Aggregate Functions (COUNT, SUM, AVG, MIN, MAX)

28. Count the total number of books in the database.
29. Find the average price of all books.
30. Show the highest-priced book.
31. Count how many orders each customer has placed.
32. Calculate the total sales (price × quantity) for each customer.

GROUP BY and HAVING

33. Count how many books are written by each author.
34. Group orders by customer_id and display total quantity ordered.
35. Show customers who have ordered more than 2 books in total (use HAVING).
36. Find the total number of books sold per author (GROUP BY author).

Experiment 3: EMPLOYEE DB

An enterprise wants to automate its employee management process by implementing an Employee Database. The goal is to replace manual record-keeping with a centralized system that stores employee, department, and project details. Use the following table structures and data set to implement Employee DB.

EmployeeDB – Table Structures

1. Departments Table

Column	Type	Constraints
dept_id	INT	PRIMARY KEY
dept_name	VARCHAR	UNIQUE, NOT NULL
location	VARCHAR	NOT NULL

2. Employees Table

Column	Type	Constraints
emp_id	INT	PRIMARY KEY
first_name	VARCHAR	NOT NULL
last_name	VARCHAR	NOT NULL
email	VARCHAR	UNIQUE, NOT NULL
phone	VARCHAR	CHECK (phone LIKE '--____')
hire_date	DATE	NOT NULL
job_title	VARCHAR	NOT NULL
salary	DECIMAL	CHECK (salary > 0)
dept_id	INT	FOREIGN KEY REFERENCES Departments(dept_id)
manager_id	INT	FOREIGN KEY REFERENCES Employees(emp_id) (self-referential)

3. Projects Table

Column	Type	Constraints
project_id	INT	PRIMARY KEY
project_name	VARCHAR	NOT NULL
start_date	DATE	NOT NULL
end_date	DATE	NULL

dept_id	INT	FOREIGN KEY REFERENCES Departments(dept_id)
---------	-----	--

4. Employee_Project Table (Many-to-Many)

Column	Type	Constraints
emp_id	INT	FOREIGN KEY REFERENCES Employees(emp_id), PRIMARY KEY(emp_id, project_id)
project_id	INT	FOREIGN KEY REFERENCES Projects(project_id)
hours_allocated	INT	CHECK (hours_allocated > 0)

Sample Data Set

Departments Table

dept_id	dept_name	Location
1	HR	New York
2	IT	San Francisco
3	Finance	Chicago
4	Marketing	Boston
5	Operations	Seattle
6	Legal	Washington D.C.
7	Sales	Dallas
8	R&D	Austin
9	Procurement	Denver
10	Customer Care	Miami

2. Employees Table

emp_id	first_name	last_name	Email	phone	hire_date	job_title	salary	dept_id	manager_id
101	Alice	Johnson	alice.j@corp.com	123-456-7890	2020-03-15	HR Manager	75000	1	NULL
102	Bob	Smith	bob.s@corp.com	234-567-8901	2019-05-20	IT Analyst	65000	2	104
103	Charlie	Brown	charlie.b@corp.com	345-678-9012	2021-01-10	Finance Executive	58000	3	106
104	Diana	Prince	diana.p@corp.com	456-789-0123	2018-07-12	IT Manager	90000	2	NULL
105	Ethan	Hunt	ethan.h@corp.com	567-890-1234	2022-02-25	Marketing Lead	62000	4	NULL
106	Fiona	Hall	fiona.h@corp.com	678-901-2345	2017-11-01	Finance Manager	85000	3	NULL
107	Greg	Miles	greg.m@corp.com	789-012-3456	2023-04-15	IT Support	45000	2	104
108	Hannah	White	hannah.w@corp.com	890-123-4567	2021-09-05	HR Executive	50000	1	101
109	Ian	Scott	ian.s@corp.com	901-234-5678	2020-11-20	Operations Analyst	56000	5	NULL
110	Julia	Adams	julia.a@corp.com	012-345-6789	2019-12-18	Legal Advisor	70000	6	NULL

3. Projects Table

project_id	project_name	start_date	end_date	dept_id
201	Payroll System	2023-01-01	NULL	3
202	Website Upgrade	2023-02-10	NULL	2
203	Recruitment Drive	2023-03-05	NULL	1
204	Ad Campaign	2023-05-20	NULL	4
205	New CRM Tool	2023-04-15	NULL	7
206	Compliance Portal	2023-06-10	NULL	6
207	Inventory System	2023-07-01	NULL	5
208	AI Research	2023-08-05	NULL	8
209	Customer Feedback	2023-09-10	NULL	10
210	Procurement System	2023-10-01	NULL	9

4. Employee_Project Table

emp_id	project_id	hours_allocated
102	202	120
104	202	80
103	201	100
106	201	150
101	203	50
105	204	70
107	202	60
109	207	90
110	206	110

108	203	40
-----	-----	----

Section A: DDL (Schema Creation & Modification)

1. Write SQL statements to create the above tables with the specified constraints
2. Alter the Employees table to add a column bonus DECIMAL(8,2) with default value 0.
3. Drop the column bonus from Employees.

