SYLLABUS

Department	Department of Computer Science	**Program**	B.Sc. (Computer Science)
Level	UG	**Semester**	I
Course Code		**Course Type**	Major
Paper	I	**Course Title**	Computer System Architecture
Credit	6	**Session**	2026-2027

Course Outcomes

CO1. Understand the basic structure, operation and characteristics of digital computer.

CO2. Be able to design simple combinational digital circuits based on given parameters.

CO3. Familiarity with working of arithmetic and logic unit as well as the concept of pipelining.

CO4. Know about hierarchical memory system including cache memories and virtual memory.

CO5. Understand concept and advantages of parallel ism, threading, multi-processors and multi-core processors.

UNIT	Content	Duration
1	Fundamentals of Digital Electronics: Number System-Binary, Decimal, Octal, Hexa-Decimal, Conversions, Binary ArithmeticAddition, Subtraction, Multiplication, Division, Underflow, Overflow, Sign Magnitude, Complements-1's and 2's, Fixed-Point Representation, Floating-Point Representation.	12
2	Boolean Algebra, Reducing Boolean Expression, Logic Gates-AND, OR, NOT, Universal Gates-NAND, NOR, Analog and Digital Signals, Clock Waveform Timing, Map Simplification, K-Map- Two, Three and Four variables.	12
3	Combinational Circuits- Adder, Subtractor, Multiplexer, Demultiplexer, Decoders, Encoders. Binary Codes – Gray Codes, ASCII code, BCD code, EBCDIC, Error Detection Code and Correction Code, Hamming Code.	12
4	Sequential Circuits - Flip - Flops, SR, D, T, JK, Master-Slave, Registers, Shift Registers-SISO, SIPO, PISO, PIPO, Counters, Instruction, Instruction Format, Instruction Codes, instructions Cycles, Addressing Modes.	12
5	Handshaking, Concepts of RISC, CISC, DMA Data Transfer, Auxiliary Memory, Cache Memory, Associative Memory, Virtual Memory, Flynn's classification - Introduction to SISD, SIMD, MISD, MIMD, Parallelism, Multicore processors.	12

Assessment Scheme

Component	Credit	Internal	External	Total
Theory	4	30	70	100
Practical	2	30	70	100
Total	**6**			**200**

Practical List

HEAD
Dept. of Computer Science
St. Aloysius College (Auto)
JABALPUR

P1. Verification and interpretation of truth table for AND gate

P2. Verification and interpretation of truth table for OR gate

P3. Verification and interpretation of truth table for NOT gate

P4. Verification and interpretation of truth table for NAND gate

P5. Verification and interpretation of truth table for NOR gate

P6. To realize basic gate AND from Universal gate NAND.

P7. To realize basic gate OR from Universal gate NAND.

P8. To realize basic gate NOT from Universal gate NAND.

P9. To verify De-Morgan's First Law Theorem.

P10. To verify De-Morgan's Second Law Theorem.

Reference Books

- 1. William Stalling, "Computer Organization & Architecture", Pearson Education Asia.
- 2. V. Carl Hamacher , "Computer Organization", TMH
- 3. Tannenbaum, "Structured Computer Organization", PHI.
- 4. Er. Rajiv Chopra, "Computer Architecture", Revised 3rd Edition, S. Chand & Company Pvt. Ltd

HEAD
Dept. of Computer Science
St. Aloysius College (Auto)
JABALPUR

SYLLABUS

Department	Department of Computer Science	**Program**	B.Sc. (Computer Science)
Level	UG	**Semester**	I
Course Code		**Course Type**	Minor
Paper	II	**Course Title**	Computer Organization
Credit	4	**Session**	2026-2027

Course Outcomes

CO1. Understand (Level-2) the basic structure, operation and characteristics of digital computer.

CO2. Design (Level-6) simple combinational digital circuits based on given parameters.

CO3. Understand (Level-2) the working of arithmetic and logic unit as well as the concept of pipelining.

CO4. CO4. Summarize (Level-2) the hierarchical memory system including cache memories and virtual memory.

CO5. CO5 Identify (Level-2) the contributions of Indians in the field of computer architecture and related technologies.

UNIT	Content	Duration
1	Vedic Mathematics in computing: Vedic methods like Nikhilam Sutra and Urdhva-Tiryagbhyam, Fundamentals of Digital Electronics: Number System-Binary, Decimal, Octal, HexaDecimal, Conversions, Binary Arithmetic-Addition, Subtraction, Multiplication, Division, Underflow, Overflow, Sign Magnitude, Complements-1's and 2's, Fixed-Point Representation, FloatingPoint Representation.	8
2	Boolean Algebra, Reducing Boolean Expression, Logic Gates-AND, OR, NOT, Universal Gates-NAND, NOR, Analog and Digital Signals, Clock Waveform Timing, Map Simplification, K-Map- Two, Three and Four variables.	7
3	Combinational Circuits- Adder, Subtractor, Multiplexer, Demultiplexer, Decoders, Encoders. Binary Codes – Gray Codes, ASCII code, BCD code, EBCDIC, Error Detection Code and Correction Code, Hamming Code.	7
4	Sequential Circuits - Flip - Flops, SR, D, T, JK, Master-Slave, Registers, Shift Registers-SISO, SIPO, PISO, PIPO, Counters, Instruction, Instruction Format, Instruction Codes, instructions Cycles, Addressing Modes.	7
5	Handshaking, Concepts of RISC, CISC, DMA Data Transfer, Auxiliary Memory, Cache Memory, Associative Memory, Virtual Memory, Indian Contribution.	7

Assessment Scheme

Component	Credit	Internal	External	Total
Theory	3	30	70	100
Practical	1	30	70	100
Total	**4**			**200**

Practical List

HEAD
Dept. of Computer Science
St. Aloysius College (Auto)
JABALPUR

P1. To study basic gates (AND, OR, NOT) and verify their truth tables.

