BANKURA UNIVERSITY

Curriculum and Credit Framework for

Computer Science

(Basic, Honours and Honours with Research)

With effect from the Academic Year 2023-2024

SEMESTER–V									
Sl. No.	**Course Code**	**Course Title**	**Credit**	**Marks**			**No. of Hours**		
				IA	**ESE**	**Total**	**L**	**T**	**P**
1	CSC/501/ MJC-09	Discrete Mathematics	4	10	40 (T:40, L:00)	50	4	0	0
2	CSC/502/ MJC-10	Computer Graphics	4	10	40 (T:25, L:15)	50	3	0	2
3	CSC/503/ MJC-11	Web Technologies	4	10	40 (T:25, L:15)	50	3	0	2
4	CSC/504/ MJC-12	Software Engineering	4	10	40 (T:40, L:00)	50	4	0	0
5	CSC/505/ MN-05	Discrete Mathematics	4	10	40 (T:40, L:00)	50	4	0	0
Total in Semester-V			**20**			**250**			

SEMESTER–VI									
Sl. No.	**Course Code**	**Course Title**	**Credit**	**Marks**			**No. of Hours**		
				IA	**ESE**	**Total**	**L**	**T**	**P**
1	CSC/601/ MJC-13	Design and Analysis of Algorithms	4	10	40 (T:25, L:15)	50	3	0	2
2	CSC/602/ MJC-14	Theory of Computations	4	10	40 (T:40, L:00)	50	4	0	0
3	CSC/603/ MJC-15	Information Security using Cryptography	4	10	40 (T:25, L:15)	50	3	0	2
4	CSC/604/ MJC-16	Artificial Intelligence	4	10	40 (T:25, T:15)	50	3	0	2
5	CSC/605/ MN-06	Design and Analysis of Algorithms	4	10	40 (T:25, L:15)	50	3	0	2
Total in Semester-VI			**20**			**250**			

COMPUTER SC (MJC-9, MN-5) **Discrete Mathematics**
Credit 04 (Theory = 04) **L/T/P: 4/0/0**

Course Learning Outcomes:

After successful completion of the course a student will be able to:

- Understand the basic mathematical problems in Computer Science and interdisciplinary areas
- Understand the basics of combinatory
- Understand some basic properties of graphs and related discrete structures
- Apply mathematical logic, mathematical proofs, and algorithmic thinking in problem solving.

THEORY

Unit-I: Introduction

- Sets - finite and Infinite sets, uncountably Infinite Sets;
- Functions, Relations, Properties of Binary Relations, Closure, Partial Ordering Relations;
- counting - Pigeonhole Principle,
- Permutation and Combination;
- Mathematical Induction,
- Principle of Inclusion and Exclusion.
- Rough Sets Versus Fuzzy Sets (in terms of basic properties only)

Unit-II: Growth of Functions

- Asymptotic Notations,
- Summation formulas and properties, Bounding Summations,
- Approximation by Integrals

Unit-III: Recurrences

- Recurrence Relations, generating functions, Linear Recurrence Relations with constant coefficients and their solution, Substitution Method,
- Recurrence Trees
- Master Theorem.

Unit-IV: Graph Theory

- Basic Terminology, Models and Types, multigraphs and weighted graphs,
- Graph Representation, Graph Isomorphism, Connectivity, Euler and Hamiltonian Paths and Circuits, Planar Graphs, Graph Coloring,
- Trees, Basic Terminology and properties of Trees, Introduction to Spanning Trees, Minimum Spanning Tree (MST) and its properties.

Unit-V: Propositional Logic

- Logical Connectives, Well-formed Formulas, Tautologies, Equivalences, Inference Theory.

Recommended Books:

1. C.L. Liu, D.P. Mahapatra, *Elements of Discrete mathematics*, 2nd Edition, Tata McGraw Hill, 1985.
2. Kenneth Rosen, *Discrete Mathematics and Its Applications*, Sixth Edition, McGraw Hill, 2006.
3. T.H. Coremen, C.E. Leiserson, R. L. Rivest, *Introduction to Algorithms*, 3rd edition, Prentice Hall on India, 2009.
4. M. O. Albertson and J. P. Hutchinson, *Discrete Mathematics with Algorithms*, John Wiley Publication, 1988.
5. J. L. Hein, *Discrete Structures, Logic, and Computability*, 3rd Edition, Jones and Bartlett Publishers, 2009.
6. D.J. Hunter, *Essentials of Discrete Mathematics*, Jones and Bartlett Publishers, 2008.