Section B: DML (Insert, Update, Delete)

4. Insert at least 10 rows into Departments, Employees, Projects, and Employee_Project.(use the above data set)
5. Try inserting an employee with a negative salary (should fail due to CHECK constraint).
6. Update the salary of the employee with emp_id = 103 by 15%.
7. Delete an employee record who has resigned (choose any emp_id).
8. Increase all employees' salaries in the IT department by 5%.
9. Change the department of an employee to "Research".(should fail due to FK constraint)

Section C: DQL (Select Queries)

10. List all employees and their details.
11. Show all employees in the "HR" department.
12. Find employees with salaries between 50,000 and 80,000.
13. Retrieve employees hired after 2020.
14. Show employees who are in either the IT or Finance department.
15. Find employees whose email ends with "@corp.com".
16. List all employees with salary > 60,000 AND located in "New York".
17. Display employees in descending order of salary.
18. Count the number of employees in each department.
19. Show the average salary of employees department-wise.

20. Display departments where the average salary is greater than 70,000.
21. Find the number of employees in each project.
22. Display departments with more than 3 employees.
23. Show the sum of all salaries department-wise.
24. List all distinct department IDs from the Employees table.
25. Show employee names with the year they were hired.
26. Show employees grouped by the year of hire.
27. List employees hired in the last 90 days.
28. List the no of years of experience of all the employees

Section D: Joins

29. List all employees with their department names (INNER JOIN).
30. Display all departments along with employees, including those departments without employees (LEFT JOIN).
31. Show employees and the projects they are working on (JOIN 3 tables: Employees, Employee_Project, Projects).
32. List projects along with total hours allocated by employees.
33. Write a query to find employees who are working on more than one project.
34. Show all projects handled by the 'Finance' department.

Section E: PL/SQL Programming

1. Write a procedure GetEmpInfo that takes emp_id as input and displays name, salary, and department.
2. Write a PL/SQL block that checks if an employee's salary is above 50,000. If yes, print "High Salary" ;Otherwise print "Standard Salary".
3. Write a PL/SQL program to display the top 10 rows in the Emp table based on their job and salary
4. Write a stored procedure GiveBonus that takes department ID and a designation as input, along with a bonus amount, and updates the salary of all employees in that department who have the specified designation by adding the bonus amount to their current salary.
5. Create a trigger to prevent inserting employees with a salary less than 30,000.
6. Create a trigger to avoid any transactions(insert, update, delete) on EMP table on Saturday & Sunday.

SEMESTER-IV
KRISHNA UNIVERSITY :: MACHILIPATNAM - 521004
Computer Applications (Minor)
MODEL QUESTION PAPER
COURSE 2: Database Management Systems

Time: 3 hrs.

Max Marks: 70

SECTION-A

*Answer any **FIVE** of the following questions*

5 x 4 M = 20 M

1. Differentiate between file systems and DBMS.
2. Explain Data Abstraction.
3. Define ISA relationship with an example.
4. What is aggregation in the E-R model? Explain with an example.
5. Define functional dependency with an example.
6. What is a view in SQL? Explain with an example.
7. Explain the GROUP BY and ORDER BY clauses in SQL.
8. What are the Cursors in PL/SQL.

SECTION-B

*Answer **ALL** Questions*

5 x 10 M = 50 M

9. Explain the components of DBMS with a neat diagram.

OR

10. Explain the data models in detail.

11. Explain the E-R model with entity sets, attributes, relationships and mapping cardinalities, using a suitable example.