P2. To study and verify NAND as Universal gate using IC 7400.

P3. To realize basic gate AND from Universal gate NAND.

P4. 4. To realize basic gate OR from Universal gate NAND.

P5. 5. To realize basic gate N OT from Universal gate NAND.

P6. 6. To study and verify NOR as Universal gate

P7. 7. To realize basic gate AND from Universal gate NOR.

P8. 8. To realize basic gate OR from Universal gate NOR.

P9. 9. To realize basic gate NOT from Universal gate NOR.

P10. 10. To study Half Adder using basic gates and verify its truth table.

P11. 11. To study Full Adder using basic gates and verify its truth table.

P12. 12. To verify De-Morgan's First Law Theorem.

P13. 13. To verify De-Morgan's Second Law Theorem.

P14. 14. To design and construct RS flip Flop using gates and verifies the truth table.

P15. 15. To design and construct JK Flip Flop using gates and verifies the truth table.

Reference Books

- 1. "Computer Architecture and Organization", 3rd Edition by John P. Hayes, WCB /McGraw-Hill
- 2. "Computer Organization and Architecture: Designing for Performance", 10th Edition by William Stallings,
- Pearson Education.
- 3. "Computer System Design and Architecture", 2nd Edition by Vincent P. Heuring and Harry F. Jordan, Pearson
- Education.

HEAD
Dept. of Computer Science
St. Aloysius College (Auto)
JABALPUR

SYLLABUS

Department	Department of Computer Science	**Program**	B.Sc. (Computer Science)
Level	UG	**Semester**	III
Course Code		**Course Type**	Core Course
Paper	NA	**Course Title**	Object Oriented Programming Using Java
Credit	6	**Session**	2026-2027

Course Outcomes

CO1. Understand the fundamental concepts of object-oriented programming, Java architecture, data types, operators, arrays, strings and control statements required in Java programming.

CO2. Develop Java programs using classes, objects, constructors, inheritance, method overloading, method overriding and access modifiers to demonstrate object-oriented design concepts.

CO3. implement packages and interfaces in Java programs for modular, reusable and reliable software development.

UNIT 1	Content	Duration
1	Introduction to OOPs and Java History and evolution of Java.' Java and Internet, bytecode, JVM, ARE, Java buzzwords, Java editions: SE, EE and ME; Object-Oriented programming (OOP): paradigms, abstraction, Three OOP principles; Data Types, Variables, Type Conversion and Casting, Arrays and String; Operators: Arithmetic, bitwise, relational, Boolean logical, assignment, ternary operators and operator precedence; Control Statements.' Selection statements — If and switch, Iteration statements — while, do-while, for, for-each Version of the for loop and nested loops; Jump statements: break and continue; Scanner class for user input.	12
2	Class, Object and Inheritance Introducing Classes: Class fundamentals, declaring and creating objects, instance variables, assigning object reference variables, methods, constructors, use of this keyword, garbage collection, method overloading, argument passing, returning objects, recursion, access modifiers, static keyword, and command-line arguments; Encapsulation: data hiding and implementation using access control mechanisms; Inheritance: Inheritance basics, Super class and super keyword, method overriding, abstract classes, final keyword, and Object class; Polymorphism: compile-time and runtime polymorphism.	12
3	Packages and Interfaces Packages: defining a package, finding packages and CLASSPATH, Java naming conventions, access protection and importing packages; Interfaces: Defining Interface, implementing Interface, Multiple inheritance issues, static methods in Interface; Exception handling: Fundamentals, exception types, using try and catch, multiple catch clauses, nested try statements, throw, throws, finally, Java's built-in exceptions, user-defined exceptions.	12

HEAD
Dept. of Computer Science
St. Aloysius College (Auto)
JABALPUR

4	Multithreading and Collections Framework Multithreaded programming: Java thread model — thread priorities, synchronization, messaging, Main thread, creating threads using Thread Class and Runnable Interface, inter-thread communication, suspending, resuming and stopping threads; Wrapper classes, String handling classes; Collection framework: Collection Interface, List Interface, ArrayList, LinkedList, HashSet and HashMap classes. Lambda Expression Fundamentals, functional Interfaces, and basic applications of Lambda Expressions.	12
5	File Handling, AWT, and Graphics Programming Stream API Basics: Byte streams and character streams; File Handling — Reading and writing files; AWT classes: Window fundamentals — Component, Container, Panel, Window, Frame and Canvas; working with frame windows — Setting dimension, hiding and showing a window, setting window title, and closing a window; Graphics: Drawing lines, rectangles, ellipsis and circles, arcs, and polygons; demonstrating drawing methods, sizing graphics; working with color and fonts.	12

Assessment Scheme

Component	Credit	Internal	External	Total
Theory	4	30	70	100
Practical	2	30	70	100
Total	**6**			**200**

Practical List

P1. Write a program to demonstrate the use of different data types, variables, and type casting.