COMPUTER SCIENCE (MJC-10) **Computer Graphics**
Credit 04 (Theory = 03, Practical = 01) **L/T/P: 3/0/2**

Course Learning Outcomes:

After successful completion of the course a student will be able to:

- Gain insight of functioning of graphics hardware
- Understand different algorithms to generate 2D and 3D objects in graphics
- Create various graphics effects using computer

THEORY

Unit-I: Introduction

- Basic elements of Computer graphics, Applications of Computer Graphics.

Unit-II: Graphics Hardware

- Architecture of Raster and Random scan display devices, input/output devices.

Unit-III: Fundamental Techniques in Graphics

- Raster scan line, circle and ellipse drawing, thick primitives, Polygon filling, line and polygon
- clipping algorithms, 2D and 3D Geometric Transformations, 2D and 3D Viewing
- Transformations Projections- Parallel and Perspective, Vanishing points.

Unit-IV: Geometric Modeling

- Representing curves & Surfaces.

Unit-V: Visible Surface determination

- Hidden surface elimination.

Unit-VI: Surface rendering

- Illumination and shading models. Basic colour models and Computer Animation.

LABORATORY

(using C or Python)

1. Write a program to implement Bresenham's line drawing algorithm.
2. Write a program to implement mid-point circle drawing algorithm.
3. Write a program to clip a line using Cohen and Sutherland line clipping algorithm.
4. Write a program to clip a polygon using Sutherland Hodgeman algorithm.
5. Write a program to apply various 2D transformations on a 2D object (use homogeneous coordinates).
6. Write a program to apply various 3D transformations on a 3D object and then apply parallel and perspective projection on it.
7. Write a program to draw Hermite/Bezier curve.

Recommended Books:

1. J.D. Foley, A. Van Dam, Feiner, Hughes Computer Graphics Principles & Practice, 2nd edition, Addison Wesley, 1990.
2. D. Hearn, Baker, Computer Graphics, Prentice Hall of India, 2008.
3. D.F. Rogers, Procedural Elements for Computer Graphics, McGraw Hill, 1997.
4. D.F. Rogers, Adams, Mathematical Elements for Computer Graphics, 2nd edition, McGraw Hill, 1989.

COMPUTER SCIENCE (MJC-11) Web Technologies

Credit 04 (Theory = 03, Practical = 01) **L/T/P: 3/0/2**

Course Learning Outcomes:

After successful completion of the course a student will be able to:-

- Design different types of Client-side and Server-side applications.
- Design Web-enabled applications using JavaScript Programming, Java Server Pages, and Java Database Connectivity.
- Work with Java Beans.
- Understand and implement different applications like stand-alone applications, web applications etc.

THEORY

Unit-I: JavaScript

- Data types, operators, functions, control structures, events, and event handling.

Unit-II: JDBC

- JDBC Fundamentals, Establishing Connectivity and working with connection interface,
- Working with statements, Creating and Executing SQL Statements,
- Working with Result Set Objects.

Unit-III: JSP

- Anatomy of a JSP Page, JSP Processing, JSP Application Design with MVC,
- The Problem with Servlets, The Environment,
- Implicit JSP Objects, Conditional Processing, Displaying Values,
- Using an Expression to Set an Attribute, Declaring Variables and Methods, Error Handling and Debugging,
- Sharing Data Between JSP Pages, Requests, and Users, Database Access.

Unit-IV: Java Beans

- Java Beans Fundamentals, JAR files, Introspection, Developing a Simple Bean, Connecting to DB

LABORATORY

(JavaScript)

Create event-driven programs for the following:

1. Print a table of numbers from 5 to 15 and their squares and cubes using alert.
2. Print the largest of three numbers.
3. Find the factorial of a number n.
4. Enter a list of positive numbers terminated by zero. Find the sum and average of these numbers.
5. A person deposits Rs 1000 in a fixed account yielding 5% interest. Compute the amount in the account at the end of each year for n years.

