Dr. B. C. Roy Engineering College, Durgapur

Department of CSE(DS)

Field	Details
Course Name	Data Structure & Algorithms using C/C++
Course Code	CY-301
Semester	3
Course Category	Program Core Courses
Credits	3
Hours per Week	3L:0T:4P

1. Prerequisites

• Basic Programming Concepts (e.g., variables, loops, conditional statements)

• Discrete Mathematics (including basic set theory, logic, and graph theory)

• Familiarity with operating systems and computer architecture

2. Course Learning Objectives

• To equip students with a foundational understanding of data structures and algorithms essential for analyzing and mitigating cybersecurity threats.

• To develop students' proficiency in C programming, emphasizing secure coding practices and the prevention of common vulnerabilities.

• To enable students to apply their knowledge of data structures and algorithms to solve real-world cybersecurity problems, including network security and cryptography.

• To foster critical thinking and problem-solving skills through the analysis of security protocols, algorithms, and case studies of real-world security breaches.

• To provide students with a comprehensive overview of cryptographic principles and their application in securing systems and data.

3. Teaching Methodology

• Lectures and Presentations

• Interactive Discussions and Case Studies

• Lab Sessions

• Guest Lectures

4. Evaluation System

Activities	Class Test Full Marks	Assignment Full Marks	Attendance Full Marks	Total Marks
CIA-1	25	10	05	40
CIA-2	25	10	05	40
End Semester Examination (ESE)	-	-	-	60
Total				100 Marks

5. Course Modules

Module	Topics	Hours
1	**Introduction to Programming and Data Structures for Cybersecurity** - Programming in C: - Primitive data types - Structures (focus on representing security-relevant data like network packets) - Pointers and structures (memory management crucial for security) - Dynamic memory allocation (and its security implications, buffer overflows) - Arrays and Matrix operations (relevant for cryptography and network analysis) - Data Structures: - Definition - Arrays, Linked Lists (focus on their use in security applications) - Stacks and Queues (role in network protocols and system calls)	10

	- Abstract Data Types (ADT) and their importance in secure code design - Mathematical Notations: - Big O notation (performance analysis of security algorithms) - Time and Space Complexity (analyzing efficiency of security protocols)	
2	**Linear Data Structures and Algorithms in Cybersecurity** - List ADT: - Operations (Create, Insert, Search, Delete, Display) - Implementations (Array-based, Linked Lists – focus on secure implementations) - Applications (Network packet analysis, intrusion detection systems) - Stack ADT: - Operations (Create, Push, Pop, Top) - Implementations (Array-based, Linked) - Applications (Reverse Polish Notation in security tools, function call stack analysis for exploit detection) - Queue ADT: - Operations (Create, Enqueue, Dequeue) - Implementations (Array-based, Linked) - Applications (Network traffic management, buffer handling in secure systems)	8
3	**Trees, Hashing, and Cryptographic Applications** - Trees: - Introduction - Binary Search Trees (efficient searching in security databases) - Tree Traversals (relevant for vulnerability scanning and code analysis) - Heaps (priority queue for threat prioritization) - Hashing: - Introduction - Hash Functions (cryptographic hash functions, collision resistance) - Collision Avoidance techniques (importance in password storage and digital signatures) - Applications (password management, intrusion detection systems)	7
4	**Graph Algorithms and Network Security** - Graphs: - Introduction - Graph Representations (adjacency matrix,	6

	adjacency list) - Graph Traversal (Depth-First Search, Breadth-First Search – network scanning, vulnerability mapping) - Shortest Path Algorithms (Dijkstra's Algorithm – network routing, attack path analysis) - Introduction to Cryptography: - Symmetric and Asymmetric Encryption - Digital Signatures and Hashing Algorithms (SHA-256, RSA)	
5	**Algorithm Design and Analysis for Security** - Algorithm Design Techniques: - Greedy Method (resource allocation in security systems) - Divide and Conquer (efficient searching and sorting in large datasets) - Analysis of Algorithms: - Time and Space Complexity (analyzing the efficiency of security algorithms) - Sorting Algorithms (importance in data analysis for security) - Searching Algorithms: Linear Search, Binary Search (basic searching techniques in security contexts) - Introduction to common security algorithms (e.g., AES, DES)	6
6	**Security Protocols and System Design** - Introduction to common security protocols (TLS/SSL, IPsec) - Secure coding practices to prevent common vulnerabilities (buffer overflows, SQL injection) - Principles of secure system design - Case studies of real-world security breaches and their root causes	4

6. References

Textbooks:

1. Data Structures, Schaum's OutLines, Seymour Lipschutz, TATA McGRAW HILL Reference Book

2. Fundamentals of Data Structures in C, 2nd edition, Horowitz, Sahani, Anderson-Freed, Universities Press.

Reference Books:

1. Data Structures and Algorithm Analysis in C, Mark Allen Weiss, Pearson Education, 2nd Edition.

2. Reema Thareja, Data Structures Using C, 1st ed., Oxford Higher Education, 2011,

3. A.V.Aho, J.E Hopcroft , J.D.Ullman, Data structures and Algorithms, Pearson Education, 2003

7. Course Outcomes

ID	Statement	Action Verb	Knowledge Level
PCC-CS 301.1	Implement fundamental data structures (arrays, linked lists, stacks, queues, trees) using C, demonstrating proficiency in memory management and addressing potential security vulnerabilities like buffer overflows.	Implement	Apply
PCC-CS 301.2	Analyze the time and space complexity of algorithms used in cybersecurity applications (searching, sorting, graph traversal) using Big O notation.	Analyze	Analyze
PCC-CS 301.3	Apply hashing techniques and cryptographic hash functions to solve problems related to password management, digital signatures, and collision avoidance, explaining their importance in secure systems.	Apply	Apply
PCC-CS 301.4	Evaluate the effectiveness of various graph algorithms (DFS, BFS, Dijkstra's) in network security scenarios such as network scanning, vulnerability mapping, and attack path analysis.	Evaluate	Evaluate
PCC-CS 301.5	Design and compare different algorithm design techniques (greedy, divide and conquer) for solving security-related problems, considering their efficiency and applicability in real-world contexts.	Design	Create
PCC-CS 301.6	Critically assess common security protocols (TLS/SSL, IPsec) and	Critically assess	Evaluate

	secure coding practices, explaining how they mitigate vulnerabilities and applying this knowledge to analyze real-world security breaches and propose solutions.		

8. CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	1	-	3	1	-	1	1	1	1	1
CO2	3	3	1	2	1	-	-	-	1	2	1	1
CO3	3	2	2	1	2	1	-	1	1	2	1	1
CO4	3	3	2	3	2	1	-	1	1	2	1	1
CO5	3	2	3	2	2	1	-	1	2	2	2	1
CO6	3	2	2	2	1	3	1	3	1	3	1	1

9. CO-PSO Mapping

CO	PSO1	PSO2	PSO3
CO1	1	3	1
CO2	2	1	1
CO3	1	1	1
CO4	2	1	1
CO5	3	1	1
CO6	1	1	2

Dr. B. C. Roy Engineering College, Durgapur

Department of CSE(CS)

Field	Details
Course Name	Operating System
Course Code	CY-302
Semester	3
Course Category	Program Core Courses
Credits	3
Hours per Week	3L:0T:0P

1. Prerequisites

• Introduction to Computer Architecture and Organization

• Fundamentals of Programming (in C or a similar language)

• Discrete Mathematics (including logic and set theory)

2. Course Learning Objectives

• To provide students with a comprehensive understanding of operating system principles and their inherent security implications, enabling them to design and implement more secure systems.

• To equip students with the knowledge and skills necessary to analyze and mitigate security vulnerabilities related to process management, inter-process communication, memory management, and I/O operations within operating systems.

• To foster a deep understanding of the security challenges posed by various operating system architectures, including real-time and embedded systems, and to explore effective security design principles and best practices.

• To develop students' ability to critically evaluate the security aspects of different operating system components and functionalities, enabling them to make informed decisions regarding system security.

• To enhance students' practical skills in secure coding practices, vulnerability analysis, and the application of cryptographic techniques within the context of operating system security.

3. Teaching Methodology

• Lectures and Presentations

• Interactive Discussions and Case Studies

• Lab Sessions

• Guest Lectures

4. Evaluation System

Activities	Class Test Full Marks	Assignment Full Marks	Attendance Full Marks	Total Marks
CIA-1	25	10	05	40
CIA-2	25	10	05	40
End Semester Examination (ESE)	-	-	-	60
Total				100 Marks

