Graphic Era HILL UNIVERSITY
Established by an Act of the State Legislature of Uttarakhand (Adhiniyam Sankhya 12 of 2011)
University under section 2(f) of UGC Act, 1956

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Course Name	:	**Object Oriented Programming**
Course Code	:	**DTCS 301**
Semester	:	**III**
Pre-requisite	:	**Basic knowledge of programming fundamentals, C programming, and logical problem-solving skills.**

L	T	P	Credits
3	0	0	3

Course Learning Objectives:

- Understand the features of object oriented programming and basic constructs of C++.
- Understand and design programs related to classes and objects.
- Comprehend and implement function and operator overloading and inheritance.
- Comprehend and design programs related to pointers, pointers to objects and functions.
- Understand and implement dynamic polymorphism and file handling.

Course Outcomes:

After completion of the course, the students will be able to:

Course Outcome	Statement	Bloom's Level	Skill
CO1	Understand fundamental concepts of object-oriented programming such as classes, objects, and encapsulation.	**L2 – Understand**	Understanding
CO2	Apply inheritance and polymorphism to design reusable program modules.	**L3 – Apply**	Programming, Implementation
CO3	Analyze the use of abstraction and interfaces in solving programming problems	**L4 – Analyze**	Programming, Implementation
CO4	Design and handle pointers for robust applications.	**L6 – Create**	Design and development skills
CO5	Develop object-oriented solutions for real-world computing problems	**L6 – Create**	Design and development skills

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Details of the Course:

Unit 1: Introduction **Contact Hours/Teaching Hours: 08**

Concept of Object Oriented Programming. History & features: It's need & requirement, procedure oriented programming versus object oriented programming, basic concepts object oriented programming, object oriented languages. Beginning with C++: Concepts & structure of C++ programming, concepts of structure.

Self-Learning Hours = 6

Students shall explore the history, features, and core concepts of object-oriented programming, understand the structure of C++ programs, and implement introductory C++ programs.

Team Activity Hours = 3

Group discussion on the advantages of object-oriented programming over procedure-oriented programming and identification of real-world applications of OOP concepts.

Unit 2: Classes and Objects

Contact Hours/Teaching Hours: 09

Syntax: Specifying a class, Defining member functions, Arrays within a class, Creating objects, memory allocation for objects, static data & member function, Arrays of objects, objects as function argument. Constructors and Destructors: Concept of Constructor (Default, Parameterized, copy), Overloaded Constructors, Constructor with default argument, Destructors.

Self-Learning Hours = 6

Students shall practice creating classes and objects, implement constructors and destructors, and develop C++ programs using member functions, arrays, and objects.

Team Activity Hours = 3

Group activity on designing classes and objects for real-world problems and comparing different types of constructors and object initialization techniques.

Unit 3: Overloading and Inheritance

Contact Hours/Teaching Hours: 08

Function overloading, Operator overloading (overloading unary & binary operators), rules for overloading operators. **Inheritance**: Concepts of inheritance, Derived classes, Member declaration (Protected), Types of inheritance (Single, multilevel, multiple, hierarchical, Hybrid inheritance), Virtual base classes, Abstract classes, Constructors in derived classes, Member classes.

Self-Learning Hours = 6

Students shall practice implementing function and operator overloading, analyze different types of inheritance, and develop C++ programs using derived classes and inheritance concepts.

Team Activity Hours = 3

Group discussion on selecting appropriate inheritance models and operator overloading techniques for different application scenarios.

Unit 4: Pointers in C++ **Contact Hours/Teaching Hours: 8**

Concepts of pointer (Pointer declaration, pointer operator, address operator, pointer expressions, and pointer arithmetic), Pointers & functions (Call by value, call by reference, pointer to functions, passing function to another function), Pointers in arrays (Searching, insertion & deletion), Pointers & objects (Pointers to objects, this pointer, and pointer to derived classes).

Self-Learning Hours = 6

Students shall practice pointer declaration and arithmetic, implement pointer-based functions and array operations, and develop C++ programs using pointers with objects.