Recommended Books:

1. Ivan Bayross, Web Enabled Commercial Application Development Using Html, Dhtml, JavaScript, Perl Cgi, BPB Publications, 2009.
2. Cay Horstmann, BIG Java, Wiley Publication, 3rd Edition, 2009.
3. Herbert Schildt, Java 7, The Complete Reference, 8th Edition, 2009.
4. Jim Keogh, The Complete Reference J2EE, TMH, 2002.
5. O'Reilly, Java Server Pages, Hans Bergsten, Third Edition, 2003.

COMPUTER SCIENCE (MJC-12) **Software Engineering**
Credit 04 (Theory = 04) **L/T/P: 4/0/0**

Course Learning Outcomes:

After successful completion of the Course a student will be able to:-

- Get the basic knowledge and understanding of analysis and design of complex systems
- Incorporate different attributes of software engineering to develop bug-free software in cost-effective manner
- Work as an active member or leader of software engineering teams.
- To manage time, processes, and resources effectively by prioritizing competing demands to achieve their goals.
- Identify and analyze the common threats in each domain

THEORY

Unit-I: Introduction

- The Evolving Role of Software, Software Characteristics, Changing Nature of Software
- Software Engineering: Program versus Software, Objective and Goal, understanding software as a Layered Technology,

Unit-II: Software Process Models

- Models: Frameworks and Properties
- Waterfall Model (classical and iterative)
- Incremental Process Model
- Spiral Model
- Prototype Model
- Hybridization and Model Selection

Unit-III: Requirement Analysis

- Software Requirement Analysis, Initiating Requirement Engineering Process, Requirement Analysis and Modeling Techniques, Flow Oriented Modeling,
- Need for SRS, Characteristics, and Components for SRS.

Unit-IV: Software Project Management

- Basic Concepts
- COCOMO Model

Unit-V: Risk Management

- Software Risks: types with examples, Risk Identification, Risk Projection and Risk Refinement, RMMM Plan.

Unit-VI: Quality Management

- Quality Concepts, Software Quality Assurance, Software Review, Metrics for Process and Project.

Unit-VII: Design Engineering

- Design Concepts, Architectural Design Elements, Software Architecture, Design at the Architectural Level and Component Level
- Mapping of Data Flow into Software Architecture,
- Modeling Component Level Design.

Unit-VIII: Capability Maturity Models

- CMM and CMMI

Unit-IX: Testing Strategies

- Software Testing Fundamentals,
- Strategic Approach to Software Testing,
- Test Strategies for Conventional Software: Validation Testing, System testing, Black-Box Testing, White-Box Testing and their type, Integrated Testing, Mutation Testing, Path testing: Linearly Independent Paths, Cyclomatic complexity (CC) and CC determination for a sample program.

Recommended Books:

1. R S Pressman, Software Engineering: A Practitioner's Approach (7th Edition), MGH, 2009
2. P. Jalote, An Integrated Approach to Software Engineering (2nd Edition), NPH, 2003
3. R. Mall, Fundamentals of Software Engineering (2nd Edition), PHI, 2004
4. I. Sommerville, Software Engineering (10^{th} Edition), Pearson Education, 2017

COMPUTER SC (MJC-13, MN-6) **Design and Analysis of Algorithms**
Credit 04 (Theory = 03 + Practical = 01) **L/T/P: 3/0/2**

Course Learning Outcomes:

After successful completion of the course, a student will be able to:-

- Gain engineering insight into problems
- Design algorithms as a mathematical process for problem-solving
- Design algorithms to solve different types of problems in computer science and interdisciplinary branches of science and engineering
- To learn how to analyze algorithms and estimate their worst-case and average-case performance (for simple to moderate problems)

THEORY

Unit-I: Introduction

- Basic Design and Analysis techniques of Algorithms, Correctness of Algorithm.

Unit-II: Algorithm Design Techniques

- Iterative techniques, Divide and Conquer, Dynamic Programming, Greedy Algorithms.

Unit-III: Sorting and Searching Technique

- **V**arious sorting techniques: Bubble Sort, Insertion Sort, Selection Sort, Merge Sort;
- Advanced Sorting techniques: Heap Sort, Quick Sort;
- Sorting in Linear Time: Bucket Sort, Radix Sort and Count Sort.
- Searching Techniques: Linear, Binary
- Median & Order Statistics,
- Complexity analysis.