OR

12. Discuss the extended E-R model - specialization, generalization and aggregation using a suitable example.

13. Define normalization. Discuss 1NF, 2NF, 3NF and BCNF with examples.

OR

14. Explain the selection and projection operations with examples.

15. Explain the SQL commands under DDL and DML with examples.

OR

16. Explain joins in SQL with suitable examples.

17. Explain the structure of a PL/SQL block. Discuss functions in PL/SQL.

OR

18. Explain the transaction concept and the ACID properties of a transaction, with a suitable example.

KRISHNA UNIVERSITY :: MACHILIPATNAM - 521004

Computer Applications (Minor)

REVISED CBCS FRAMEWORK WITH EFFECT FROM 2025-2026 ADMITTED BATCH

BLUEPRINT

Database Management Systems

Unit	S.A.Q Section: A (including choice) 4 Marks	E.Q Section: B (including choice) 10 Marks	Marks Allotted
I	2	2	28
II	2	2	28
III	1	2	24
IV	2	2	28
V	1	2	24
Total	8	10	132

SEMESTER-V

COURSE 3: INTRODUCTION TO WEB PROGRAMMING

Theory

Credits: 3

3 hrs/week

Course Objectives

1. Understand the principles of web design and distinguish between web and desktop application architectures. Develop static web pages using HTML elements, attributes, and multimedia integration techniques.
2. Style web pages effectively using CSS, including layout control, responsive design, and UI enhancements.
3. Implement dynamic behaviors and form validations using JavaScript and the Document Object Model (DOM).
4. Explore client-side scripting techniques to build responsive, interactive interfaces.
5. Gain foundational knowledge of Content Management Systems (CMS) and apply practical skills in platforms such as WordPress.

Course Outcomes

At the End of the Course, The Students will be able to:

1. Design and structure HTML-based webpages incorporating text, images, tables, forms, and multimedia content.
2. Apply CSS styling rules to manage layout aesthetics, interactivity, and responsiveness across devices.
3. Use JavaScript for string manipulation, event handling, arrays, object operations, and basic validation.
4. Employ client-side scripting to enhance form functionality, create dialog interactions, and add animation via mouse and keyboard events.
5. Analyze different types of CMS and operate WordPress features like posts, pages, themes, plug-ins, and SEO tools to create and deploy basic websites.

Unit 1.HTML:

Introduction to web designing, difference between web applications and desktop applications, introduction to HTML, HTML structure, elements, attributes, headings, paragraphs, images, tables, lists, blocks, symbols, embedding multi-media components in HTML, HTML forms

Unit 2.CSS:

CSS home, introduction, syntax, CSS combinators, colors, background, borders, margins, padding, height/width, text, fonts, tables, lists, position, overflow, float, pseudo class, pseudo elements, opacity, tool tips, image gallery, CSS forms, CSS counters.

Unit 3.Java Script:

What is DHTML, JavaScript, basics, variables, operators, statements, string manipulations, mathematical functions, arrays, functions. objects, regular expressions, exception handling.

Unit 4. Client-Side Scripting:

Accessing HTML form elements using Java Script object model, basic data validations, data format validations, generating responsive messages, opening windows using java script, different kinds of dialog boxes, accessing status bar using java script, embedding basic animative features using different keyboard and mouse events.

Unit 5. Content Management Systems

Introduction to CMS: What is a CMS?, Types: traditional, headless, cloud-based

Popular CMS Platforms: WordPress, Joomla, Drupal, Shopify, When to choose each

Wordpress Basics: Introduction to word press, features, and advantages, wordpress (hosted access and local access), working with posts, managing pages, working with media, working with widgets, working with themes, extending wordpress with plug-ins, SEO and deployment.

Text Book(s)

1. Web Programming Building Internet Applications, Chris Bates, Second Edition, Wiley
2. An Introduction to Web Design plus Programming, Paul S.WangSanda S. Katila, Thomson.

Reference Books

1. Head First HTML and CSS, Elisabeth Robson, Eric Freeman, O'Reilly Media Inc.
2. An Introduction to HTML and JavaScript: for Scientists and Engineers, David R. Brooks. Springer, 2007
1. Schaum's Easy Outline HTML, David Mercer, Mcgraw Hill Professional.
2. Wordpress for Beginners, Dr.Andy Williams

SEMESTER-V

COURSE 3: INTRODUCTION TO WEB PROGRAMMING

Practical

Credits: 1

2 hrs/week

List of Experiments:

1. Create an HTML document with the following formatting options:
 - (a) Bold, (b) Italics, (c) Underline, (d) Headings (Using H1 to H6 heading styles), (e) Font (Type, Size and Color), (f) Background (Colored background/Image in background), (g) Paragraph, (h) Line Break, (i) Horizontal Rule, (j) Pre tag
2. Create an HTML document which consists of:
 - (a) Ordered List (b) Unordered List (c) Nested List (d) Image
3. Create a Table with four rows and five columns. Place an image in one column.
4. Collect any ten images of your choice. Using table tag, align the images as follows:



Using table tag, align the images as follows:

5. Create a menu form using HTML.
6. Style the menu buttons using CSS.
7. Create a form using HTML which has the following types of controls:
 - (a) Text Box (b) Option/radio buttons (c) Check boxes (d) Reset and Submit buttons
8. Embed a calendar object in your web page.
9. Create a form that accepts the information from the subscriber of a mailing system.