P2. Write a program to perform arithmetic, relational, logical, and bitwise operations using operators.

P3. Write a program to find the largest of three numbers using if-else and switch statements.

P4. Write a program to check if a number is odd or even.

P5. Write a program to check if a number is prime or not.

P6. Write a program to display tables from 2 to 10.

P7. Write a program to calculate the sum and average of array elements.

P8. Write a program to find the largest and smallest element in an array.

P9. Write a program to search an element in an array using linear search.

P10. Write a program to perform matrix addition using two-dimensional arrays.

P11. Write a program to store and display student marks using arrays and strings.

P12. Write a program to create a class and object for storing and displaying student information.

P13. Write a program demonstrating constructor overloading and use of the this keyword.

P14. Write programs demonstrating method overloading using different parameter lists. IS. Write a program to calculate the factorial of a number using recursion.

P15. Write a program to create a superclass Shape and subclasses Circle and Rectangle that override the area() method.

P16. Create a user-defined package college containing a class Student, and import this package into another program to display student details.

P17. Write a program to demonstrate different access modifiers within packages.

P18. Write a program to define an interface Shape and implement it in Circle and Rectangle classes to calculate area.

P19. Write a program to demonstrate multiple inheritance using interfaces.

HEAD
Dept. of Computer Science
St. Aloysius College (Auto)
JABALPUR

Reference Books

- Herbert Schildt and Danny Coward, Java: The Complete Reference, 13" Edition, McGraw-Hill Education,2024.
- Deitel P. and Deitel H., Java How to Program, 12" Edition, PearsonEducation, 2025.
- Kathy Sierra and Bert Bates, Head First Java, 3 Edition, O'Reilly Media, 2022.
- Cay S. Horstmann, Core Java, Volume I. Fundamentals, 12th Edition, Pearson Education, 2024.
- E. Balaguruswamy, Programming with Java, 7° Edition, McGraw Hill Education, 2023.
- Y. Daniel Liang, Introduction to Java Programming and Data Structures, 12* Edition, Pearson Education, 2019.
- Joshua Bloch, Effective Java, 3" Edition, Addison-Wesley Professional, 2018.

HEAD
Dept. of Computer Science
St. Aloysius College (Auto)
JABALPUR

SYLLABUS

Department	Department of Computer Science	**Program**	B.Sc. (Computer Science)
Level	UG	**Semester**	III
Course Code		**Course Type**	Minor
Paper	NA	**Course Title**	Introduction To Java Programming
Credit	4	**Session**	2026-2027

Course Outcomes

CO1. Install, configure, and setup the Java development environment for program development and execution.

CO2. Understand the fundamental concepts of Object-Oriented Programming, Java architecture, data types, operators, arrays, strings, and control statements required in Java programming.

CO3. Develop Java programs using classes, objects, constructors, inheritance, method overloading, method overriding, and access modifiers to demonstrate object oriented design concepts.

CO4. Implement packages and interfaces in Java programs for modular, reusable, and reliable software development.

CO5. Evaluate Java-based solutions for real-world computational problems using object-oriented design principles, and exception handling.

UNIT 1	Content	Duration
1	Introduction to Java and Programming Fundamentals : History and evolution of Java, features of Java, Java architecture: JVM, JRE, JDK, bytecode; Java editions: SE, EE and ME; installing and setting up Java environment; Structure of a Java program, tokens, keywords, identifiers, literals, comments, separators; Variables, constants, data types, type conversion and casting; Operators: Arithmetic, bitwise, relational, logical, assignment and ternary operators; expressions; Operator precedence; Input/output using Scanner and console methods.	12
2	Control Structures and Arrays : Decision-making statements: if, if-else, nested if and switch; looping statements: while, do-while, for, for-each version of the for loop and nested loops; Jump statements: break, continue and return; One-dimensional arrays, multidimensional arrays; command-line arguments; Introduction to Strings and String operations.	12
3	Classes, Object and Encapsulation : Object-Oriented programming concepts, Three OOP principles; Introducing Classes: Class fundamentals, declaring and creating objects, instance variables, assigning object reference variables, methods, constructors, method overloading, argument passing, returning objects, recursion, access modifiers, Encapsulation and data hiding, static keyword, and use of this keyword, garbage collection.	12

HEAD
Dept. of Computer Science
St. Aloysius College (Auto)
JABALPUR

4	Inheritance Polymorphism and Exception : Handling Inheritance - single, multilevel, hierarchical inheritance, Super class and super keyword, method overriding; Polymorphism: compile-time and runtime polymorphism, dynamic method dispatch; abstract classes and abstract methods, final keyword, and Object class; Exception handling: Fundamentals, exception types, using try, catch and finally, multiple catch clauses, nested try statements, throw, throws, Java's built-in exceptions, user-defined exceptions.	12
5	Packages and Interfaces : Packages - defining a package, finding packages and CLASSPATH, Java naming conventions, access protection and importing packages; Interfaces: concept and need of interfaces, defining Interface, implementing Interface, Multiple inheritance using interface, extending interface, default and static methods in interface, abstract class vs. interface; Introduction to functional interfaces and lambda basics.	12

Assessment Scheme

Component	Credit	Internal	External	Total
Theory	3	30	70	100
Practical	1	30	70	100
Total	**4**			**200**

Practical List

P1. Write a program to demonstrate the use of different data types, variables, and type casting.

