Daffodil International University

B.Sc. in Software Engineering
(2010-2020

Level	Course Code with Course Title	Credit	Course Category	Sem.-wise Credit
Level-1, Term-1	SWE112: Computer Fundamentals with Lab	4	CORE	11
	SWE111: Introduction to Software Engineering	3	CORE	
	PHY114: Physics with Lab	4	GED	
Level-1, Term-2	ENG123: English Language	3	GED	13
	MAT113: Mathematics-I (Calculus and Differential Equations)	3	GED	
	SWE121: Software Requirement Analysis and Design	3	CORE	
	SWE122: Programming Language with Lab	4	CORE	
Level-1, Term-3	MAT221: Mathematics-II	3	GED	13
	SWE231: Software Engineering Project-I	3	LAB-P	
	SWE133: Data Structure with Lab	4	CORE	
	STA134: Statistics & Probabilities	3	GED	
Level-2, Term-1	SWE132: Java Programming with Lab	4	CORE	12
	SWE213: Computer Algorithms with Lab	4	CORE	
	SWE211: Introduction to Database with Lab	4	CORE	
Level-2, Term-2 100	SWE233: Object Oriented Concept & Design with lab	4	CORE	14
	SWE222: Software Engineering Quality Assurance & Testing	3	CORE	
	SWE223: Digital Electronics with Lab	4	CORE	
	SWE224: Discrete Mathematics	3	CORE	
Level-2, Term-3 175	SWE131: Documentation of Software Engineering	3	CORE	13
	SWE232: Operating System with Lab	4	CORE	
	SWE212: Software Project Management	3	CORE	
	ACC 124: Principle of Accounting / SWE435: Business Communication	3	PGC	
Level-3, Term-1 210	SWE323: System Analysis and Design	3	CORE	13
	SWE312: Theory of Computing	3	CORE	
	SWE322: Software Security	3	CORE	
	SWE313: .Net Programming with Lab	4	CORE	
Level-3, Term-2 70	SWE321: Data Communication with Lab	4	CORE	14
	SWE333: Desktop and Web Programming with Lab	4	CORE	
	SWE311: Computer Architecture & Organization	3	CORE	
	SWE413: Software Engineering and Cyber Laws	3	PGC	

Level-3, Term-3 100	SWE412: Management Information System	3	PGC	14
	SWE331: Object Oriented Software Development	3	LAB-P	
	SWE422: Numerical Analysis with Lab	4	CORE	
	SWE424: Artificial Intelligence with lab	4	CORE	
Level-4, Term-1 70	SWE423: Advance Database Management System with Lab	4	CORE	15
	SWE425: Telecommunication Engineering with Lab	4	CORE	
	SWE426: Distributive Computing and Network Security with Lab	4	CORE	
	SWE332: Software Engineering Project II (Web Programming)	3	LAB-P	
35	SWE421: Multimedia Programming with Lab / SWE411: Computer Networking with Lab	4	PGC	7
	SWE439: Project/Thesis	3	PRO-THE:	
	Total Credits	139		

* Course Categories:

1. GED: General Education Courses
2. PGC: Preparatory General Core
3. CORE: Software Engineering Core
4. LAB-P: Lab Based Project Courses
5. PRO-THE: Final Year Project/ Thesis

B.Sc. in Software Engineering
Course Details

Course No. SWE-111
Course Title: *Introduction to Software Engineering*
Number of Credit: 3

The subject provides an introduction to the discipline of Software Engineering. The emphasis is upon a broad coverage of the areas, since students will at this early stage not have adequate programming skills to tackle many of the topics in greater depth. The notion of a software system as a model or approximation of a desired system is introduced, and used as a way of describing such things as the software life cycle and its various models, programming by contract, design and testing issues, maintenance, reuse, complexity, divide and conquer strategies, metrics, and measurement. Topics in project management include project planning and scheduling; project control; risk analysis; planning and monitoring; process management and process improvement; configuration management and control; software acquisition; contract briefing, negotiation and management.

Course No. SWE - 112
Course Title: *Computer Fundamentals with Lab*
Number of Credit: 4

Introduction to fundamental concepts of computing systems. Evolution and generations of computers, capabilities, applications, and limitations of modern computing systems. Overview of computer hardware components including CPU, memory, storage devices, and input/output systems. Software concepts including system software, operating systems, and application software. Data representation and storage, number systems (binary, octal, hexadecimal), binary arithmetic, two's complement representation, and conversion between number systems including fractional values.