Team Activity Hours = 3

Group activity on analyzing the use of pointers in functions, arrays, and objects, and comparing call-by-value and call-by-reference approaches.

Unit 5: Polymormphism & I/O System

Contact Hours/Teaching Hours: 09

Polymorphism: Concepts of polymorphism, types of polymorphism, Overloading & overriding, Virtual function, Static & dynamic binding. **Basic function of I/O system basics & File Processing**: Stream classes, using formatted& unformatted functions, using manipulator to format I/O, Basics of file system, opening & closing a file, reading & writing character from a file (get, put, getline, write), Command line arguments..

Self-Learning Hours = 6

Students shall practice implementing polymorphism through overloading, overriding, and virtual functions, and develop C++ programs for file handling and formatted input/output operations.

Team Activity Hours = 3

Group presentation on applying polymorphism in software design and analyzing file handling techniques and input/output operations in C++.

Course Articulation Matrix

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3
CO01	3	1	-	-	1	-	1	-	2	1
CO02	2	2	1	1	-	-	1	-	2	–
CO03	1	3	2	1	1	-	1	-	3	1
CO04	-	2	2	2	1	1	2	-	2	1
CO05	1	1	3	2	2	2	2	1	3	2
Avg.	1.75	1.8	2	1.5	1.25	1.5	1.4	1	2.4	1.25

Text Books:

1. B. Stroustrup, " C++ Programming Language", 4th Edition,Pearson Education.

Reference Books:

1. Herbert Schildt, "C++ The complete reference", 4 thEdition,Tata McGraw Hill.
2. Balgurusamy, "Object oriented programming with C++", 4 thEdition,Tata McGraw Hill.

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Online Resources:

1. **GeeksforGeeks – C++ Programming Language**
   https://www.geeksforgeeks.org/c-plus-plus/

2. **NPTEL – Programming in C++**
   https://nptel.ac.in/

3. **cplusplus.com – C++ Reference and Tutorials**
   https://cplusplus.com/

4. **Microsoft Learn – C++ Documentation**
   https://learn.microsoft.com/en-us/cpp/

Graphic Era HILL UNIVERSITY
Established by an Act of the State Legislature of Uttarakhand (Adhiniyam Sankhya 12 of 2011)
University under section 2(f) of UGC Act, 1956

Department of Computer Science and Engineering

Diploma Computer Science and Engineering

Course Name	:	**Digital Logic**
Course Code	:	**DTCS 302**
Semester	:	**III**
Pre-requisite	:	**Basic knowledge of Mathematics, Boolean concepts, and fundamental Electronic Devices & Circuits.**

L	T	P	Credits
3	0	0	3

Course Learning Objectives:

- To familiarize students with number systems, binary codes, and their conversions used in digital circuits.
- To develop the ability to apply Boolean algebra theorems and Karnaugh map techniques to minimize Boolean expressions.
- To enable students to analyze and design combinational logic circuits such as adders, subtractors, multiplexers, and decoder.
- To build competence in differentiating between latches and flip-flops and in designing counters using flip-flops.
- To introduce students to digital logic families, gate realization using MOS technology, and the working of various memory devices.

Course Outcomes:

After completion of the course, the students will be able to:

Course Outcome	Statement	Bloom's Level	Skill
CO1	Understand number systems and Boolean algebra used in digital circuits.	**L2 – Understand**	Understanding, Foundational Knowledge
CO2	Apply Karnaugh maps and minimization techniques to simplify logic expressions.	**L3 – Apply**	Problem Solving, Simplification
CO3	Analyze combinational logic circuits such as adders, multiplexers, and decoders.	**L4 – Analyze**	Analytical Thinking, Circuit Design
CO4	Evaluate the working of digital logic families and memory devices.	**L5 – Evaluate**	Evaluation, Technical Analysis
CO5	Design sequential circuits including flip-flops, counters, and registers.	**L6 – Create/Design**	Design, Implementation