5. Course Modules

Module	Topics	Hours
1	**Introduction to Operating Systems and Computer Security Fundamentals** - Operating system functions and security roles - Evolution of operating systems and security threats - Types of operating systems (focus on security implications of each: Batch, Interactive, Real-Time) - Computer system organization and security vulnerabilities - Introduction to system calls and their security	10

	relevance - User and operating system interface security (authentication, authorization) - Open-source operating systems and their security considerations - Basic system programming concepts relevant to security (e.g., input validation)	
2	**Process Management, Inter-process Communication, and Security** - Process concept and security implications (e.g., privilege escalation) - Process scheduling and its impact on security (e.g., real-time systems) - Operations on processes and security (e.g., process termination, signal handling) - Cooperating processes and inter-process communication (IPC) security challenges - Threads and multithreading security issues (race conditions, deadlocks) - Inter-process communication (IPC) mechanisms and their security vulnerabilities - Client-server systems and security (authentication, authorization, encryption) - Introduction to secure coding practices for multithreaded applications	9
3	**CPU Scheduling, Deadlock Handling, and Security** - Scheduling algorithms and their security implications (e.g., denial-of-service attacks) - Multiprocessor scheduling and security considerations - Real-time CPU scheduling and its role in securing time-sensitive systems - Deadlock characterization and its relation to security vulnerabilities - Deadlock prevention, avoidance, detection, and recovery strategies in secure systems - Secure design principles to mitigate deadlocks and security risks	6
4	**Memory Management, Virtualization, and Security** - Memory management techniques and their security implications (e.g., buffer overflows, memory leaks) - Virtual memory and its role in enhancing system security (e.g., address space layout randomization) - Page replacement algorithms and their security considerations	6

	- Introduction to memory protection mechanisms (e.g., segmentation, paging) - Virtualization technologies and their security aspects (hypervisors, virtual machine escape) - Secure virtualization techniques and best practices	
5	**I/O Management, File Systems, and Security** - I/O devices and their security vulnerabilities - Secure I/O operations (e.g., data encryption, access control) - Disk scheduling algorithms and their impact on security - File systems and their security considerations (access control lists, permissions) - File system security vulnerabilities (e.g., directory traversal, symbolic link attacks) - Secure file handling and data protection techniques - Introduction to cryptography and its application in securing file systems - Overview of common security threats related to I/O and file systems	6
6	**Real-Time Operating Systems, Operating System Security, and Case Studies** - Real-time operating systems (RTOS) and their security challenges - RTOS scheduling and security considerations - Security in embedded systems and IoT devices - Operating system security architecture and design principles - Case studies of operating system security breaches and vulnerabilities - Hands-on labs focusing on security aspects of OS concepts (e.g., secure coding practices, vulnerability analysis)	4

6. References

Textbooks:

1. Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, “Operating System Concepts”, John Wiley & Sons, Inc., 10th Edition, 2021.

2. William Stallings, “Operating Systems – Internals and Design Principles”, 9th Edition, Pearson, 2017

Reference Books:

1. Operating System - A Design Approach-Crowley, TMH

2. Modern Operating Systems, Andrew S. Tanenbaum 2nd edition, Pearson/PHI

7. Course Outcomes

ID	Statement	Action Verb	Knowledge Level
PCC-CS 302.1	Students will be able to define and explain the fundamental functions of operating systems and identify common security roles within an OS.	Define, Explain, Identify	Understand
PCC-CS 302.2	Students will be able to apply secure coding practices to prevent common vulnerabilities such as buffer overflows and race conditions in multithreaded applications, and will be able to analyze the security implications of different process scheduling algorithms.	Apply, Analyze	Apply
PCC-CS 302.3	Students will be able to compare and contrast different memory management techniques and evaluate their impact on system security, including the role of virtualization in enhancing security.	Compare, Contrast, Evaluate	Analyze
PCC-CS 302.4	Students will be able to design and implement secure I/O operations, including the application of cryptography to protect data at rest and in transit, and will be able to analyze the security vulnerabilities of common file system operations.	Design, Implement, Analyze	Apply
PCC-CS 302.5	Students will be able to evaluate the security challenges of real-time operating systems (RTOS) and embedded systems, including those in IoT devices, and will critically analyze case studies of operating system security breaches.	Evaluate, Analyze	Evaluate
PCC-CS 302.6	Students will be able to synthesize knowledge of	Synthesize, Propose, Justify	Create

	operating system security principles to propose and justify solutions for mitigating specific security vulnerabilities in a given scenario, potentially leveraging AI/ML techniques for threat detection or response where applicable.		

8. CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	1	1	1	1	1	1	1	1	1	1
CO2	3	3	2	2	2	1	1	1	2	1	1	1
CO3	3	3	2	2	2	1	1	1	2	1	1	1
CO4	3	2	3	2	2	1	1	1	2	1	1	1
CO5	3	3	2	3	2	2	2	1	2	2	1	1
CO6	3	3	3	3	2	2	1	2	3	3	2	1

9. CO-PSO Mapping

CO	PSO1	PSO2	PSO3
CO1	3	1	1
CO2	3	2	1
CO3	3	1	1
CO4	3	2	1
CO5	3	1	1
CO6	3	2	2

Dr. B. C. Roy Engineering College, Durgapur

Department of CSE(CS)

Field	Details
Course Name	Discrete Mathematics
Course Code	CY-303
Semester	3
Course Category	Basic Science Courses
Credits	3
Hours per Week	3L:0T:0P

1. Prerequisites

• Discrete Mathematics covering logic, set theory, proofs, and basic combinatorics

• Fundamental programming skills (e.g., Python or Java) and algorithmic thinking

• Elementary number theory and modular arithmetic (e.g., familiarity with gcd and the Euclidean algorithm)

2. Course Learning Objectives

• Enable students to develop rigorous logical reasoning and proof-writing skills that underpin the analysis of security policies, cryptographic protocols, and algorithm correctness.

• Equip learners with a solid foundation in algebraic structures, number-theoretic tools, and binary relations, preparing them to understand and design cryptographic primitives and access-control mechanisms.

• Foster the ability to model, analyze, and solve combinatorial and probabilistic problems relevant to security contexts such as password strength estimation, key-space sizing, and risk assessment.

• Cultivate competence in graph-theoretic concepts and algorithms--including traversals, spanning trees, matchings, and colorings--and their application to network security, routing, and resource allocation challenges.

• Promote the integration of discrete-mathematical techniques across topics, encouraging students to synthesize logical, algebraic, combinatorial, and graph-based methods when tackling complex problems in computer security and related domains.

3. Teaching Methodology

• Lectures and Presentations

• Interactive Discussions and Case Studies

• Lab Sessions

• Guest Lectures

4. Evaluation System

Activities	**Class Test Full Marks**	**Assignment Full Marks**	**Attendance Full Marks**	**Total Marks**
CIA-1	25	10	05	40
CIA-2	25	10	05	40
End Semester Examination (ESE)	-	-	-	60
Total				100 Marks

5. Course Modules

Module	**Topics**	**Hours**
1	**Mathematical Logic and Proof Foundations** - Statements, notation and basic set concepts - Logical connectives and truth tables - Well-formed formulas, tautology, contradiction, implication - Quantifiers (universal, existential) and predicate logic - Free vs. bound variables - Rules of inference and proof strategies (direct,	8

	contrapositive, induction) - Proof by contradiction and proof by cases - Consistency and soundness of logical systems - Basic applications to security policies and access-control reasoning	
2	**Algebraic Structures and Binary Relations** - Algebraic systems: examples and defining properties - Semigroups, monoids and groups (finite & infinite) - Modular arithmetic and basic number-theoretic tools (gcd, Euclidean algorithm) - Introduction to finite fields (GF(p)) - relevance to cryptography - Binary relations: definitions, properties, composition - Equivalence relations and partitioning - Partial orders and Hasse diagrams - Lattices (basic concepts)	7
3	**Combinatorics, Recurrences, and Discrete Probability** - Counting principles: sum rule, product rule - Permutations, combinations and binomial coefficients - Binomial theorem and basic multinomial expansion - Inclusion-exclusion principle - Pigeonhole principle and typical applications - Generating functions (intuitive use for counting) - Solving simple recurrence relations by substitution - Basic discrete probability theory (sample spaces, events) - Discrete random variables, expectation and variance - Simple probabilistic models used in security (e.g., password-guessing)	7
4	**Fundamentals of Graph Theory** - Graph terminology, basic definitions and examples - Matrix representations: incidence and adjacency matrices - Walks, paths, trails, circuits - Connectivity, cut vertices and bridges - Eulerian graphs - necessary and sufficient conditions - Hamiltonian graphs - necessary and sufficient conditions - Directed graphs (digraphs) and basic properties - Planarity basics and simple network-flow ideas relevant to security	6

5	**Trees and Bipartite Graph Algorithms** - Trees: definitions, characterizations and traversal orders (pre-, in-, post-order) - Cayley's formula and counting labelled trees - Spanning trees: BFS & DFS construction - Minimum-spanning-tree algorithms (Kruskal's and Prim's) and their security-network applications (STP) - Counting minimum spanning trees (Kirchhoff's theorem - intuitive view) - Bipartite graphs: properties and Hall's marriage theorem - Matching in bipartite graphs and the Chinese Postman problem - Practical algorithms for network routing and fault-tolerant design	7
6	**Graph Coloring, Independent Sets, and Matchings** - Vertex coloring: definitions, chromatic number, cliques - Greedy coloring algorithm and Brooks' theorem - Edge coloring basics - Independent sets and vertex covers - approximation ideas - Matchings: definitions, augmenting paths, König's theorem - Perfect matchings and Hall's condition (reviewed) - Approximation/greedy algorithms for maximum matchings - Applications to resource allocation, frequency assignment and security-policy partitioning	7

6. References

Textbooks:

1. Rosen, Kenneth H.: Discrete Mathematics and its Applications with Combinatorics and Graph Theory (7th Edition), TMH (Tata McGraw-Hill).