Fundamentals of algorithmic thinking and problem solving. Introduction to algorithms, flowcharts, and basic control structures such as loops, counters, accumulators, and conditional branching. Emphasis on computational logic and structured problem analysis as a foundation for software engineering and computer science studies.

Laboratory Component:
Hands-on laboratory sessions focused on practical computing skills and digital productivity tools commonly used in academic and professional environments. Practical exercises include word processing, spreadsheet-based data analysis, database fundamentals, web page creation, and presentation design using widely adopted office productivity and open-source tools (e.g., Microsoft Office and/or LibreOffice suites). The lab emphasizes document preparation, data

organization, basic automation, and effective digital communication.

Course No. MATH – 113
<u>Course Title: *Mathematics-1 (Calculus)*</u>
Number of Credit: 3

Differential Calculus:

Functions and their properties: domain and range, odd and even functions, one-to-one and onto functions, inverse functions, and graphical representation of functions. Limits of functions, continuity, and differentiability. Techniques of differentiation and their applications. Successive differentiation and Leibniz's theorem. Rolle's Theorem and the Mean Value Theorem in Lagrange's and Cauchy's forms. Taylor's theorem and series expansion. Indeterminate forms and L'Hôpital's rule. Partial differentiation and Euler's theorem on homogeneous functions. Applications of derivatives including maxima and minima of functions, tangents, and normals.

Multivariable Calculus:

Functions of several variables; limits and continuity of multivariable functions. Partial derivatives and higher-order derivatives. Total differential and chain rule for multivariable functions. Gradient and directional derivatives. Jacobian and its applications. Taylor's expansion for functions of two or more variables. Unconstrained and constrained optimization, including extrema of multivariable functions using Lagrange multipliers. Applications of multivariable calculus in geometry and optimization problems.

Integral Calculus:

Definition and physical significance of integration. Integration as the inverse process of differentiation. Definite integrals and the Fundamental Theorem of Integral Calculus. Reduction formulae. Improper integrals. Double integrals and their applications. Evaluation of areas and volumes using integration. Beta and Gamma functions. Rectification of curves.

Course No. PHY - 114
Course Title: *Physics with Lab*
Number of Credit: 4

Mechanics: Basic Concept of Mechanics: Classification of Mechanics, Motion in One Dimension: Equations of motion; Motion in Two Dimensions: Projectile motion; Particle Dynamics: Force and momentum; Frictional Forces: Laws of Friction, Co-efficient of Friction, Angle of Friction, Equilibrium of a body on an inclined surface, advantage due to friction. Work and Energy: Work done by variable force, Work Energy Theorem.

Waves & Oscillation: Wave motion and propagation: Different types of waves, Simple Harmonic Oscillator: Characteristics of SHM, Differential Equation for SHM, Doppler Effect, Stationary Waves.

Heat & Thermodynamics: Concept of Temperature & Zeroth Law; Kinetic theory of Gases: General postulates, Mean Square Velocity and RMS Velocity, Pressure of gas according to Kinetic Theory, Mean free path; Brownian Motion; Laws of Thermodynamics & Their Applications: First Law of Thermodynamics, Internal Energy, Isothermal and Adiabatic relations; Second law of Thermodynamics, Reversible and Irreversible Processes, PV Diagram, Carnot‘s Cycle, Carnot Theorem, Concept of Entropy;

Optics: Theory of Light: Corpuscular Theory, Wave Theory, EM Theory, Quantum Theory, Huygens‘ principle,

Interference: Interference of Light, Young‘s Double Slit Experiment, Interference Fringes; Diffraction: Diffraction of Light, Fresnel, and Fraunhofer Diffraction; Polarization: Different types of polarization

Modern Physics: Particle and Wave Nature of Light: Dual Nature of Light, De Broglie Hypothesis; Atomic Models:
Thomson Model, Rutherford model, Bohr Model; Radioactivity and Nuclear Phenomena: Radioactivity, Mean Life, Half Life, Disintegration Constant, Radioactive Decay law, Nuclear Fission and Fusion; Basis of Quantum Mechanics: Failure of Classical Mechanics, Origin of Quantum Mechanics, Uncertainty Principle, Relativistic Mechanics: Special Theory of relativity, Length Contraction, Time Dilation.