---

Department of Computer Science and Engineering

Diploma Computer Science and Engineering

Details of the Course:

Unit 1: Introduction to Digital Techniques

Contact Hours/Teaching Hours: 8

Digital circuit, digital signal, use of digital circuit and digital signal, advantages and disadvantages of digital circuits, generation of digital signal, introduction to digital ICs, characteristics of digital ICs, logic families – comparison of TTL, CMOS and ECL logic families. Number System: Introduction to Binary, Octal, Decimal, Hexadecimal number systems, conversion of number systems, 1's complement and 2's complement, binary arithmetic (addition, subtraction), BCD code, BCD arithmetic (addition, subtraction).

Self-Learning Hours = 3

Students shall practice number system conversions, complement arithmetic, and BCD arithmetic problems, and prepare a comparative summary of TTL, CMOS and ECL logic families.

Team Activity Hours = 3

Group discussion on selecting an appropriate logic family (TTL/CMOS/ECL) for a given application, comparing power consumption, speed and cost trade-offs.

Unit 2: Logic Gates and Boolean Algebra

Contact Hours/Teaching Hours: 06

Logical symbol, logical expression and truth table of AND, OR, NOT, NAND, NOR, EX-OR and EX-NOR gates, universal gates – NAND and NOR gates, logical circuits of basic gates using universal gates, gates using more than two inputs, basic laws of Boolean algebra, duality theorem, De Morgan's theorems.

Self-Learning Hours = 4

Students shall practice realizing all basic gates using only NAND/NOR gates and solve Boolean expression simplification problems using the basic laws and De Morgan's theorems.

Team Activity Hours = 3

Group activity to implement a given logic function using only universal gates and verify the truth table using simulation software or a breadboard.

Unit 3: Combinational Logic Design / Circuits

Contact Hours/Teaching Hours: 10

Simplification of Boolean expressions using Boolean algebra, construction of logical circuits from Boolean expressions, Boolean expressions using Sum of Products and Product of Sums forms, K-map representation of logical functions, minimization of logical expressions using K-map (2, 3, 4 variables), standardization of SOP and POS equations. Concept of adders/subtractors, half adder and full adder, half subtractor and full subtractor, block diagram, truth table, logical expression and logic diagram of multiplexers (4:1 and 8:1), multiplexer IC. Block diagram and truth table of demultiplexer (1:4, 1:8, 1:16), demultiplexer IC.

Self-Learning Hours = 4

Students shall solve K-map minimization problems (2, 3 and 4 variables) and design half/full adder and subtractor circuits, verifying outputs against truth tables.

Team Activity Hours = 3

Group-based design activity to build a 4:1 multiplexer or 1:4 demultiplexer circuit and demonstrate its use in a simple data-routing application.

Unit 4: Flip-Flops and Sequential Logic Design

Contact Hours/Teaching Hours: 10

One-bit memory cell, clock signal, symbol and logic diagram using NAND gates, working and truth table of R-S flip-flop, symbol and logic diagram using NAND gates, working, truth table and timing diagram of clocked R-S flip-flop, triggering: edge triggering and level triggering, symbol and logic diagram using NAND gates, working, truth table and timing diagram of J-K flip-flop. Block diagram and truth table of Master-Slave J-K flip-flop. Symbol, working and truth table of D flip-flop and T flip-flop. Applications of flip-flops. Concept, modulus, working, truth table, timing diagram of a counter. Asynchronous counter (3-bit, 4-bit), design of mod-N counter: working, truth table and timing diagram, 3-bit synchronous counter: working, truth table and timing diagram.

Self-Learning Hours = 3

Students shall practice constructing timing diagrams for R-S, J-K, D and T flip-flops, and design asynchronous and synchronous counters of a given modulus.

Team Activity Hours = 3

Group project to design and simulate a mod-N counter for a specified value of N and present its timing diagram and applications.