2. Mott, Joe L., Kandel, Abraham, and Baker, Teodore P.: Discrete Mathematics for Computer Scientists and Mathematicians, Pearson Education.

3. Johnsonbaugh, Richard: Discrete Mathematics, Pearson Education

4. Chandrasekaran, N., and Umaparvathi, M.: Discrete Mathematics, PHI Learning.

Reference Books:

1. Tremblay, J.P., and Manohar, R.: Discrete Mathematical Structures with Applications to Computer Science, TMH (Tata McGraw-Hill).

2. Mott, Joe L., Kandel, Abraham, and Baker, Teodore P.: Discrete Mathematics for Computer Scientists and Mathematicians (2nd Edition), Pearson Education.

3. Johnsonbaugh, Richard: Discrete Mathematics (7th Edition), Pearson Education

7. Course Outcomes

ID	Statement	Action Verb	Knowledge Level
CY-303.1	Recall and correctly state the definitions of logical connectives, truth tables, well-formed formulas, quantifiers, and basic set concepts.	Recall	Remember
CY-303.2	Explain how rules of inference and proof techniques (direct, contrapositive, contradiction, induction) are used to validate access-control policies, and construct a correct proof for a given policy statement.	Explain	Understand
CY-303.3	Apply modular arithmetic and properties of finite fields to implement basic cryptographic operations such as modular exponentiation and secret-sharing schemes.	Apply	Apply
CY-303.4	Analyze combinatorial counting methods and discrete-probability models to assess the likelihood of successful password-guessing attacks and compute the expected effort for specified password policies.	Analyze	Analyze
CY-303.5	Evaluate the suitability of graph algorithms (BFS, DFS, Kruskal's and Prim's MST) for designing resilient network topologies, and select and execute the appropriate algorithm to construct a secure spanning tree for a given network graph.	Evaluate	Evaluate
CY-303.6	Design and implement graph-coloring and matching algorithms to optimize resource allocation and	Design	Create

	enforce security-policy partitioning in complex networks, justifying algorithmic choices based on correctness, complexity, and security considerations.		

8. CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	1	1	1	1	1	1	-	1	-	1
CO2	3	3	2	2	1	2	1	1	1	2	1	2
CO3	3	2	3	2	3	1	1	1	1	2	2	2
CO4	3	3	2	3	2	2	1	1	1	2	1	2
CO5	3	3	3	2	3	2	1	1	1	2	2	2
CO6	3	3	3	3	3	3	1	2	2	3	2	2

9. CO-PSO Mapping

CO	PSO1	PSO2	PSO3
CO1	3	2	1
CO2	3	2	2
CO3	3	2	1
CO4	3	2	2
CO5	3	2	1
CO6	3	3	1

Dr. B. C. Roy Engineering College, Durgapur

Department of CSE(CS)

Field	Details
Course Name	Introduction to Object Oriented Programming
Course Code	CY-304
Semester	3
Course Category	Program Core Courses
Credits	3
Hours per Week	3L:0T:4P

1. Prerequisites

• Basic understanding of computer science principles

• Familiarity with a high-level programming language (e.g., Python, C++)

• Basic algebra and discrete mathematics concepts

2. Course Learning Objectives

• To equip students with a foundational understanding of Java programming and object-oriented principles, emphasizing secure coding practices throughout the development lifecycle.

• To enable students to design and implement secure Java applications, incorporating robust exception handling, input validation, and multithreading considerations.

• To foster students' ability to build secure and user-friendly graphical user interfaces (GUIs) that mitigate common web application vulnerabilities such as cross-site scripting and SQL injection.

• To provide students with a working knowledge of database connectivity using JDBC, focusing on secure data handling and prevention of database-related attacks.

• To introduce students to fundamental data structures, algorithms, and cryptographic concepts, enabling them to understand and apply basic security principles in software development.