P2. Write a program to perform arithmetic, relational, logical, and bitwise operations using operators.

P3. Write a program to find the largest of three numbers using if-else statements.

P4. Write a program to check if a number is odd or even.

P5. Write a program to check if a number is prime or not.

P6. Write a program to display tables from 2 to 10.

P7. Write a program to calculate the sum and average of array elements.

P8. Write a program to find the largest and smallest element in an array.

P9. Write a program to perform matrix addition and multiplication using two-dimensional arrays.

P10. Write a program to store and display student marks using arrays and strings.

P11. Write a program to check whether the number entered through command line argument is palindrome or not.

P12. Write a program to create a class and object for storing and displaying student information.

P13. Write a program demonstrating constructor overloading and use of this keyword.

P14. Write programs demonstrating method overloading using different parameter lists.

P15. Write a program to calculate factorial of a number using recursion.

P16. Write a program to create a superclass Shape and subclasses Circle and Rectangle that override the area() method.

P17. Write a program to create a superclass Employee and subclasses Manager and Developer that override the calculate Salary() method.

P18. Write a program with classes vehicle, Car, and Sports Car. Add one method in each class and show how sports Car inherits methods from both parent classes.

P19. Write a program to demonstrate different access modifiers within packages.

P20. Write a program to define an interface shape and implement it in Circle and Rectangle classes to calculate area.

HEAD
Dept. of Computer Science
St. Aloysius College (Auto)
JABALPUR

P21. Write a program to demonstrate multiple inheritance using interfaces.

P22. Write a program to handle arithmetic and array exceptions using try-catch, finally, throw, and throws keywords.

P23. Write a program to enter an age and create a user-defined exception called Invalid Age Error. If the age is less than 18, display the message: "Age must be 18 or above" otherwise, display: "Access granted".

Reference Books

- Deitel P. and Deitel H., Java: How to Program, 12th Edition, Pearson Education, 2025.
- Kathy Sierra and Bert Bates, Head First Java, 3rd Edition, O'Reilly Media, 2022.
- Cay S. Horstmann, Core Java, Volume I: Fundamentals, 12th Edition, Pearson Education, 2024.
- E. Balagurusamy, Programming with Java, 7th Edition, McGraw Hill Education, 2023.
- Y. Daniel Liang, Introduction to Java Programming and Data Structures, 12th Edition, Pearson Education, 2019.
- Joshua Bloch, Effective Java, 3rd Edition, Addison-Wesley Professional, 2018.

HEAD
Dept. of Computer Science
St. Aloysius College (Auto)
JABALPUR

SYLLABUS

Department	Department of Computer Science	Program	B.Sc. (Computer Science)
Level	UG	Semester	III
Course Code		Course Type	Discipline Specific Elective (DSE)
Paper	NA	Course Title	Discrete Mathematics and Combinatorics
Credit	4	Session	2026-2027

Course Outcomes

CO1. Understand {BT Level-2} the foundational concepts of discrete mathematics, logic, sets, and combinatorics including Indian Knowledge System contributions.

CO2. Apply {BT Level-3} mathematical logic, proof techniques, induction, recursion, and counting principles in solving computational problems.

CO3. Analyze {BT Level-4} relations, functions, recurrence relations, graphs, trees, and Boolean expressions for computer science applications.

CO4. Design {BT Level-6} combinatorial models and graphbased solutions for real-world problems related to networks, databases, cybersecurity, and AI systems.

CO5. Evaluate {BT Level-5} discrete mathematical structures and algorithms used in optimization, cryptography, machine learning, and social network analysis.

CO6. Develop {BT Level-6} logical reasoning and algorithmic thinking skills required for higher studies, competitive examinations, placements, and research.

UNIT	Content	Duration
1	Foundations of Discrete Mathematics and Indian Knowledge System Integration: Nature and scope of discrete mathematics, Propositions, logical connectives, truth tables, tautology, contradiction, logical equivalence,Indian Knowledge System (IKS) integration: Pingala's Chandas Shastra and binary representations, Meru Prastara and combinatorial arrangements, Panini's formal grammar and rule-based systems, Nyaya and Navya-Nyaya logical reasoning traditions, Recursive and algorithmic thinking in ancient Indian mathematics, Relevance of discrete structures in Computer Science. Practical Relevance and Applications: Digital logic design, Rule-based AI systems, Formal language processing, Compiler construction, Algorithmic reasoning and proofverification. Keywords: Logic, Proof Techniques, Predicate Logic, Binary Patterns, Pingala, Panini Grammar, Recursion, Induction, Nyaya Logic. Predicate logic and quantifiers, Methods of proof: direct proof, contradiction, contraposition, Mathematical induction and recursive reasoning, Introduction to combinatorics and finite structures.	9