Unit 5: Memories

Contact Hours/Teaching Hours: 07

Classification of memories, RAM, ROM, PROM, EPROM, E2PROM. A-D and D-A converters, circuit diagram and working of R-2R ladder DAC and weighted resistor DAC, DAC specifications.

Self-Learning Hours = 3

Students shall prepare a comparative note on RAM, ROM, PROM, EPROM and E2PROM, and solve numerical problems on R-2R ladder and weighted resistor DAC output voltages.

Team Activity Hours = 3

Case study and group presentation on memory technologies used in modern computing devices and the role of DACs in real-world signal-conversion applications.

Course Articulation Matrix

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3
CO01	3	1	-	-	1	-	1	2	-	-
CO02	2	2	1	1	-	-	1	2	-	-
CO03	1	3	2	1	1	-	1	3	-	-
CO04	-	2	2	2	1	1	2	2	-	-
CO05	1	1	3	2	2	2	2	3	-	-
Avg.	1.75	1.8	2	1.5	1.25	1.5	1.4	2.4	-	-

Department of Computer Science and Engineering

Diploma Computer Science and Engineering

Text Books:

1. Mano M. Morris and Ciletti M. D., 'Digital Design', Pearson Education, 4th Edition.
2. Malvino, Leach, Saha, 'Digital Principles and Applications', TMH.
3. Jain R.P., 'Digital Electronics', PHI.
4. Floyd L. Thomas, 'Digital Fundamentals', Pearson, 10th Edition.

Reference Books:

1. Sedra A.S. & Smith K.C., 'Microelectronic Circuits', (5/e), Oxford, 2004.
2. Zvi Kohavi, 'Switching & Finite Automata Theory', TMH, 2nd Edition.
3. Taub & Schilling, 'Digital Electronics', TMH.

Online Resources:

1. **NPTEL – Digital Circuits**
   https://nptel.ac.in/courses
2. All About Circuits – Digital Electronics
   https://www.allaboutcircuits.com/textbook/digital/
3. **Electronics Tutorials – Digital Electronics**
   https://www.electronics-tutorials.ws/digital/
4. **CircuitVerse – Digital Logic Circuit Simulator**
   https://circuitverse.org/

Graphic Era HILL UNIVERSITY
Established by an Act of the State Legislature of Uttarakhand (Adhiniyam Sankhya 12 of 2011)
University under section 2(f) of UGC Act, 1956

Department of Computer Science and Engineering

Diploma Computer Science and Engineering

Course Name	:	**Relational Database Management Systems**
Course Code	:	**DTCS 303**
Semester	:	**III**
Pre-requisite	:	**Basic knowledge of programming fundamentals and data organization.**

L	T	P	Credits
3	0	0	3

Course Learning Objectives:

- To introduce students to database system concepts, data abstraction, data models and the overall structure of a DBMS.
- To develop understanding of the relational data model, keys, integrity constraints, security, and relational query languages.
- To enable students to populate and query databases using SQL DDL/DML commands
- To build competence in relational database design through functional dependencies and normalization,
- To familiarize students with query processing strategies, transaction concepts, concurrency control and deadlock handling.

Course Outcomes:

After completion of the course, the students will be able to:

Course Outcome	Statement	Bloom's Level	Skill
CO1	Explain the underlying concepts of database technologies, data abstraction, data independence, data models and the components and structure of a DBMS.	**L2 – Understand**	Understanding, Conceptual Thinking
CO2	Apply relational model concepts including keys, integrity constraints, security and authorization, and express queries using relational algebra, relational calculus and views.	**L3 – Apply**	Data Modelling, Analytical Thinking
CO3	Construct SQL queries using DDL/DML commands, constraints, joins, nested subqueries and aggregate functions	**L3 – Apply**	Query Writing, Programming