3. Teaching Methodology

• Lectures and Presentations

• Interactive Discussions and Case Studies

• Lab Sessions

• Guest Lectures

4. Evaluation System

Activities	Class Test Full Marks	Assignment Full Marks	Attendance Full Marks	Total Marks
CIA-1	25	10	05	40
CIA-2	25	10	05	40
End Semester Examination (ESE)	-	-	-	60
Total				100 Marks

5. Course Modules

Module	Topics	Hours
1	**Introduction to Programming and Java Fundamentals for Cybersecurity** - Introduction to Programming Paradigms (Structured vs. Object-Oriented) - Need for OOP paradigm in secure software development - Java programming Environment and Runtime Environment - Java Virtual Machine (JVM) security considerations - Java program structure, Comments, and secure coding practices - Data types (primitive and reference) - Variables, scope, and lifetime of variables - Operators and operator precedence - Expressions and their evaluation - Control statements (selection, iteration, jump) and their secure use	10

	- Type conversion and casting - Arrays and array bounds checking - Simple Java program demonstrating secure input handling - Garbage Collection and its impact on memory security - Introduction to basic debugging techniques	
2	**Object-Oriented Programming in Java for Secure Systems** - Concepts of classes, objects, constructors, methods, and their application in secure design patterns - Access control and its role in information hiding and security - `this` keyword - Method overloading and constructors for flexible and secure code - Inheritance (specialization, extension) and its implications for secure code reuse - Polymorphism and its use in building flexible and secure systems - Exception handling and secure error management - Introduction to design patterns (Singleton, Factory) - Secure coding practices related to object-oriented programming - Understanding vulnerabilities related to inheritance and polymorphism	9
3	**Advanced Java Concepts and Secure Exception Handling** - Packages and namespaces for organizing and securing code - Interfaces and their use in designing secure APIs - Exception Handling: Concepts of exception handling, Benefits of exception handling, Exception hierarchy, Usage of `try`, `catch`, `throw`, `throws`, and `finally` for secure error handling. Handling security exceptions. - Input/Output - Secure file handling, input validation and sanitization to prevent injection attacks - Multithreading and concurrency issues in secure applications (race conditions, deadlocks) - Generics and their use in writing type-safe and secure code - Secure coding practices related to I/O and multithreading	6
4	**GUI Programming and Secure User Interfaces**	6

	- Event Handling: Secure event handling to prevent cross-site scripting (XSS) and other attacks - Swing: Introduction, Components, Containers, and secure UI design principles - Secure input validation and sanitization in GUI applications - Preventing SQL injection and other database vulnerabilities in GUI applications - Designing secure user interfaces to protect against phishing and other social engineering attacks	
5	**Database Connectivity and Security** - JDBC (Java Database Connectivity): JDBC overview, Secure database interactions, Preventing SQL injection vulnerabilities - Secure data storage and retrieval techniques - Introduction to database security best practices - Implementing secure authentication and authorization mechanisms for database access - Data encryption and decryption techniques for database security	5
6	**Data Structures, Algorithms, and Cryptography Basics** - Basic data structures (arrays, linked lists, stacks, queues) and their applications in security algorithms - Introduction to common cryptographic algorithms (symmetric and asymmetric encryption, hashing) - Basic understanding of digital signatures and certificates - Introduction to secure coding practices for cryptographic operations - Understanding common vulnerabilities related to cryptography	5

6. References

Textbooks:

1. Herbert Schildt, Java: The Complete Reference, 8/e, Tata McGraw Hill, 2011.

2. Kathy Sierra and Bert Bates, Head First Java, O'Reilly Media.

Reference Books:

1. Nageswararao R., Core Java: An Integrated Approach, Dreamtech Press, 2008.

2. Y. Daniel Liang, Introduction to Java Programming, 7/e, Pearson, 2013.

7. Course Outcomes

ID	Statement	Action Verb	Knowledge Level
PCC-CS 304.1	Students will be able to write basic Java programs that demonstrate secure input handling, including proper data type usage, array bounds checking, and garbage collection awareness.	Write	Apply
PCC-CS 304.2	Students will be able to design and implement secure object-oriented Java applications using inheritance, polymorphism, and appropriate design patterns (e.g., Singleton, Factory), explaining the security implications of each.	Design, Implement, Explain	Apply, Understand
PCC-CS 304.3	Students will be able to analyze and handle exceptions in Java applications, including secure error management and the prevention of vulnerabilities related to I/O and multithreading, using `try`, `catch`, `throw`, `throws`, and `finally` blocks.	Analyze, Handle	Analyze, Apply
PCC-CS 304.4	Students will be able to develop secure graphical user interfaces (GUIs) using Swing, incorporating secure input validation and sanitization techniques to prevent attacks such as cross-site scripting (XSS) and SQL injection.	Develop	Apply
PCC-CS 304.5	Students will be able to evaluate and implement secure database connectivity using JDBC, including techniques for preventing SQL injection and implementing secure authentication and authorization mechanisms, and explain the importance of data encryption and decryption.	Evaluate, Implement, Explain	Analyze, Apply, Understand