Word press:

10. Installation and configuration of word press

11. Access admin panel and manage posts
12. Access admin panel and manage pages
13. Add widgets and menus
14. Create users and assign roles
15. Create a site and add a theme to it

SEMESTER-V

COURSE 4: FUNDAMENTALS OF PYTHON PROGRAMMING

Theory

Credits: 3

3 hrs/week

Course Objectives

1. **Introduce the foundational concepts** of Python programming including its syntax, IDEs, and control structures.
2. **Develop proficiency in modular programming** using functions, lambda expressions, recursion, and Python's built-in modules and packages.
3. **Explore core data structures** like strings, lists, tuples, and dictionaries for effective data manipulation.
4. **Teach exception handling mechanisms** and the use of regular expressions for pattern matching and text processing.
5. **Enable students to interact with files and visualize data** using Python to build real-world applications involving persistent storage and data retrieval.

Course Outcomes

At the end of the Course, the students will be able to

1. **Write and execute structured Python programs** using variables, expressions, and flow control statements.
2. **Implement modular code** leveraging functions, argument types, recursion, and reusable libraries.
3. **Manipulate and organize data efficiently** using Python's string operations and complex data structures.
4. **Handle runtime errors and apply regular expressions** for robust and flexible program behaviour.
5. **Perform file operations and visualize data** through Python scripts to store, manage and visualize data effectively.

Unit 1. Basics of Python Programming:

Features of python, history of python, Python IDEs, Writing and Executing Python Program, literal constants, variables and identifiers, Data types, input operation- comments, Reserved words, Indentation, Operators and Expressions: Expressions in python, Operations in Strings, Other Data types, Type conversion, Decision control Statements, Iterative Statements, Nested loops, break, Continue, Pass Statements, else statement used with loops.

Unit 2. Strings and Collections:

Strings: Built-in String Methods and Functions

Lists: Access values in List, Updating values in Lists, Nested lists, Basic list operations, List Methods.

Tuple: Creating, Accessing, Updating and Deleting Elements in a tuple, Nested tuples.

Dictionaries: Creating a dictionary, Accessing values, Modifying an Entry, Deleting items, Built-in Dictionary Functions and Methods

Unit 3. Functions and Modules:

Function Definition, Function Call, Variable Scope and lifetime, The return statement, Required Arguments, Keyword Arguments, Default Arguments and Variable Length Arguments, Lambda Functions, Recursive Functions.

Importing Libraries, Modules, Packages in python, Standard library modules- Globals(), Locals(), and Reload(), Function Redefinition.

Unit 4. Exception Handling:

Errors and Exceptions, Handling Exceptions, Multiple Except blocks, Multiple Exceptions in a single block, Except Block without Exception, The else clause, Built-in and user-defined Exceptions, The finally block, Re-raising Exception, Assertions in python

Unit 5. File Handling & Data Visualization:

Types of files in Python, Opening and Closing files, Reading and Writing files: write() and writelines() methods- append() method, read() and readlines() methods, Splitting words, File Positions.

Introduction to matplotlib: Various plots - bar, line, histogram, box plot, customization, styling

Textbooks:

1. Python Programming using Problem Solving Approach, Reema Thareja, Oxford University Press 2020
2. Exploring Python, Budd T A, McGraw-Hill Education, 1st Edition, 2011.

Reference Book:

1. Python: The Complete Reference, Martin C. Brown, McGraw-Hill, 2018
2. Fundamentals of Python, Kenneth A. Lambert. (2019), First Programs, 2nd Edition, CENGAGE Publication.

Activities:

Outcome: Write and execute structured Python programs using variables, expressions, and flow control statements.

Activity: Create a calculator program that uses variables, arithmetic expressions, and flow control (if-else) to perform basic operations (add, subtract, multiply, divide) based on user input.