Department of Computer Science and Engineering

Diploma Computer Science and Engineering

Course Outcome	Statement	Bloom's Level	Skill
CO4	Analyze functional dependencies and normalize a database using 1NF, 2NF, 3NF and BCNF.	**L4 – Analyze**	Database Design, Critical Thinking
CO5	Apply query processing strategies, transaction concepts, serializability, lock-based and time-stamp based protocols, and deadlock handling techniques.	**L3 – Apply**	Transaction Management, Problem Solving

---

Details of the Course:

Unit 1: Database System Concepts **Contact Hours/Teaching Hours: 08**

Basic Concepts, Advantages of a DBMS over File Processing System, Data Abstraction, Database Languages, Data Independence, Components of a DBMS and Overall Structure of a DBMS. Data Models: Network Model, Hierarchical Model, E-R Model, Client Server Architecture.

Self-Learning Hours = 6

Students shall study real database applications, compare file processing systems with DBMS and draw E-R diagrams for given problem domains.

Team Activity Hours = 3

Group activity to design an E-R model for a real-world domain such as library, hospital or university management.

Unit 2: Entity Relationship Model **Contact Hours/Teaching Hours: 08**

Relational Model: Basic Concepts, Attributes and Domains, Entity Types, Entity Sets, Attributes and Keys; Relationship types, Relationship Sets, Structural Constraints; Weak Entity Types; Keys Concept: Candidate and Primary Key, Integrity Constraints: Domain, Entity Integrity Constraints,.

Self-Learning Hours = 6

Students shall convert E-R diagrams into relational schemas and practice writing relational algebra expressions for given queries.

Team Activity Hours = 3

Group exercise on identifying keys and integrity constraints for a shared schema and framing equivalent relational algebra queries.

Unit 3: SQL **Contact Hours/Teaching Hours: 10**

Introduction to SQL Queries, Creating, Inserting, Updating and Deleting Tables and Using Constraints, Set Operations and Operators, Aggregate Functions, String Functions, Date and Time Functions, Null Values, Nested Subqueries, Complex Queries, Join Concepts, Query Languages: Relational Algebra.

Self-Learning Hours = 6

Students shall practice SQL DDL/DML commands, joins, nested subqueries

Team Activity Hours = 3

Department of Computer Science and Engineering

Diploma Computer Science and Engineering

Group project to build and query a small database application using SQL

Unit 4: Relational Database Design **Contact Hours/Teaching Hours: 09**

Purpose of Normalization, Data Redundancy and Updating Anomalies, Functional Dependencies and Decomposition, Process of Normalization Using 1NF, 2NF, 3NF, Multivalued Dependencies and BCNF.

Self-Learning Hours = 6

Students shall identify anomalies in unnormalized relations and normalize given schemas step by step up to BCNF.

Team Activity Hours = 3

Group activity to normalize a real dataset and present the decomposition with functional dependency justification.

Unit 5: Transaction Management **Contact Hours/Teaching Hours: 07**

ACID properties, Concept of Transaction, States of Transactions, Concurrent Executions, Serializability, Recoverability. Lock Based Protocols: Shared and Exclusive Modes, Protocols: Two-Phase Locking, Time-Stamp Based, Validation Based, Deadlock Handling, Deadlock Prevention, Detection and Recovery.

Self-Learning Hours = 6

Students shall solve problems on serializability of schedules and trace lock-based and time-stamp based protocols on sample transactions.

Team Activity Hours = 3

Group discussion and case study on concurrency issues and deadlock scenarios in banking or reservation systems.

Course Articulation Matrix

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO01	3	2	1	-	1	-	-	-	1	-	1	1	2	1	2
CO02	3	2	2	1	1	-	-	-	1	-	1	2	2	1	2
CO03	3	3	3	1	2	-	-	-	1	1	2	2	3	2	3
CO04	3	3	3	2	2	-	-	-	1	1	2	2	3	2	2
CO05	3	3	2	2	2	-	-	-	1	-	2	2	3	2	2
Avg.	3	3	2	2	2	0	0	0	1	1	2	2	3	2	2

Text Books:

1. Elmasri and Navathe, “Fundamentals of Database Systems”, 5th Edition, Pearson Education.

Reference Books:

1. Silberschatz, Korth and Sudarshan, "Database System Concepts", 6th Edition, McGraw-Hill, 2010.
2. Raghu Ramakrishnan and Johannes Gehrke, "Database Management Systems", 3rd Edition, McGraw-Hill

Online Resources:

1. GeeksforGeeks – DBMS https://www.geeksforgeeks.org/dbms/
2. NPTEL – Data Base Management System https://nptel.ac.in/
3. W3Schools – SQL Tutorial https://www.w3schools.com/sql/
4. Oracle Live SQL https://livesql.oracle.com/

Graphic Era HILL UNIVERSITY
Established by an Act of the State Legislature of Uttarakhand (Adhiniyam Sankhya 12 of 2011)
University under section 2(f) of UGC Act, 1956

Department of Computer Science and Engineering

Diploma in Computer Science and Engineering

Course Name	:	**Software Engineering**
Course Code	:	**DTCS304**
Semester	:	**III**
Pre-requisite	:	**Basic knowledge of programming and Computer Fundamentals.**

L	T	P	Credits
3	0	0	3

Course Learning Objectives:

- Wangle skills for developing and testing new software by using multiple approaches that are practiced in software engineering.
- Design a SRS document which include the introduction, goals of implementation, functional, non-functional requirement and structural behavior.
- Access structured model of any software by making the DFDs and UMLs.
- Attain good module decomposition by low coupling and high cohesion.
- Learn different type of testing to make software technically error free and bug free.

Course Outcomes:

After completion of the course, the students will be able to:

Course Outcome	Statement	Bloom's Level	Skill
CO1	Understand software development life cycle models and their applicability.	L2 – Understand	Understanding, Analytical Thinking
CO2	Apply requirement analysis and specification techniques for software projects.	L3 – Apply	Problem Solving, Documentation
CO3	Analyze software testing strategies and quality assurance practices.	L4 – Analyze	Analytical Thinking, Critical Thinking
CO4	Evaluate project management and cost estimation techniques for software projects.	L5 – Evaluate	Decision Making, Evaluation
CO5	Design software systems using appropriate architectural and design patterns.	L6 – Create	Design Thinking, Problem Solving

Department of Computer Science and Engineering

Diploma in Computer Science and Engineering

Details of the Course:

Unit 1: Introduction to Software Engineering **Contact Hours/Teaching Hours: 9**
Basics of Software Engineering: Need for Software Engineering -Definition - Software Characteristics - Software Myths - Program versus Software Products. **Software Development Life Cycle Models:** Introduction - Waterfall Model - Prototyping model - Spiral Model - Iterative Enhancement model. **Software Requirement Analysis (SRS):** Value of good SRS - Requirement Process - Requirement Specification - Desirable characteristics of an SRS - Components of an SRS - Structures of a requirements documents - Problems in SRS - Requirements gathering.

Self-Learning Hours = 6

Students shall study SDLC models and practice drafting requirements.

Team Activity Hours = 3

Group discussion on selecting appropriate SDLC for given project scenarios.

Unit 2: Software Design and Planning **Contact Hours/Teaching Hours: 8**
Software Design: Definition of software design - Objectives of software design - Process of software design - Architectural design - Modular design - Structure chart - Coupling and Cohesion. **Software Planning:** Software metrics - Definition - Types of metrics - Product and Project metrics - Function point and feature point metrics. Software project estimation: Steps for estimation - Reason for poor and inaccurate estimation - Project estimation guidelines - Models for estimation - COCOMO Model Introduction.

Self-Learning Hours = 6

Students shall practice software estimation using the COCOMO model.

Team Activity Hours = 3

Group activity on designing modular architecture and calculating function points.

Unit 3: Software Maintenance and Risk Management **Contact Hours/Teaching Hours: 8**
Software Maintenance: Software as an evolution entity - Software configuration management activities - Change control process - Software version control - Software configuration management - Need for maintenance - Categories of maintenance - Maintenance cost - Factors affecting the effort. **Risk management:** Definition of risk - Basics for different types of software risks - Monitoring of risks - Risk management - Risk avoidance - Risk detection - Risk control - Risk recovery - Sources of risks - Types of risks.