Course No. SWE - 121

Course Title: *Software Requirement Analysis and Design*

Number of Credit: 3

Software requirements: Functional and non-functional requirements, user requirements, system requirements, interface specifications, Software requirement documents.

Requirement Engineering Process: Feasibility studies, requirements elicitations and analysis, requirement validations, and requirement management.

System models: context models, behavioral models, data models, object models, CASE workbenches.

Architectural Design: Architectural design decisions, system organization, Decomposition Styles, Control Styles, Reference architectures.

Distributed system architectures: Multiprocessor architectures, client-server architectures, distributed object architectures, and inter-organizational computing.

Course No. SWE - 122

Course Title: *Programming Language with Lab*

Number of Credit: 4

This course introduces fundamental programming concepts and structured program development. Topics include program development stages, problem analysis, and algorithm representation using flowcharts. The course emphasizes structured programming principles and procedural programming paradigms.

Core language constructs include character sets, reserved words, identifiers, data types and type modifiers, constants, variables, expressions, and operators. Control structures such as conditional statements, loops, and branching are studied in detail. Functions and modular program design are covered, including parameter passing mechanisms, scope rules, storage classes, recursion, and program structure. Use of header files and preprocessor directives is introduced.

Advanced topics include pointers and arrays, dynamic memory usage, and user-defined data types such as structures, unions, and enumerations. Input and output mechanisms include standard and formatted I/O, file handling, and command-line parameters. Additional topics include variable-length argument lists, basic error handling techniques, and an introduction to simple graphics programming.

Laboratory Component:

The laboratory sessions reinforce theoretical concepts through hands-on programming exercises. Students design, implement, debug, and test programs using a structured programming language (e.g., C or an equivalent). Emphasis is placed on writing clean, efficient, and well-documented code, applying good programming practices, and solving practical computational problems.

Course No. SWE - 123

Course Title: *English Language*

Number of Credit: 3

Course Contents:

A: ***Written communication Skills will include communicative expression for:***

Sentence: Its parts, kinds, position, & usages; Structure on: SVOCA, Vocabulary, Derivative; The art of effective writing; Note taking and summarizing; Story writing; Paragraph writing; Effective rules on grammar to make a correct sentence; Style between UK & USA English; The ways to translate easily.

B: ***Oral communication Skills will include communicative expression for:***

Phonetic sounds and symbol, stress and intonation etc. on daily life words; Conversation**:** Personal Identification (name, occupation, nationality etc); life at home; education and future career; traveling, postal, telephonic and telegraphic purposes; Health and welfare; Food and drink; Giving advice, Instruction and direction; Request, complaints, problem & apologies; Describing objects, places, people; Comparison and contrast; Processes and sequencings; Debate on the current issues telling; Life story/history; Situational conversation; Story telling; Extemporary/offhand speech; Hobby/liking/disliking.

C: ***Writing Skills*****:**

Basic: Sentence constructions & Grammar review

Paragraph: Topic, Sentence, Supporting, and Orders of development

From Paragraph to Essay: Making Outline; Organizing Paragraphs; Learning: Introduction, Transition, and Conclusion, etc.

D: ***Specific Applications:***

a.Formal Letter, Blocked, Semi-blocked, Application,
Complaint, Request/Order b. Résumé/CV
c. Report
d. Memo

Course No. SWE-131

Course Title: *Documentation of Software Engineering*

Number of Credit: 3

Fundamental models, tools, and techniques for working with documents. Problems with paper-based and on-line documentation; types of technical documentation used in software engineering; document specifications; minimalist design philosophy; graphic design of technical documentation; the context of technical writing; the writing process (analysis, planning, generation, testing, revision and maintenance of written texts); document publication techniques (including Lyx and LaTeX); the role of hypertext, hypermedia and markup languages in technical documentation; small-volume and large-volume hypertext; collaborative hypertext; intelligent hypertext. Theoretical and commercial perspectives, including its pivotal role in building and using the World Wide Web.

Course No. SWE-132
Course Title: *Java Programming with Lab*
Number of Credit: 4

Course Description

This course introduces Java as a hybrid, object-oriented programming language with strong support for both procedural and object-oriented paradigms. Topics include the structure of Java programs, the compilation and execution process, and the design of modular, maintainable code.