HEAD
Dept. of Computer Science
St. Aloysius College (Auto)
JABALPUR

HOD

Generated: 2046

2	Sets, Relations, Functions, and Mathematical Foundations: Sets and set operations, Venn diagrams and power sets, Cartesian products, Relations: properties, equivalence relations, partial order relations, Hasse diagrams, Functions: injective, surjective, bijective functions, Composition and inverse of functions, Recursive definitions and recursive algorithms, Recurrence relations and their solutions. Introduction to generating functions Practical Relevance and Applications: Database normalization, Knowledge representation systems, Functional programming, Data mapping and transformations, Recursive algorithm development. Keywords: Sets, Relations, Functions, Equivalence Relations, Partial Orders, Recurrence Relations, Hasse Diagram, Recursion.	9
3	Counting Principles and Combinatorics: Fundamental principles of counting, Permutations and combinations, Pigeonhole principle, Inclusion-Exclusion principle, Binomial theorem and multinomial coefficients, Combinatorial identities, Discrete probability basics, Probability distributions in finite sample spaces, Applications of combinatorics in algorithm analysis, Counting techniques in coding theory and cryptography. Practical Relevance and Applications: Cryptographic key generation, Data science and probability models, Machine learning feature combinations, Scheduling and optimization, Competitive programming and algorithm design. Keywords: Permutations, Combinations, Pigeonhole Principle, InclusionExclusion, Probability, Binomial Theorem, Cryptography.	9
4	Graph Theory, Trees, and Boolean Algebra: Introduction to graphs and graph terminology, Types of graphs and graph representations, Degree sequence and graph properties, Paths, circuits, connectivity, Euler and Hamiltonian graphs. Trees and spanning trees, Binary trees and traversal techniques, Graph coloring and applications, Boolean algebra and Boolean functions, Canonical forms, Karnaugh maps, Logic gates and switching circuitsPractical Relevance and Applications: Network routing and communication systems, Social-network analysis, Web graph analysis, Digital circuit design, AI search trees and decision systems.Keywords: Graphs, Trees, Connectivity, Spanning Trees, Graph Coloring, Boolean Algebra, K-Map, Logic Gates.	9
5	dvanced Applications and Emerging Trends: Discrete structures in Artificial Intelligence and Machine Learning, Graph analytics in social networks and recommendation systems, Discrete optimization techniques, Applications in cybersecurity and cryptography, Finite state machines and automata applications, Discrete mathematics in blockchain and distributed systems, Mathematical modelling in Data Science, Introduction to coding theory and error detection. Emerging trends: Quantum-inspired discrete models, Network science, Computational combinatorics, Algorithmic game theory Practical Relevance and Applications: Al and ML algorithms, Cybersecurity systems, Blockchain verification, Recommendation engines, Smart network design. Keywords: Artificial Intelligence, Machine Learning, Network Science, Cryptography, Blockchain, Data Science, Optimization.	9

Assessment Scheme

Component	Credit	Internal	External	Total
Theory	4	30	70	100

Reference Books

- Ralph P. Grimaldi, Discrete and Combinatorial Mathematics
- C. L. Liu and D. P. Mohapatra, Elements of Discrete Mathematics
- J. P. Tremblay and R. Manohar, Discrete Mathematical Structures with Applications to Computer Science

HEAD
Dept. of Computer Science
St. Aloysius College (Auto)
JABALPUR

HOD

- Richard Johnsonbaugh, Discrete Mathematics
- Seymour Lipschutz, Schaum's Outline of Discrete Mathematics

HEAD
Dept. of Computer Science
St. Aloysius College (Auto)
JABALPUR

HOD

SYLLABUS

Department	Department of Computer Science	**Program**	B.Sc. (Computer Science)
Level	UG	**Semester**	III
Course Code		**Course Type**	Discipline Specific Elective (DSE)
Paper	I	**Course Title**	Software Engineering
Credit	4	**Session**	2026-2027

Course Outcomes

CO1. Learn, remember {BT Level-1} and understand {BT Level-2} the concepts of Software Engineering.

CO2. Understand {BT Level-2} the applications of software engineering processes and models.

CO3. Understand {BT Level-2} and Apply {BT Level-3} the analysis principles to software development.

CO4. Develop {BT Level-4} skills for analysis and design of software projects using structured and object-oriented approaches.

CO5. Apply {BT Level-3} - testing and quality assurance mechanisms to produce a secure and reliable software system.

CO6. Develop {BT Level-4} software systems using software engineering principles and methodologies.

UNIT	Content	Duration
1	Introduction to Indian Knowledge System (IKS): Meaning, Importance, and Scope, Contributions of Indian Scientists in Software Engineering, Ethical Values in Indian Traditions and Professional Ethics in Software Engineering. Green Software Engineering. Traditional Indian collaborative work models and community-driven project execution, overview of the Indian It industry's contributions to global software framework. Software: What is software, Types of software, Characteristics of Software, Attributes of good software, Introduction to Software Engineering, Program vs Software Product, Characteristics of Software, Software Crisis and Software Myths, Systems engineering & Software Engineering, Software Development Life Cycle (SDLC).	12
2	Software Development Process Models: What is a Software Process? What is a Software Process Model, Elements and Characteristics of Process Model, Process Classification, The Waterfall Model, Iterative Waterfall, Evolutionary Development, Spiral Development, RAD Model, Agile Software Development: Principles, Practices & Methods, Overview of CASE approach, Classification of CASE Tools?	12
3	Requirement Engineering Process, Requirement Gathering Techniques, Feasibility Study, Types of Feasibility, Functional Requirements, Non-functional Requirements, Software Requirement Specification (SRS), Characteristics of Good SRS, Structure of SRS Document, Requirement Validation, Requirement Management, Data Flow Diagram (DFD), Decision Tree, Decision Table, Entity Relationship Diagram (ERD), Use Case Modelling, Introduction to UML, Requirement Documentation Estimation and cost estimation: software estimation, size estimation, LOC estimation, COCOMO.	12