PCC-CS 304.6	Students will be able to compare and contrast common cryptographic algorithms (symmetric and asymmetric encryption, hashing), and analyze the security implications of their application in the context of common vulnerabilities, demonstrating an understanding of digital signatures and certificates.	Compare, Contrast, Analyze	Analyze, Understand

8. CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	3	1	3	1	1	1	1	1	1	1
CO2	3	2	3	1	3	1	1	1	2	1	1	1
CO3	3	3	2	2	3	1	1	1	1	1	1	1
CO4	3	2	3	1	3	2	1	1	2	1	1	1
CO5	3	2	3	1	3	2	1	1	1	1	1	1
CO6	2	3	1	2	1	1	1	1	1	2	1	1

9. CO-PSO Mapping

CO	PSO1	PSO2	PSO3
CO1	3	2	1
CO2	3	2	1
CO3	3	2	1
CO4	3	2	1
CO5	3	2	1
CO6	3	1	1

Dr. B. C. Roy Engineering College, Durgapur

Department of CSE(CS)

Field	Details
Course Name	Introduction to Cyber Security Essentials
Course Code	CY-305
Semester	3
Course Category	Program Core Courses
Credits	3
Hours per Week	3L:0T:4P

1. Prerequisites

• Basic Networking Concepts

• Understanding of Operating Systems

• Fundamental Computer Literacy

2. Course Learning Objectives

• To provide students with a comprehensive understanding of the fundamental concepts of cyberspace, cybersecurity threats, and the legal and ethical considerations governing the digital landscape.

• To equip students with the knowledge and skills necessary to analyze and mitigate cybersecurity risks within organizational and national contexts, including the implementation and evaluation of effective cybersecurity policies and strategies.

• To enable students to apply practical cybersecurity techniques, including network security principles, cryptography, and secure coding practices, to protect digital assets and infrastructure.

• To develop students' investigative skills in the area of cybercrime, including the ability to collect and analyze digital evidence, understand forensic methodologies, and apply relevant legal frameworks.

• To foster critical thinking and problem-solving abilities in students, enabling them to analyze real-world cybersecurity challenges, evaluate emerging trends, and contribute to the ongoing evolution of cybersecurity practices.

3. Teaching Methodology

• Lectures and Presentations

• Interactive Discussions and Case Studies

• Lab Sessions

• Guest Lectures

4. Evaluation System

Activities	**Class Test Full Marks**	**Assignment Full Marks**	**Attendance Full Marks**	**Total Marks**
CIA-1	25	10	05	40
CIA-2	25	10	05	40
End Semester Examination (ESE)	-	-	-	60
Total				100 Marks

5. Course Modules

Module	**Topics**	**Hours**
1	**Introduction to Cyberspace and Cybersecurity Fundamentals** - Defining Cyberspace and its components (Internet, WWW, Networks) - Basic network architecture (client-server, peer-to-peer) - Internet protocols (TCP/IP, HTTP, DNS) - The CIA triad (Confidentiality, Integrity, Availability)	10

	- Common cybersecurity threats (malware, phishing, denial-of-service) - Introduction to risk management (identification, assessment, mitigation) - Basic legal and ethical considerations in cyberspace - History of the internet and its impact on society - Overview of cybercrime and cybercriminals	
2	**Cybersecurity Policies, Governance, and Legal Frameworks** - Cybersecurity policies (organizational, national) - Importance of a strong cybersecurity policy - Key elements of a cybersecurity policy (access control, incident response, data protection) - Legal frameworks for cybersecurity (data protection laws, computer crime laws) - Governance frameworks (NIST Cybersecurity Framework, ISO 27001) - Role of international organizations in cybersecurity (e.g., Interpol) - Indian Cybersecurity Policy and relevant legislation (IT Act 2000 and amendments) - Ethical considerations in cybersecurity	7
3	**Network Security and Infrastructure** - Network security concepts (firewalls, intrusion detection/prevention systems) - Virtual Private Networks (VPNs) and their security implications - Secure network configurations (best practices for routers, switches, etc.) - Access control mechanisms (authentication, authorization, accounting) - Basic concepts of cryptography (symmetric vs. asymmetric encryption) - Introduction to Public Key Infrastructure (PKI) - Understanding common network vulnerabilities and attacks - Introduction to security information and event management (SIEM)	6
4	**Security Principles and Cryptography** - Software vulnerabilities (buffer overflows, SQL injection) - Secure coding practices - Common web application vulnerabilities (OWASP Top 10) - Introduction to risk assessment methodologies - Cybersecurity safeguards (access control, authentication, authorization)	7