Core Language Features:
Java data types, control structures, functions, scoping rules, and references. String manipulation and namespace management. Use of constants (final) and references. Pointers are handled implicitly through references to objects.

Object-Oriented Concepts:
Classes and data abstraction, separation of interface and implementation, inheritance, abstract classes, and multiple inheritance via interfaces. Polymorphism, dynamic method dispatch (late binding), type conversion, and runtime type identification (RTTI). Static class members and access control mechanisms. Exception handling and robust error management.

Advanced Features:
Generics with function templates and class templates, and the use of the Java Standard Library (collections, I/O streams, and utility classes). Stream-based input/output covering standard I/O, file I/O, and device I/O. Overview of operator overloading restrictions in Java, and understanding of friend-like behavior via package access.

Laboratory Component:
The lab complements theoretical topics through hands-on exercises. Students design, implement, and test Java programs, applying object-oriented principles, exception handling, generics, and standard libraries. Emphasis is placed on problem-solving, code modularity, documentation, debugging, and performance analysis.

Course No. SWE - 133

Course Title: *Data Structure with Lab*

Number of Credit: 4

This course introduces fundamental concepts of **data structures and algorithms**, emphasizing the organization, storage, and efficient manipulation of data. Students study both **abstract concepts** and their **practical implementation**.

Introduction and Basic Concepts:
Elementary data organization, basic operations on data, and algorithmic thinking. Time and space complexity, trade-offs, and mathematical notations for algorithm analysis. Introduction to strings, string storage, operations, and pattern-matching algorithms. Overview of data abstraction, primitive operations, and memory management.

Arrays, Records, and Pointers:
Linear arrays: representation in memory, traversal, insertion, and deletion. Sorting algorithms including Bubble, Insertion, Selection, Merge, Quick, and Radix sort. Searching techniques: linear and binary search. Linked lists: memory representation, traversal, insertion, deletion, header-linked lists, and doubly linked lists. Pointer usage and dynamic memory allocation.

Stacks, Queues, and Recursion:
Stack and queue structures, array-based and linked representations, arithmetic expression evaluation, and Polish notation. Recursion and recursive problem-solving techniques.

Advanced Data Structures:
Trees, graphs, priority queues, hash tables, and compound structures. Operations on sets, graph traversal, and basic dictionary structures. Emphasis on performance analysis and algorithm efficiency.

Algorithm Analysis and Applications:
Introduction to algorithm design methods including recursion and backtracking. Analysis of algorithms in terms of time and space complexity. Overview of data structures in practical systems including databases, and discussion of their performance and real-world applications.

Laboratory Component:
The lab component provides hands-on experience implementing and testing data structures and algorithms using a programming language (e.g., C, C++, or Java). Students perform memory management, implement sorting and searching algorithms, work with dynamic structures, and analyze algorithm performance experimentally. Emphasis is placed on correctness, efficiency, and documentation.

Course No. SWE -134
Course Title: *Statistics & Probabilities*
Number of Credit: 3

Statistics: Definition, Scope, and Importance. Statistics and computer science.
Statistical data: Type of data: Qualitative and quantitative, discrete and continuous variables, variable and attribute. Level of measurement: normal, ordinal, interval, and ratio scale. Population and sample. **Frequency distribution:** Presentation of statistical data, classification, and tabulation. Construction of a frequency distribution table.

Graphical and diagrammatic presentation of data: Use of bar diagram, histogram, frequency polygon, cumulative frequency curve, scatter diagram, pie diagram, etc. Types of frequency curve. **Descriptive statistics**: Measures of central tendency: mean, median, mode, partition values. Measures of dispersion: range, quartile deviation, mean deviation, and coefficient of variation. Moments of distribution, skewness, kurtosis, and their measures.

Correlation and regression: Simple correlation, measures of correlation, and its interpretation. Rank correlation and linear and non-linear regression. Method of least squares and curve fitting. **Sampling and sample designs:** sample vs. Census. Role of sampling. Different types of sample designs. Simple random sampling, stratified sampling, systematic sampling, and cluster sampling. **Probability:** Elements of set algebra. Definition of probability and related concepts. Marginal and conditional probability. Laws of probability. Bayes' theorem on probability.