Evaluation Method: Code walkthrough and output validation. Use a checklist to assess on a 10-point scale to check the:

- Correct use of variables and expressions
- Proper flow control logic
- Accurate results for different inputs

Outcome: Implement modular code leveraging functions, argument types, recursion, and reusable libraries.

Activity: Build a factorial calculator using both iterative and recursive functions. Include parameterized functions and import the math library for comparison.

Evaluation Method: Code review and oral explanation. Assess on 10-point scale based on:

- Function structure and argument usage
- Recursion logic
- Use of reusable libraries

Outcome: Manipulate and organize data efficiently using Python's string operations and complex data structures.

Activity: Develop a contact manager that stores names and phone numbers using dictionaries and lists. Include string formatting and search functionality.

Evaluation Method: Practical demo with test cases. Evaluate based on:

- Use of string methods (split, join, format)
- Data structure selection and manipulation
- Search and retrieval accuracy

Outcome: Handle runtime errors and apply regular expressions for robust and flexible program behaviour.

Activity: Create a form validator that checks email and phone number formats using regular expressions. Include try-except blocks to handle invalid inputs.

Evaluation Method: Scenario-based testing. Assess based on:

- Regex accuracy for pattern matching
- Robust error handling
- Program stability with edge cases

Outcome: Perform file operations and visualize data through Python scripts to store, manage and visualize data effectively.

Activity: Build a student grade logger that reads from a file, stores data in a file, and visualizes the data.

Evaluation Method: Lab test with sample data. Evaluate on a 10-point scale:

- File read/write operations
- Visualization of data stored in files.

SEMESTER-V

COURSE 4: FUNDAMENTALS OF PYTHON PROGRAMMING

Practical

Credits: 1

2 hrs/week

List of Experiments

1. Display a welcome message using print().
2. Declare and use identifiers belonging to strings, integers, floats, and booleans.
3. Accept user input (name, age, height, student status) and display each value with its type using type().
4. Perform operations like .upper(), .find(), .replace() on strings.
5. Write a program to reverse the string, count vowels and words.
6. Write a program for slicing, sorting, and list comprehension.
7. Program to apply list methods: append(), extend(), insert(), remove(), pop(), sort().
8. Create tuples to store student (name, age, course) data and perform
 - a. Accessing elements using indexing and slicing.
 - b. Demonstrate immutability by attempting to modify a tuple.
 - c. Create and navigate nested tuples.
9. Create a dictionary with student roll number as keys and names/marks as values.
 - a. Accessing, adding, updating, and deleting key-value pairs.
 - b. Iterating through keys, values, and items.
10. Write a program to demonstrate variable length arguments.
11. Write a program to illustrate lambda and recursive functions.
12. Write a program to demonstrate Globals(), Locals(), and Reload() functions.
13. Demonstrate exception handling and assertions in Python.
14. Write a Python program to copy the contents of one file into another in reverse order.
15. Working with Datasets using Numpy and Pandas
16. Use matplotlib to generate line, bar, pie, and scatter graphs for the data stored in files.

SEMESTER-VI

COURSE 5: CYBER SECURITY

Theory

Credits: 3

3 hrs/week

Course Objectives

1. Understand the fundamental concepts of cybersecurity, cybercrimes, and cyber laws.
2. Identify vulnerabilities, threats, and security measures in modern information systems.
3. Explore emerging technologies and their applications in enhancing cybersecurity and digital trust.

Course Outcomes

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

1. Explain key concepts of cybersecurity, types of cybercrimes, and the role of national/international policies.
2. Identify and assess system vulnerabilities and apply basic security management practices.
3. Implement data protection, secure system design, and apply cybersecurity tools and technologies.
4. Analyze the impact of emerging technologies such as AI, blockchain, and quantum computing in the cybersecurity domain.

Unit 1. Introduction Security and Cyber Crimes

Cyberspace and Cybersecurity, What is Information Security? Organization and Governance of the Internet and Cybersecurity, Information Security Models

What is a Cybercrime?, Classification of Cybercrimes , Forms of Cybercrimes , Cyber Scams and Frauds , Sources of Threats: Threat Actors and their Motivations , Tools and Methods Used in Cyberattacks/Cybercrimes , What is a Cyberattack? , Responding to Cyberattacks and the Cyber Kill Chain, National Cybersecurity Policy , Online Code of Conduct and Computer Ethics