	- Symmetric encryption algorithms (AES) - Asymmetric encryption algorithms (RSA) - Hashing algorithms (SHA-256, MD5) - Digital signatures and certificates - Introduction to password management best practices	
5	**Cybercrime Investigation and Forensics** - Classification of cybercrimes - Common cybercrime techniques (phishing, malware, ransomware) - Digital forensics principles and methodologies - Evidence collection and preservation - Chain of custody - Introduction to forensic tools (e.g., Autopsy, FTK Imager) - Investigating common cybercrimes (e.g., data breaches, denial-of-service attacks) - Reporting cybercrimes - Legal aspects of cybercrime investigations	6
6	**Cybersecurity Strategies, E-commerce Security, and Privacy** - Developing a cybersecurity strategy (risk assessment, incident response planning) - Cybersecurity awareness training - E-commerce security (payment gateways, secure transactions) - Social media security and privacy - Data privacy principles (GDPR, CCPA) - Data protection techniques (encryption, access control) - Introduction to ethical hacking and penetration testing - Current trends and challenges in cybersecurity - Case studies of real-world cyberattacks	6

6. References

Textbooks:

1. Yuri Diogenes, Erdal Ozkaya, Cyber security - Attack and Defense Strategies, Packt Publishers, 2018.

2. Charles J. Brooks, Christopher Grow, Philip A. Craig, Donald Short, Cybersecurity Essentials, Wiley Publisher, 2018.

3. B.B. Gupta, D.P. Agrawal, Haoxiang Wang, Computer and Cyber Security: Principles, Algorithm, Applications, and Perspectives, CRC Press, ISBN 9780815371335,2018.

4. Cyber Security Essentials, James Graham, Richard Howard and Ryan Otson, CRC Press. 2.R18 B.Tech. CSE (Cyber Security) III & IV Year

Reference Books:

1. William Stallings, Effective Cybersecurity: A Guide to Using Best Practices and Standards, 1st edition, 2019.

2. Nina Godbole, Sunit Belapure, Cyber Security - Understanding cybercrimes, Computer Forensics and Legal Perspectives, Wiley, 2011.

3. Patrick Engebretson, "The Basics of Hacking and Penetration Testing: Ethical Hacking and Penetration Testing Made easy", Elsevier, 2011

4. Anand Shinde, "Introduction to Cyber Security Guide to the World of Cyber Security", Notion Press, 2021

7. Course Outcomes

ID	Statement	Action Verb	Knowledge Level
PCC-CS 304.1	Define cyberspace and its key components, including the Internet, WWW, and networks; identify common cybersecurity threats such as malware, phishing, and denial-of-service attacks.	Define, Identify	Remember, Understand
PCC-CS 304.2	Explain the CIA triad (Confidentiality, Integrity, Availability) and apply basic risk management principles to identify and assess cybersecurity risks in given scenarios.	Explain, Apply	Understand, Apply
PCC-CS 304.3	Analyze the key elements of cybersecurity policies and governance frameworks (e.g., NIST Cybersecurity Framework, ISO 27001) and evaluate their effectiveness in mitigating specific cybersecurity threats.	Analyze, Evaluate	Analyze, Evaluate
PCC-CS 304.4	Design secure network configurations by applying knowledge of network security concepts (firewalls, intrusion detection/prevention systems,	Design, Apply	Apply, Create

	VPNs) and access control mechanisms (authentication, authorization, accounting).		
PCC-CS 304.5	Evaluate the effectiveness of different cryptographic techniques (symmetric and asymmetric encryption, hashing algorithms, digital signatures) and apply secure coding practices to mitigate common software vulnerabilities (e.g., buffer overflows, SQL injection).	Evaluate, Apply	Analyze, Apply
PCC-CS 304.6	Develop a comprehensive cybersecurity strategy for a given organization, incorporating risk assessment, incident response planning, and data privacy principles (GDPR, CCPA); critically analyze real-world cyberattacks and propose effective mitigation strategies leveraging AI/ML techniques where applicable (e.g., anomaly detection in network traffic).	Develop, Analyze, Propose	Create, Evaluate

8. CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	1	-	-	1	1	-	-	-	1	-	1
CO2	1	2	1	1	1	2	1	1	1	2	1	1
CO3	1	2	1	2	1	2	1	1	1	2	1	1
CO4	3	2	3	2	3	2	1	1	2	2	2	1
CO5	2	2	2	2	3	1	1	1	2	2	1	1
CO6	2	3	3	3	2	3	2	2	3	3	3	2

9. CO-PSO Mapping

CO	PSO1	PSO2	PSO3
CO1	1	1	1
CO2	1	1	1
CO3	1	1	1
CO4	1	2	1
CO5	1	2	1
CO6	3	3	2