Random variables: Probability mass function. Probability density function, joint marginal and conditional probability distribution. Independence of random variables. Laws of expectation. **Test of significance:** basic concepts, important steps in the test of significance. Use of normal test, t-test, chi-square, and F-test.

Stochastic process: Basic concept and illustration. Markov chain, Poisson process, birth and death process. Queuing process

Course No. SWE- 211
Course Title: Introduction to *Database with Lab*
Number of Credit: 4

Principles, design, administration, and implementation of database management systems. Topics include: entity-relationship model, relational model, and relational algebra And calculus, database design and normalization, database query languages, indexing schemes, security, integrity, concurrency control, and contemporary topics in database management systems.

CSE 311/CIS 311 Database Management System: *Introduction: purpose of DBMS, view of data, data models, database languages, transaction management, database users, overall system* *structure; Entity Relationship Model: design issues, mapping constraints, keys, ER diagram, extended ER features; Relational Model/SQL: basic structures, set operations, aggregate functions, nested subqueries, derived relations, views, modification of database, joined relations, data-definition language, embedded SQL; Integrity Constraints: domain constraints, referential integrity, assertions, triggers; Relational Database Design: pitfalls in relational-database design, decomposition, normalization; Object-Relational Databases: complex types and object orientation, querying with complex types, creation of complex values and objects; Storage and File Structure: file organization, organization of records in files, data-dictionary storage, storage structures for object-oriented databases; Indexing and Hashing: ordered indices, B^+-tree index files, B-tree index files, static hashing, dynamic hashing, index definition in SQL, multiple-key access.*

Course No. SWE – 212

Course Title: *Software Project Management*

Number of Credit: 3

This course provides a comprehensive overview of **software project management principles, practices, and tools** in contemporary organizational settings. Students gain an understanding of the complete project lifecycle, from initiation to closure, with emphasis on practical application in software development projects.

Key Topics:

- **Project Foundations:** Project selection, definition, and feasibility analysis. Characteristics and responsibilities of the project manager.

- **Team and Organization:** Project team structures, roles, responsibilities, and coordination mechanisms. Communication and leadership within project teams.

- **Project Planning:** Budgeting, scheduling, resource allocation, risk management, and project documentation. Use of planning techniques and tools for effective project execution.

- **Project Monitoring and Control:** Performance tracking, progress reporting, project auditing, quality management, and corrective actions.

- **Project Termination:** Successful project closure, evaluation, and post-project review.

- **Software Project Management Tools:** Use of software aids and information systems to support planning, monitoring, control, and documentation.

- **Information Systems Projects:** Specific considerations for managing IT and software projects, including software development methodologies and lifecycle models.

Laboratory / Practical Component (if applicable):
Students gain hands-on experience in project planning, scheduling, and monitoring using project management software tools (e.g., MS Project, Primavera, or open-source alternatives). Case studies and simulations provide practical insight into real-world project challenges, decision-making, and team management.

Course No. ACC- 124
Course Title: *Principles of Accounting*
Number of Credit: 3

Financial Accounting: Objectives and importance of accounting; Accounting as an information system; computerized system and applications in accounting. Recording system: double entry mechanism; accounts and their classification; Accounting equation; Accounting cycle: journal, ledger, trial balance; Preparation of financial statements considering adjusting and closing entries; Financial statement analysis and interpretation: ratio analysis; Accounting concepts (principles) and conventions.

Cost and Management Accounting: Cost concepts and classification; Overhead cost: meaning and classification; Distribution of overhead cost; Overhead recovery method/rate; Job order costing: preparation of job cost sheet and quotation price; Inventory valuation: absorption costing and marginal/variable costing technique; Cost-Volume-Profit analysis: meaning, breakeven analysis, contribution margin approach, sensitivity analysis. Short-term investment decisions: relevant and differential cost analysis; Long-term investment decisions: capital budgeting, various techniques of evaluation of capital investments.

Course No. SWE - 213
Course Title: *Computer Algorithms with Lab*
Number of Credit: 4

This course focuses on the design, analysis, and implementation of efficient algorithms. It covers fundamental techniques for algorithm design and performance analysis, including time and space complexity, asymptotic notation, and correctness proofs.