Unit 2. Cybersecurity Vulnerabilities and Security Management

Types of Vulnerabilities, Project OWASP, Vulnerabilities Assessment, Common Vulnerabilities and Exposures (CVE). Overview of Cybersecurity Management, Information Classification Process, Security Policies, Security Procedures and Guidelines, Security Controls, Security Organization, Incident Response, Business Continuity and Disaster Recovery

Unit 3. Developing Secure Information Systems

Introduction, Securing Information Assets, Data Security and Protection, Application Security, Security Architecture and Design, Security Issues in Hardware, Mobile Devices and Internet of Things, Techniques/Methods for Data Security and Protection, Issues Related to Digital File Sharing

Unit 4. Cybersecurity Technologies:

Introduction, Securing Networks, Web Applications, Services and Servers, Email Security, Antivirus Technologies and Solutions, Identity and Access Management, Authentication, Cryptography, How Do Digital Money, Cryptocurrency and NFTs Work?, Digital Signatures.

Unit 5: Cyber Laws and Cyber Security applications

Need for Cyber Laws and Regulations, Role of International Law and Governments, Challenges for Law-makers and Law Enforcement Agencies, Cybersecurity Regulations

Web 3.0, Harnessing Artificial Intelligence for Cybersecurity, Blockchain for Cybersecurity, Quantum Computing and Cybersecurity, Combating Advanced Persistent Threats, Digital Trust and Identity Management, 5G Networks and Cybersecurity.

Text Books:

1. Introduction to Cybersecurity: Concepts, Principles, Technologies and Practices, Ajay Singh, 2023, Publisher: Universities Press, ISBN 9789393330314

Reference Book:

2. The Art of Invisibility: The World's Most Famous Hacker Teaches You How to Be Safe in the Age of Big Brother and Big Data. – Kevin Mitnick, 2017, Publisher: Little, Brown and Company, ISBN 978-0316380508

Activities

Outcome: Explain key concepts of cybersecurity, types of cybercrimes, and the role of national/international policies.

Activities:

- Seminar on types of cybercrimes and real-world case studies.
- Group discussion on cybersecurity policies (India's IT Act, GDPR, etc.).
- Quiz on definitions, cyber laws, and key terms.

Evaluation Methods:

- Written quiz/test (MCQs and short answers).
- Group presentation assessment.
- Evaluation of seminar participation and content depth.

Outcome: Identify and assess system vulnerabilities and apply basic security management practices.

Activities:

- Lab activity: Use of vulnerability scanning tools (e.g., Nessus, OpenVAS).
- Case study analysis on system breaches.
- Risk assessment report for a sample IT infrastructure.

Evaluation Methods:

- Practical lab test with hands-on vulnerability identification.

- Report evaluation based on thoroughness and recommendations.
- Oral questioning/viva to assess understanding of system vulnerabilities.

Outcome: Implement data protection, secure system design, and apply cybersecurity tools and technologies.

Activities:

- Demonstration and use of encryption techniques (e.g., AES, RSA).
- Firewall and antivirus configuration in the lab.
- Mini project: Design a secure login system or simulate a secure network.

Evaluation Methods:

- Practical exam on encryption and tool usage.
- Project evaluation (design, functionality, security coverage).
- Peer review and demonstration of secure system designs.

Outcome: Analyze the impact of emerging technologies such as AI, blockchain, and quantum computing in the cybersecurity domain.

Activities:

- Debate or panel discussion on AI in Cyber Defense.
- Research paper/presentation on Blockchain for data integrity.
- Seminar on threats posed by quantum computing to encryption.

Evaluation Methods:

- Rubric-based assessment of research presentation.
- Critical analysis writing assignment.
- Participation and insight evaluation during debates/seminars.

SEMESTER-VI

COURSE 5: CYBER SECURITY

Practical

Credits: 1

2 hrs/week

Practicals

1. Use command-line tools to find host/domain name and IP address
2. Check assigned IP address of Ethernet adapter using appropriate command
3. Verify network connectivity using ping or similar command
4. Use netstat to find current network connections
5. Find all IP addresses of a domain using DNS lookup (nslookup, dig)
6. View and manipulate routing tables using route or ip route
7. Download a website for offline viewing using HTTRACK
8. Monitor live network traffic using tcpdump or wireshark
9. Identify packets containing HTTP GET requests using Wireshark
10. Analyze number of HTTP requests and time gaps between them in Wireshark
11. Demonstrate SQL Injection in a controlled environment
12. Explore OSSEC: list components, agents, and key features
13. Explain the role of HIDS and LIDS in intrusion detection
14. Analyze mail/web logs using pflogsumm, Webalizer, and AWStats
15. List firewall rules using iptables and monitor packet filtering