Unit IV: Lower Bounding Techniques

- Decision Trees

Unit-V: Balanced Trees

- AVL Tree, Red-Black Trees

Unit-VI: Advanced Analysis Technique

- Amortized analysis

Unit-VII: Graphs

- Graph Algorithms: Breadth First Search, Depth First Search, Minimum Spanning Trees

Unit-VIII: String Processing

- String Matching, KMP Technique

Unit-IX: Computational Complexity

- P, NP, NP-complete, NP-hard

LABORATORY

(using C or Python)

1. **Sorting Algorithms Implementation:**
   - Implement Insertion Sort, Merge Sort, Heap Sort, Randomized Quick Sort, Radix Sort.
2. **Red-Black Tree Implementation:**
   - Create a Red-Black Tree and perform the following operations like Insert/deletion of a node; Search for a number & also report the colour of the node containing this number.
3. **Graph Algorithms Implementation:**
   - LCS (Longest Common Subsequence) detection for two given sequences; Breadth-First Search & Depth-First Search in a graph, Minimum spanning tree of a graph.
4. **Algorithm Performance Analysis:**
   - For the algorithms from Sl.No. 1 to 3, test run the algorithm on 100 different inputs of sizes varying from 30 to 1000.
   - Count the number of comparisons and draw the graph..

Recommended Books:

1. T H. Cormen, C E. Leiserson, Ronald L. Rivest, Clifford Stein, *Introduction to Algo*, PHI, 2E
2. S. Sahni & A.V. Aho, *Computer Algorithms: Introduction to Design and Analysis*, Pearson, 3rd Edition, 1999.

COMPUTER SC (MJC-14) Theory of Computations

Credit 04 (Theory = 04) **L/T/P: 4/0/0**

Course Learning Outcomes:

After successful completion of the course, a student will be able to:-

- Describe different elements of automata theory and formal languages.
- Design different mathematical models associated with computation theory.
- Solve different problems of machine automaton.
- Apply their understanding of key notions through complex problem-solving.

THEORY

Unit-I: Languages:

- Alphabets, String, Language, Basic Operations on Language, Concatenation, Kleene Plus and Kleene Star

Unit-II: Finite Automata and Regular Languages:

- Regular Expressions, Transition Graphs
- Finite Automata: without output (NFA, DFA), with output (Mealy, Moore)
- NFA to DFA Conversion
- NFA to Regular Expression Conversion: State Elimination Method
- Regular languages and their relationship with finite automata
- Pumping lemma and closure properties of regular languages

Unit-III: Context-Free Languages:

- Context-Free Grammars, Parse Trees, Ambiguities in Grammars and Languages
- Pushdown Automata (Deterministic and Non-Deterministic)
- Pumping Lemma, Properties of Context-Free Languages

Unit-IV: Turing Machines and Models of Computation:

- RAM, Turing Machine as a model of computation, Universal Turing Machine
- Language acceptability, Decidability, Halting Problem
- Recursively Enumerable and Recursive Languages, Un-decidable problems

---

Recommended Books:

1. Daniel I.A. Cohen, *Introduction to Computer Theory*, John Wiley, 1996.
2. Lewis & Papadimitriou, *Elements of the Theory of Computation*, PHI, 1997.
3. Hopcroft, Aho, Ullman, *Introduction to Automata Theory, Languages & Computation*, 3rd Edition, Pearson Education, 2006.
4. P. Linz, *An Introduction to Formal Language and Automata*, 4th Edition, Jones & Bartlett, 2006.

COMPUTER SC (MJC-15) Information Security using Cryptography

Credit 04 (Theory = 03, Practical = 01) L/T/P: 3/0/2

Course Learning Outcomes:

After successful completion of the course, a student will be able to:-

- Understand and describe various public key as well as secret key cryptographic algorithms.
- Implement different cryptographic algorithms in the laboratory.
- Learn about different cyber-security measures.

THEORY

Unit-I: Introduction

- Security Attacks, Cyber Criminals, Security Services, Security Mechanisms.

Unit-II: Cryptographic Techniques

- Classical Ciphers: Transposition Ciphers, Confusion, Diffusion, Symmetric, Asymmetric Cryptography.
- DES, Modes of DES, Uses of Encryption, Discrete Logarithm, Diffie-Hellman Key Exchange.
- RSA Algorithm, Hash Function, Digital Signatures, Digital Certificates.

Unit-III: Program Security

- Secure programs, Malicious program and code, Virus, Trap doors, Salami attacks.