Algorithm design paradigms include divide and conquer, greedy algorithms, dynamic programming, and backtracking. Core algorithmic topics include sorting and searching algorithms, data structures for sets such as heaps and hash tables, and fundamental graph algorithms including breadth-first search, depth-first search, shortest path algorithms, and network flow. Additional topics include elementary parallel algorithms, computational geometry, algebraic simplification and transformation, integer arithmetic (including greatest common divisor computation), polynomial and matrix computations.

The course introduces lower bound theory, complexity classes, and computational intractability, with an overview of NP-hard and NP-complete problems. Emphasis is placed on both theoretical foundations and practical algorithm design techniques used in real-world software systems.

Laboratory Component:
The laboratory sessions support the theoretical content through hands-on implementation and experimental analysis of algorithms. Students implement and test algorithms using appropriate programming languages and tools, analyze empirical performance, and compare theoretical and practical complexity. The lab emphasizes problem-solving, algorithmic efficiency, and correctness verification.

Course No. MATH - 221

Course Title: *Mathematics-II*

Number of Credit: 3

Linear Algebra:
Matrices and matrix algebra, including definitions and classifications of matrices. Algebraic operations on matrices. Adjoint and inverse of matrices. Elementary row and column transformations and their applications. Matrix polynomials. Cayley–Hamilton theorem with applications to matrix powers, rank, and nullity. Rank–nullity theorem. Normal forms and canonical forms of matrices. Systems of linear equations and solution techniques. Eigenvalues and eigenvectors, diagonalization, and applications in engineering and applied sciences.

Complex Variables:
Complex number system and functions of a complex variable. Limits, continuity, and differentiability of complex functions. Analytic functions and elementary properties. Complex integration and related theorems.

Fourier Analysis:
Real and complex forms of Fourier series. Finite Fourier transforms. Fourier integrals and Fourier transforms. Applications of Fourier methods in solving boundary value problems, particularly wave and diffusion equations.

Multivariable Calculus:
Functions of several variables, partial derivatives, total derivatives, multiple integrals, and optimization of multivariable functions. Constrained optimization using Lagrange multipliers with applications in science and engineering.

Differential and Difference Equations:
First- and second-order linear differential equations and their solutions. Difference equations and discrete dynamical systems. Applications in modeling physical, engineering, and economic systems, with emphasis on stability analysis.

Course No. SWE- 222
Course Title: *Software Engineering Quality Assurance and Testing*
Number of Credit: 3

Introduction to Software Maintenance: Maintenance types, Maintenance factor, Requirement and Analysis, Reengineering Factor.

Quality Assurance: Introduction to Quality Assurance, Quality Factor, McCall's Factor, FTR Guideline, Quality Assurance Group, Quality Assurance Guideline, Documentation, Different types of Team. Testing: Selection of test cases, White Box Testing and Black Box Testing, Testing Strategy, Program instrumentation, Data flow analysis, Domain testing strategy, Mutation analysis, Basics of reliability theory, Reliability modeling.

Course No. SWE - 223
Course Title: *Digital Electronics with Lab*
Number of Credit: 4

This course provides a comprehensive introduction to **digital electronics**, including the theoretical foundations, circuit design techniques, and practical applications of digital systems. Emphasis is placed on Boolean algebra, combinational and sequential logic, integrated circuits, and memory devices.

Boolean Algebra and Minimization:
Number systems and digital codes. Boolean constants and variables, truth tables, and basic logic functions. Boolean expressions and circuit implementation from expressions. Boolean theorems and DeMorgan’s laws. Sum-of-products and product-of-sums forms. Simplification techniques including algebraic methods, Karnaugh maps, and Quine–McCluskey method.

Logic Gates and Combinational Circuits:
Design and implementation of logic gates and combinational circuits. Circuit realization using NAND and NOR gates, alternate representations, and design examples. Study of XOR and XNOR gates.

Flip-Flops and Sequential Circuits:
Clocked SR, JK, D, and T flip-flops, master-slave configurations, timing considerations, and practical applications. Frequency division, counters, and timing circuits. Stability and metastability in flip-flop circuits.

Arithmetic Circuits:
Binary arithmetic circuits including adders, subtractors, carry look-ahead and parallel adders. BCD adders, binary multipliers, and 2’s complement arithmetic.

Counters and Registers:
Asynchronous (ripple) and synchronous counters, modulo-n counters, shift registers, and cascading counters. Applications of counters in frequency division and digital clocks. Propagation delay considerations in sequential circuits.