Self-Learning Hours = 6

Students shall study risk management techniques and mitigation strategies.

Team Activity Hours = 3

Case study on software maintenance, version control, and risk analysis.

Unit 4: Software Testing **Contact Hours/Teaching Hours: 8**
Software Testing: Introduction to testing - Testing principles - Testing objectives - Basic terms used in testing - Fault - Error - Failure - Test cases - Black box and white box testing - Advantages and disadvantages of above testing - Testing activities - Test plan. **Levels of testing:** Unit testing - Integration tests - System testing Types. **Software Testing strategies:** Static testing strategies - Formal technical reviews - Code walkthrough - Code inspection - Debugging - Definition - Characteristics of bugs - Life cycle of a Debugging task - Debugging approaches.

Self-Learning Hours = 6

Students shall practice writing test cases and debugging approaches.

Team Activity Hours = 3

Group programming activity to perform black-box and white-box testing.

Unit 5: Software Reliability and Quality Assurance **Contact Hours/Teaching Hours: 7**

Software Quality Assurance: Verification and validation - SQA Objectives and Goals - SQA plan - Definition of software quality - Classification of software qualities - Software quality attributes - Five levels. ISO 9000: Need for ISO Certification - Benefits of ISO 9000 certification - Limitation of ISO 9000 certification - Uses of ISO - Salient features of ISO 9000 Requirements - Introduction to ISO 9126.

Self-Learning Hours = 6

Students shall study software quality standards and attributes.

Team Activity Hours = 3

Group discussion on ISO 9000 certification benefits and limitations.

Course Articulation Matrix

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3
DTCS304.1	3	1	-	-	1	-	1	-	2	-
DTCS304.2	2	2	1	1	-	-	1	-	2	-
DTCS304.3	1	3	2	1	1	-	1	-	3	-
DTCS304.4	-	2	2	2	1	1	2	-	2	-
DTCS304.5	1	1	3	2	2	2	2	-	3	-
Avg.	**1.75**	**1.8**	**2**	**1.5**	**1.25**	**1.5**	**1.4**	**-**	**2.4**	**-**

Text Books:

1. R. S. Pressman, Software Engineering: A Practitioner's Approach, McGraw-Hill, Seventh Edition, 2010.
2. P. K. J. Mohapatra, Software Engineering (A Life Cycle Approach), New Age International Publishers, First Edition, 2009.

Reference Books:

1. K. K. Aggarwal, Yogesh Singh, Software Engineering, New Age International Publishers, Third Edition, 2007.
2. Ian Sommerville, Software Engineering, Addison Wesley, Ninth Edition, 2010.

Department of Computer Science and Engineering

Diploma in Computer Science and Engineering

Online Resources:

1. **GeeksforGeeks – Software Engineering**
   https://www.geeksforgeeks.org/software-engineering/

2. **Tutorialspoint – Software Engineering Tutorial**
   https://www.tutorialspoint.com/software_engineering/

3. **ISO 9000 Quality Management**
   https://www.iso.org/iso-9001-quality-management.html

Department of Computer Science and Engineering

Course Name	:	**Applied Mathematics**
Course Code	:	**DTMA301**
Semester	:	**III**
Pre-requisite	:	**Fundamentals knowledge of applied Maths and its properties.**

L	T	P	Credits
3	0	0	3

Course Learning Objectives:

- Understand the concepts and techniques of integration and their applications.
- Solve definite integrals using appropriate integration methods and properties.
- Formulate and solve first-order differential equations arising in engineering and physical sciences.
- Apply probability distributions to analyze uncertainty and production-related problems.
- Use numerical methods to obtain approximate solutions of algebraic and simultaneous equations efficiently.