Unit-IV: Threats in OS

- Memory and Address Protection, Access Control, File Protection, User Authentication

Unit-V: Database Security

- Requirements, Reliability, Integrity, Sensitive Data, Inference, Multilevel Security.

Unit-VI: Security in Networks

- Threats in Networks, Security Controls, Firewalls, Intrusion Detection Systems, Secure Emails.

LABORATORY

(using C or Python)

1. Demonstrate the use of Network tools: ping, ipconfig, ifconfig.
2. Write a program to perform encryption and decryption of Caesar cipher.
3. Write a program to perform encryption and decryption of Rail fence cipher.
4. Design and implement product ciphers using substitution and transposition ciphers.
5. Write a program to perform encryption and decryption of affine cipher.
6. Implement Diffie-Hellman Key exchange algorithm.
7. Implement RSA public key cryptosystem.
8. Demonstrate sending of a digitally signed document.
9. Demonstrate sending of a protected worksheet.
10. Demonstrate use of steganography tools.

Recommended Books:

1. C. P. Pfleeger, S. L. Pfleeger, *Security in Computing*, Prentice Hall of India, 2006.
2. B. A. Forouzan, *Introduction to Cryptography and Network Security*, McGraw Hill.
3. W. Stallings, *Network Security Essentials: Applications and Standards*, 4/E, 2010.

COMPUTER SC (MJC-16) Artificial Intelligence

Credit 04 (Theory = 03, Practical = 01) L/T/P: 3/0/2

Course Learning Outcomes:

After successful completion of the Course, a student will be able to:-

- Understand and describe AI-based problem-solving and searching algorithms.
- Gain insight of different knowledge representation techniques.
- Solve basic AI-problems using PROLOG programming.

THEORY

Unit-: Introduction

- Introduction to Artificial Intelligence, Background, and Applications.
- Turing test and Rational Agent approaches to AI.
- Introduction to Intelligent Agents, structure, behavior, and environment.

Unit-II: Problem Characteristics and Searching Techniques

- Problem characteristics and its variations.
- Production Systems, Control Strategies, Breadth-First Search, Depth-First Search.
- Hill Climbing and its variations.
- Heuristic Search Techniques: Best First Search, A*, Min-Max, and Alpha-Beta pruning, IDA*.
- Means-End Analysis, Introduction to Game Playing.

Unit-III: Knowledge Representation

- Conceptual Level Knowledge: Predicate Logic, Resolution Principle, Unification, Semantic Nets.
- Inferencing: Forward and Backward Chaining.
- Frames and Scripts, Production Rules, Conceptual Graphs.

Unit-IV: Dealing with Uncertainty and Inconsistencies

- Non-Monotonic Reasoning.
- Probabilistic Reasoning, Bayesian Probabilistic Inference.
- Fuzzy Logic.

Unit-V: Introduction to Natural Languages

- Parsing Techniques, Context-Free and Transformational Grammars, Recursive and Augmented Transition Nets.

Unit-VI: Programming Language: Logic (PROLOG)

LABORATORY

1. Write a Prolog program to calculate the sum of two numbers.
2. Write a Prolog program to find the maximum of two numbers.
3. Write a Prolog program to calculate the factorial of a given number.
4. Write a Prolog program to calculate the nth Fibonacci number.
5. Write a Prolog program to insert an element into a list.
6. Write a Prolog program to remove the Nth item from a list.
7. Write a Prolog program to remove all occurrences of an element from a list.
8. Write a Prolog program to implement append for two lists.
9. Write a Prolog program to implement palindrome checking .
10. Write a Prolog program to implement maxlist (List, Max) so that Max is the greatest number in the list of numbers.
11. Write a Prolog program to implement sumlist (List, Sum) so that Sum is the sum of a given list of numbers.
12. Write a Prolog program to implement two predicates evenlength (List) and oddlength (List) so that they are true if their argument is a list of even or odd length, respectively.
13. Write a Prolog program to compute GCD and LCM of two numbers
14. Write a Prolog program to implement semantic nets

RECOMMENDED BOOKS:

1. Dan W. Patterson, *Introduction to AI and Expert Systems*, PHI, 2007.
2. Rich & Knight, *Artificial Intelligence*, Tata McGraw Hill, 2nd Edition, 1991.