SEMESTER-VI

COURSE 6: DATA VISUALIZATION USING TABLEAU & POWER BI

Theory

Credits: 3

3 hrs/week

Course Objectives:

1. **Understand the fundamentals of** Tableau's interface, data connectivity, and basic visualization options.
2. **Use appropriate** chart types and interactive filters to represent univariate and multivariate data.
3. **Design** interactive dashboards and geographic maps using Tableau features and custom settings.
4. **Analyze and transform** raw data from multiple sources using Power BI's data tools.
5. **Evaluate and construct** optimized data models using relationships, hierarchies, and DAX calculations.

Course Outcomes

At the end of the Course, the students will be able to

1. create a new workbook, connect to various data sources, and apply formatting and dual-axis charts using Tableau's visualization design tools.
2. construct bar charts, scatter plots, box plots, and apply filters, quick filters, and parameters to allow viewer interaction with data visualizations.
3. build dashboards integrating maps, charts, and interactive elements such as filter actions and URL links, customized to specific analytical requirements.
4. import, transform, and clean data using Group By, Unpivot, Merge, and Append queries and evaluate the readiness of data for visualization.
5. assess data modeling approaches, build star schemas, create calculated measures using DAX (like YTD, QTD), and generate insights through drilldown and drill-through reports.

Unit 1. Introduction to Tableau:

What is Tableau? - Opening Existing Workbooks - Creating New Workbooks.

Basic Visualization Design: Choosing Color Options - Setting Mark Size - Choosing Shapes - Text Tables and Mark Labels - Formatting Options - Evaluating Multiple Measures - Shared Axis Charts - Measure Names and Measure Values - Dual Axis Charts.

Connecting to Data: Connecting to Various Data Sources - The Data Source Page - Customizing Your View of the Data: Changing Data Type - Modifying Dimension / Measure Assignment - Hiding -Renaming and Combining Fields - Splitting Fields - Saving and Sharing Metadata Extracting Data -Data Blending - Moving from Test to Production Database.

Unit 2. Chart Types & Viewer

Top 10 Chart Types (Uni-variate/Bi-Variate& Multi-variate Charts): Bar Chart - Line/Area Chart - Pie Chart - Text Table / Crosstab - Scatter Plot - Bubble Chart - Bullet Graph - Box Plot - Tree Map - Word Cloud.

Interacting with the Viewer: Filtering Data - Basic Filtering -Quick Filters - Parameters - Creating a Parameter - Displaying a Parameter - Using a Parameter in a Worksheet - Filter Actions - Highlight Actions - URL Actions.

Unit 3. Tableau Maps & Dashboards

Tableau Maps: Geographic Hierarchies and Ambiguity - Custom Geocoding - Background Maps and Layers - Navigating Maps and Selecting Marks - Map Options - Web Map Services - Mapping and Mark Types - Custom Background Images - Generating Your Own Coordinate System

Creating Dashboards and Stories: Creating a Simple Dashboard - Setting Dashboard - Size - Adding Sheets - Layout Container - Blank Text - Image - Webpage - Setting Dashboard and Element - Sizes - Dashboard Actions - Highlight Action - Filter Action - URL Action.

Unit 4. Introduction to Power BI

Power Bi: Getting Data from Existing Systems - Data Sources of Power Bi - Natural Language Queries - Getting data from web - Importing Data from Northwind ODATA feed T3_IMF - Functions & list Dates in Power Bi - Group By and unpivot in Power Bi - Merging Queries in Power Bi- Append Query in Power Bi - Charts in Power Bi

Unit 5. Data Modelling in Power BI

Power Pivot Model: Explain what a Data Model is, Create Relationships between Tables in the Model, Create and use a Star Schema Data Modeling in Power Bi

Power BI Environment: Adding Calculations and Measures - User Graphs, Dash boards - Dashboard for Corona Cases analysis.

Hierarchies with Drilldown - Hierarchies with Drill Through into Power BI - Create Reports using Aggregation Functions and calculate a (scalar) value such as Count, Sum, Average, Create Reports Using MTD QTD YTD I - Create Reports Using filter functions in DAX.