Course Outcomes:

After completion of the course, the students will be able to:

Course Outcome	Statement	Bloom's Level	Skill
CO1	Apply the fundamental concepts of integration, including standard integrals, substitution, partial fractions, trigonometric substitution, and integration by parts to solve mathematical problems.	**L3 – Apply**	Problem Solving
CO2	Evaluate definite integrals using their properties and apply integration techniques to solve practical mathematical problems.	**L5 – Evaluate**	Critical Thinking
CO3	Solve first-order differential equations using appropriate methods and apply them to model engineering problems such as rectilinear motion and simple harmonic motion.	**L3 – Apply**	Mathematical Modeling
CO4	Analyze Binomial, Poisson, and Normal probability distributions and apply them to solve problems related to production processes and uncertainty.	**L4 – Analyze**	Analytical Reasoning
CO5	Apply numerical methods, including Bisection, Regula-Falsi, Newton-Raphson, Gauss Elimination, Jacobi, and Gauss-Seidel methods, to solve algebraic and simultaneous equations.	**L3 – Apply**	Computational Skills

Department of Computer Science and Engineering

Details of the Course:

Unit 1: **Contact Hours/Teaching Hours: 8**

INTEGRATIONS: Definition of integration as anti-derivative. Integration of standard function. Rules of integration (Integrals of sum, difference, scalar multiplication). Methods of Integration. Integration by substitution Integration of rational functions. Integration by partial fractions.

Unit 2: **Contact Hours/Teaching Hours: 8**

Integration by trigonometric transformation. Integration by parts. Definite Integration: Definition of definite integral. Properties of definite integral with simple problems.

Unit 3: **Contact Hours/Teaching Hours:8**

DIFFERENTIAL EQUATION: Definition of differential equation, order and degree of differential equation. Formation of differential equation for function containing single constant. Solution of differential equations of first order and first degree such as variable separable type, reducible to Variable separable, Homogeneous, Nonhomogeneous, Exact, Linear and Bernoulli equations. Applications of Differential equations. Rectilinear motion (motion under constant and variable acceleration): Simple Harmonic Motion.

Unit 4: **Contact Hours/Teaching Hours: 7**

PROBABILITY DISTRIBUTION: Binomial distribution. Poisson's distribution. Normal distribution Simple examples corresponding to production process.

Unit 5: **Contact Hours/Teaching Hours: 08**

NUMERICAL METHODS: Solution of algebraic equations: Bisection method. Regulafalsi method. Newton – Raphson method. Solution of simultaneous equations containing 2 and 3 unknowns: Gauss elimination method. Iterative methods- Gauss seidal and Jacobi's methods.

Course Articulation Matrix

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3
CO01	3	3	2	2	2	1	1	2	2	3
CO02	3	3	2	3	3	2	1	2	3	3
CO03	3	3	2	3	2	2	2	2	3	3
CO04	3	3	2	2	3	3	2	2	3	3
CO05	3	3	3	2	2	2	2	3	3	3
Avg.	3	3	2.2	2.4	2.4	2	1.6	2.2	2.8	3

Text Books:

1. Calculus: single variable Robert T. Smith Tata McGraw Hill
2. Advanced Mathematics for Engineers and Scientist Murray R Spiegel Schaum outline series McGraw Hill.
3. Higher Engineering Mathematics B. S. Grewal Khanna Publication, New Delhi.

Reference Books:

1. Numerical methods for Engg. 4th ed. Chapra Tata McGraw Hill.

2. Numerical methods for scientific & engineering computations M. K. Jain & others Wiley Eastern Publication.
3. Introductory Methods of Numerical analysis S. S. Sastry Prentice Hall Of India New Dehli.

Online Resources:

1. https://books.google.co.in/books?id=5XappvcENCMC&printsec=copyright#v=onepage&q&f=false
2. https://byjus.com/maths/differentiation-integration/
3. https://vfu.bg/en/e-Learning/Math--Soong_Fundamentals_of_probability_and_statistics_for_engineers.pdf