HEAD
Dept. of Computer Science
St. Aloysius College (Auto)
JABALPUR

HOD

4	Software Design Concepts, Design Principles, Concept Abstraction, Pattern, Cohesion and Coupling, Functional Independence, Structured Design, ObjectOriented Design using UML, Design Methodology, Modular Design, User Interface Design, Coding Standards, Structured Programming, Object-Oriented Programming Concepts, Class Diagram, Sequence Diagram, Activity Diagram, Component Design, Features of Modern GUI-Menus, Scrollbars, Buttons. Design Documentation.	12
5	Introduction to Software Testing, Verification and Validation, Levels of Testing, Unit Testing, Integration Testing, System Testing, Acceptance Testing, Black Box Testing, White Box Testing, Test Case Design, Quality Assurance, Activities, Debugging Techniques, Software Reliability, Software Maintenance, Types of Maintenance, Software Reengineering, Software Quality Assurance (SQA), ISO and CMM Standards, Software Reuse and Reverse Engineering.	12

Assessment Scheme

Component	Credit	Internal	External	Total
Theory	4	30	70	100

Reference Books

- Pressman, R.S. & Maxim, B.R. (2020). Software Engineering: A Practitioner's Approach (9th ed.) McGrawHill Education
- Software Engineering: Concepts and Practices - Ugrasen Suman, Cengage Learning.
- Fundamentals of Software Engineering - Rajiv Mall, Prentice Hall of India Private Limited
- An Integrated Approach of Software Engineering - Pankaj Jalote, Narosa Publishing House
- Software Project Management - Bob Hughes & Mike Cottrell, McGraw Hill Education
- Software Engineering: A practitioner's Approach - Roger Pressman, 6th edition. Hill International Edition
- Software Engineering - Sommerville,7Th edition, Pearson Education
- Software Engineering Principles and Practice - Waman S. Jawadekar, The McGraw-Hill companies
- Fundamentals of Object-Oriented Design Using UML, Meiler Page Jones: Pearson Education
- Sharma, A (2021), Indian IT Industry and Global Value Chains, Oxford University press.
- Fundamentals of Software Engineering, PHI.
- Software Engineering Concepts and Practice, Cengage Learning.
- Software Architecture: Foundations, Theory and Practice, Wiley.
- Software Engineering: An Engineering Approach, Wiley India.

HEAD
Dept. of Computer Science
St. Aloysius College (Auto)
JABALPUR

HOD

SYLLABUS

Department	Department of Computer Science	**Program**	B.Sc. (Computer Science)
Level	UG	**Semester**	V
Course Code		**Course Type**	Core Course
Paper	I	**Course Title**	Relational Database Management System
Credit	6	**Session**	2026-2027

Course Outcomes

CO1. Understand database concepts, applications, structure, need for a database and ER-

CO2. Model terminologies.

CO3. Be able to understand the fundamentals of Relational Algebra and relational calculus

CO4. To gain skills in creating a logical design of databases, including the E R method and

CO5. normalization approach.

CO6. Know about SQL functions and operators

CO7. Understand the knowledge of Database and transaction management.

UNIT 1	Content	Duration
1	Introduction: Advantages of the DBMS approach, various views of data, dataindependence, schema & sub-schema, Primary concepts of data models, Databaselanguages, transaction management, database administrator, & uses, data dictionary, andoverall system architecture. ER Model: Basic concepts, design issues, mapping constraintskeys, ER diagram, weak & strong entity sets, specialization & generalization, aggregation,inheritance, design of ER schema, reduction of ER schema to tables.	11
2	Domain Relation & Keys: Domains, relations, kinds of relation, relational databases, various types of keys, candidate, primary, alternate & foreign keys. Relation algebra & SQL: The structure, relation algebra with extended operations, modification of database, idea of relational calculus, basic structure of SQL, set operation, aggregate function.	11
3	Functional dependencies & Normalization: Base definition, trivial and nontriviadependencies, closure set of dependencies, & of attributes, irreducible set of dependenciesintroduction to normalization, non-loss decomposition, FD diagram, I,II & III NFdependencies prevention, BCNF, multivalued dependencies, preventions, BCNFMultivalued dependencies & 4NF, Join dependencies & 5NF.	11
4	Introduction to SQL, Data types, key constraints:- primary key, Candidate key, Integrity rules Entity integrity, Referential integrity rule. SQL Commands: - DDL, DML, DCL, TCL syntax and examples, select query with all the clauses. Like Predicate, Operator (Between, In, Not in)	11

HEAD
Dept. of Computer Science
St. Aloysius College (Auto)
JABALPUR

5	12 V Advanced SQL: - SQL join operations, Sub queries, indexes, sequences, and views SQL Functions. Introduction to PL/SQL:-PL/SQL structure, Cursors, Triggers, Stored Procedures and functions. Transaction Management-concurrency & recovery, ACID properties, transaction state, implementation of atomicity and durability, Storage structure in database - types, hashing.	11

Assessment Scheme

Component	Credit	Internal	External	Total
Theory	4	30	70	100
Practical	2	30	70	100
Total	**6**			**200**

Practical List

P1. Write an SQL query to fetch "FIRST_NAME" from the Worker table using the alias name as <WORKER_NAME>.

P2. Write an SQL query to fetch "FIRST_NAME" from the Worker table in upper case.