Textbooks:

1. Tableau 9 - The Official Guide, George Peck, McGraw Hill, 2016
2. Beginning Power BI: A Practical Guide to Self Service Data Analytics with Excel 2016 and Power BI Desktop, Dan Clark, O'Reilley, Second Edition

Reference Books:

1. Tableau Data Visualization Cookbook, Ashutosh Nandeshwar, Packt Publishing Ltd, 2013
2. The Excel User's Guide to DAX Power Query, Power BI & Power Pivot in Excel 2010-2016, Rob Collie &Avi Singh, Holy Macro! Books, 2016

3. Tableau Your Data! Fast and Easy Visual Analysis with Tableau Software, second Edition, Daniel G. Murray, John Wiley & Sons

Activities:

CO1: Create a new workbook, connect to various data sources, and apply formatting and dual-axis charts using Tableau's visualization design tools.

Activities:

- Guided tutorial on connecting to sample datasets.
- Group discussion: Dimensions vs. Measures.
- Hands-on task: Clean and prepare data using Tableau tools (split, rename, blend).

Evaluation Methods:

- Quiz on Tableau terminology and interface.
- Practical test: Import a dataset and perform preparation steps.
- Viva or oral review of the interface and steps taken.

CO2: Construct bar charts, scatter plots, box plots, and apply filters, quick filters, and parameters to allow viewer interaction with data visualizations.

Activities:

- Create a set of charts using a given dataset.
- Lab activity: Add parameters and filter actions to make visuals interactive.
- Peer demo: Walkthrough of created interactive visuals.

Evaluation Methods:

- Lab assessment: Submission of a Tableau workbook with charts and interactivity.
- Observation checklist during practical sessions.
- Rubric-based evaluation of visual quality and interactivity.

CO3: Build dashboards integrating maps, charts, and interactive elements such as filter actions and URL links, customized to specific analytical requirements.

Activities:

- Project: Create a dashboard with at least one map, chart, and action.
- Group assignment: Compare dashboards created by peers.
- Use case study: Build a dashboard from a real-world dataset (e.g., sales by region).

Evaluation Methods:

- Final dashboard project submission with rubric for creativity, accuracy, and interactivity.
- Peer evaluation using feedback forms.
- Instructor assessment of dashboard structure and functionality.

CO4: Import, transform, and clean data using Group By, Unpivot, Merge, and Append queries and evaluate the readiness of data for visualization.

Activities:

- Guided Power BI lab: Load and transform data using Power Query Editor.
- Individual worksheet: Identify transformation needs from messy data.
- Case analysis: Suggest cleaning steps for sample raw datasets.

Evaluation Methods:

- Practical test on using transformation tools (Group By, Merge, etc.).
- Submission of a cleaned dataset with documentation.
- Checklist-based evaluation of transformation accuracy.

CO5: Assess data modeling approaches, build star schemas, create calculated measures using DAX (like YTD, QTD), and generate insights through drilldown and drill-through reports.

Activities:

- Data modeling exercise: Create a star schema using sample sales data.
- DAX lab: Write measures for aggregations (MTD, YTD).
- Dashboard development with drillthrough functionality.

Evaluation Methods:

- Rubric-based evaluation of data model quality and relationships.
- Quiz on DAX functions and schema concepts.
- Demo presentation of final Power BI dashboard with drilldowns.

SEMESTER-VI

COURSE 6: DATA VISUALIZATION USING TABLEAU & POWER BI

Practical

Credits: 1

2 hrs/week

Practicals

1. Install and Explore Tableau
2. Blend Data from Multiple Sources in Tableau
3. Create Univariate Charts in Tableau
4. Create Dual Axis and Shared Axis Charts in Tableau
5. Create Bivariate Charts in Tableau
6. Create Multivariate Charts in Tableau
7. Create Word Cloud and Bubble Chart in Tableau
8. Build Interactive Dashboards in Tableau
9. Create Maps and Geographic Visualizations in Tableau
10. Import Data from Web and External Sources in Power BI
11. Use Natural Language Queries and Date Functions in Power BI
12. Transform Data with Group By, Unpivot, Merge, and Append in Power BI
13. Create Charts, Reports, and Dashboards in Power BI
14. Apply Advanced Reporting Features in Power BI
15. Demonstrate Hierarchies with Drill down.
16. Demonstrate Hierarchies with Drill through.
17. Generate reports using Aggregation functions to calculate a (scalar) value such as Count, Sum, Average, Minimum, Maximum for all rows in a column or table as defined by the expression.
18. Generate reports Using MTD QTD YTD in Power BI.
19. Generate reports Using filter functions in DAX.