P3. Write an SQL query to fetch unique values of DEPARTMENT from the Worker table.

P4. Write an SQL query to print the first three characters of FIRST_NAME from the Worker table.

P5. Write an SQL query to find the position of the alphabet ('a') in the first name column 'Amitabh' from the Worker table.

P6. Write an SQL query to print the FIRST_NAME from the Worker table after removing white spaces from the right side.

P7. Write an SQL query to print the DEPARTMENT from the Worker table after removing white spaces from the left side.

P8. Write an SQL query that fetches the unique values of DEPARTMENT from the Worker table and prints its length.

P9. Write an SQL query to print the FIRST_NAME from the Worker table after replacing 'a' with 'A'.

P10. Write an SQL query to print the FIRST_NAME and LAST_NAME from the Worker table into a single column COMPLETE_NAME. A space char should separate them.

P11. Write an SQL query to print all Worker details from the Worker table order by FIRST_NAME Ascending.

P12. Write an SQL query to print all Worker details from the Worker table order by FIRST_NAME Ascending and DEPARTMENT Descending.

P13. Write an SQL query to print details for Workers with the first names as "Vipul" and "Satish" from the Worker table.

P14. Write an SQL query to print details of Workers with DEPARTMENT name as "Admin".

P15. Write an SQL query to print details of the Workers whose FIRST_NAME contains 'a'.

P16. Write an SQL query to print details of the Workers whose FIRST_NAME ends with 'a'.

P17. Write an SQL query to print details of the Workers whose FIRST_NAME ends with 'h' and contains six alphabets.

P18. Write an SQL query to print details of the Workers whose SALARY lies between 100000 and 500000.

P19. Write an SQL query to print details of the Workers who joined in Feb'2014.

P20. Write an SQL query to fetch the count of employees working in the department 'Admin'.

P21. Write an SQL query to fetch worker names with salaries >= 50000 and <= 100000.

P22. Write an SQL query to fetch the no. of workers for each department in descending order.

P23. Write an SQL query to print details of the Workers who are also Managers.

P24. Write an SQL query to fetch duplicate records having matching data in some fields of a table.

P25. Write an SQL query to show only odd rows from a table.

HEAD
Dept. of Computer Science
St. Aloysius College (Auto)
JABALPUR

HOD

P26. Write an SQL query to show only even rows from a table.

P27. Write an SQL query to clone a new table from another table.

P28. Write an SQL query to fetch intersecting records of two tables.

P29. Write an SQL query to show records from one table that another table does not have.

Reference Books

- Database Management system by Arun K. Majumdar & P. Bhattacharya, TMH Pub.
- Principles of Database System by Jeffrey O. Ullman, Galgotia Pub, Co. Ltd.
- Principles of Database Management System by James Martin, PHI

HEAD
Dept. of Computer Science
St. Aloysius College (Auto)
JABALPUR

SYLLABUS

Department	Department of Computer Science	**Program**	B.Sc. (Computer Science)
Level	UG	**Semester**	V
Course Code		**Course Type**	Elective
Paper	III	**Course Title**	Operating System
Credit	4	**Session**	2026-2027

Course Outcomes

CO1. Describe the importance of computer system resources and the role of
CO2. operating systems in their management policies and algorithms.
CO3. Specify the objectives of modern operating systems and describe how
CO4. operating systems have evolved over time
CO5. Understand various process management concepts and can compare various
CO6. scheduling techniques, synchronization, and deadlocks.
CO7. Describe the concepts of multithreading and memory management
CO8. techniques.
CO9. Identify the best-suited memory management techniques.

UNIT 1	Content	Duration
1	Introduction to Operating System: What is an Operating System? History and Evolution of OS, Basic OS functions, Resource Abstraction, 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.	12
2	Process Management: Process Concepts, Process States & Process Control Block. Process Scheduling: Scheduling Criteria, Scheduling Algorithms (Preemptive & Non-Preemptive) - FCFS, SJF, SRTN, RR, Priority, Multiple-Processor, Real-Time, Multilevel Queue and Multilevel Feedback Queue Scheduling. Deadlock - Definition, Deadlock Characterization, Necessary and Sufficient Conditions for Deadlock. Deadlock Handling Approaches: Prevention, Avoidance, Detection and Recovery.	12
3	Memory Management: Introduction, Address Binding, Logical versus Physical Address Space, Swapping, Contiguous & Non-Contiguous Allocation, Fragmentation (Internal & External), Compaction, Paging, Segmentation, Virtual Memory, Demand Paging, Performance of Demand Paging, Page Replacement Algorithms. File Management: Concept of File System (File Attributes, Operations, Types), Functions of File System, Types of File System, Access Methods (Sequential, Direct & other methods),	12

HEAD
Dept. of Computer Science
St. Aloysius College (Auto)
JABALPUR

HOD

4	Disk Management: Structure, Disk Scheduling Algorithms (FCFS, SSTF, SCAN, C- SCAN, LOOK),LINUX: Introduction, History and features of Linux, advantages, Linux architecture, the File system of LINUX - boot block, super block, inode table, data blocks. Linux standard directories, Linux kernel, Partitioning the hard drive for Linux, system - startup and shut-down process, init and run levels. Process, Swap, Partition, fdisk, checking disk free spaces. CLI OS vs GUI OS, Windows v/s Linux, Files and Directories. Concept of Open-Source Software.	12
5	Linux Administration: Types of user- Root and normal user, Multiple logins simultaneously (Ctr+ Alt + F1, F2,F6), who command. Help: what is, --help, man command Basic Commands: For displaying current directory, files and directories of current/absolute/relative location(s), creating,removing, renaming, copying and moving files or directories.Managing multiple processes:connecting processes with pipes, tee, redirecting input-output, changing process priority with nice,cron commands, kill, ps. Managing user accounts- Sudo, users: useradd, usermod, userdelpasswd. Group: Primary & Secondary Group, chgrp, chown, groupadd, groupdel. Permissions:adding and removing permissions.	12

Assessment Scheme

Component	Credit	Internal	External	Total
Theory	3	30	70	100
Practical	1	30	70	100
Total	**4**			**200**

Practical List

P1. Linux Directory Commands: pwd, mkdir, rm -rf, ls, cd, cd/, cd-

P2. Linux File Commands: touch, cat, cat >, cat >>, rm, cp, mv, rename

P3. Linux Permission Commands: su, id, useradd, passwd, groupadd, chmod, groupdel, chown, chgrp

P4. Linux File Content & Filter Commands: head, tail, tac, more, less, grep, cat, cut, grep, comm, sed, tee, tr, uniq, wc, od, sort, diff

P5. Linux Utility Commands: find, bc, locate, date, cal, sleep, time, df, mount, exit, clear, gzip, gunzip

P6. Vi editor: Create a file, edit, save and quit. Highlighting the searched term within a file, cut, yank, undo.

P7. Write the Linux command to copy a File to some other location.

P8. Write the Linux command to delete a file.

Reference Books

- G.Nutt, Operating Systems: A Modern Perspective, Pearson Education
- W. Stallings, Operating Systems, Internals & Design Principles, Pearson Education
- M.Milenkovic, Operating Systems – Concepts and Design, Tata McGraw Hill
- A Silberschatz , P.B Galvin , G. Gagne , Operating System Concepts, John Wiley
- Publications
- A.S Tanenbaum, Modern Operating System, Pearson Education
- J.L Peterson, Operating System Concepts Sumitabh Das, Linux, TMH

HEAD
Dept. of Computer Science
St. Aloysius College (Auto)
JABALPUR

HOD

SYLLABUS

Department	Department of Computer Science	**Program**	B.Sc. (Computer Science)
Level	UG	**Semester**	V
Course Code		**Course Type**	Elective
Paper	II	**Course Title**	Software Engineering
Credit	4	**Session**	2026-2027

Course Outcomes

CO1. Learn and understand the Concepts of Software Engineering

CO2. Learn and understand the Software Development Life Cycle

CO3. To Understand and apply the analysis principles to software development.

CO4. To apply the design & testing principles to software engineering.

Theory to supplement Practical
Software-What is software? Types of software, Characteristics of Software, Attributes of good software, Software Engineering-What is software engineering? Software engineering costs, what are the key challenges facing software engineering? Systems engineering & Software Engineering.
Software Development Process Models-What is a software process? What is a software process model? The waterfall model, Evolutionary development, Component-Based Software Engineering (CBSE). Process Iteration-Incremental delivery, Spiral development. Rapid software development- Agile methods, Extreme programming, Rapid application development (RAD), Software prototyping. Computer Aided Software Engineering (CASE)-Overview of CASE approach, Classification of CASE tools.
Software Requirement Analysis and Specification-System and software requirements, Types of software requirements- Functional and non-functional requirements, Domain requirements, User requirements. Elicitation and analysis of requirements- Overview of techniques, Viewpoints Interviewing, Scenarios, Use-cases, Process modeling with physical and logical DFDs, Entity Relationship Diagram, Data Dictionary, Requirement validation, Requirement specification, Software requirement Specification (SRS), Structure and contents of SRS, SRS format. Software Size Estimation and Cost Estimation-Software Estimation –Size Estimation, Function Point Analysis, LOC Estimation, COCOMO.
Software Design-Design Concepts-Abstraction, Architecture, Patterns, Modularity, Cohesion Coupling, Information hiding, Functional independence, Refinement. Design of input and Control, Design of User Interface design-Elements of good design, Design issues. Features of modern GUI - Menus, Scroll bars, windows, Buttons, icons, panels, error Messages, etc.
Good programming practices and Coding Standards. Software Testing and Quality AssuranceVerification and validation, Techniques of testing-Black-box and White-box testing, Levels of testing-Unit testing, Integration Testing, Interface testing, System testing, Alpha and beta testing, Regression testing. Design of test cases, Quality management activities, Product and process quality, Standards-ISO9000, Capability Maturity Model (CMM).

Assessment Scheme

Component	Credit	Internal	External	Total

HEAD
Dept. of Computer Science
St. Aloysius College (Auto)
JABALPUR

Theory	N/A	N/A	N/A	N/A
Practical	4	30	70	100
Total	**4**			**100**

Practical List

Not Available

Reference Books

- Software Engineering, A practitioner's Approach- Roger S. Pressman, 6th edition, Mc Graw
- Hill International Edition.
- An Integrated Approach to Software Engineering by Pankaj Jalote.
- Software Engineering- Sommerville, 7th edition, Pearson Education.
- The unified modeling language user guide Grady Booch, James Rambaugh, Ivar Jacobson,

HEAD
Dept. of Computer Science
St. Aloysius College (Auto)
JABALPUR