Medium-Scale Integration (MSI) Logic Circuits:
Decoders, encoders, multiplexers, demultiplexers, and BCD-to-7-segment driver circuits. Practical design applications of MSI circuits.

Integrated-Circuit Logic Families:
Digital IC terminologies. TTL, CMOS, and ECL logic families, characteristics, interfacing techniques, tristate logic, and MOSFET applications.

Memory Devices:
Memory concepts, operation, and types of semiconductor memories including ROM, static and dynamic RAM, magnetic bubble memory, and CCD memory.

Laboratory Component:
The laboratory reinforces theoretical concepts through hands-on experiments. Students design and implement combinational and sequential circuits, work with flip-flops, counters, registers, arithmetic circuits, MSI components, and memory devices. Emphasis is placed on simulation, circuit testing, debugging, and performance evaluation.

Course No. SWE - 224
Course Title: *Discrete Mathematics*
Number of Credit: 3

Set theory, Relations; Functions; Vectors and Matrices; Graph Theory; Planner Graphs and Trees; Directed Graphs and Binary trees; Combinational Analysis; algebraic Systems; Languages; Grammar and automata; Ordered sets and lattices; Prepositional calculus; Boolean algebra; Logic gates; Lattice and group theory; Cyclic Groups; Permutation groups, Symmetry groups; quotients; Homomorphism; Basic structure theory. Set and combinatory. Prepositional and predicate logic, mathematical reasoning, and proof techniques. Theories and induction. Counting and countability. Graphs and trees. Relations and functions. Orphisms and algebraic structures. Modeling computation. Program correctness and verification.

Logic and Proofs: Propositional and predicate logic, mathematical induction, and various proof techniques.

Set Theory and Relations: Operations on sets, functions, equivalence relations, and partial orders.

Combinatorics: Counting principles (permutations, combinations), binomial coefficients, and the pigeonhole principle.

Graph Theory: Basic concepts, Eulerian and Hamiltonian graphs, trees, and spanning trees.

Number Theory: Divisibility, modular arithmetic, prime numbers, and the Chinese Remainder Theorem.

Recursion: Difference equations and generating functions.

Course No. SWE-231

Course Title: *Software Engineering Project-1 (using C)*

Number of Credit: 3

The design and implementation of a program covering all aspects of the software engineering process, including requirements analysis, specification, design, coding, testing and maintenance. Students will be expected to attend regular meetings with their project supervisor.

Systems Programming: Building low-level applications, such as file systems, memory managers, or basic operating system components.

Software Lifecycle: Applying the Software Development Life Cycle (SDLC), including requirements gathering, design, and documentation.

Version Control: Mandatory use of Git for collaborative coding and history management.

Technical Quality: Emphasis on manual memory management (pointers), unit testing, and debugging with tools like GDB.

This lab based on this course.

Course No. SWE -232

Course Title: *Operating System with Lab*

Number of Credit: 4

This course introduces the fundamental principles and design concepts of modern operating systems. Topics include operating system structures and design objectives, process management, and the execution of sequential and concurrent processes. Inter-process communication, synchronization mechanisms, mutual exclusion, semaphores, and deadlock characterization, prevention, avoidance, and recovery are studied in detail.

The course covers CPU scheduling and dispatching strategies, process cooperation, and performance evaluation of scheduling algorithms. Memory management techniques including addressing schemes, paging, segmentation, virtual memory, store multiplexing, and time-sharing systems are discussed. File systems, storage management, and access methods are examined along with basic concepts of operating system security and protection mechanisms.

Case studies of contemporary operating systems are used to illustrate design trade-offs and practical implementations in large-scale information processing systems.

Laboratory Component:

The laboratory complements the theoretical concepts through hands-on experiments and implementation exercises. Students work with operating system mechanisms such as process scheduling, synchronization, memory management, and file systems using simulated or real operating system environments. The lab emphasizes system-level programming, performance measurement, and experimental evaluation of operating system policies.

Course No. SWE - 233

Course Title: *Object Oriented Concepts & Design with Lab*

Number of Credit: 4

Philosophy of Object Oriented Programming (OOP); Advantages of OOP over structured programming; Encapsulation; Classes and objects; Access specifiers; Static and non-static members; Constructors, destructors and copy constructors; Array of objects, object pointers, and object references; Inheritance: single and multiple inheritance; Polymorphism: overloading, abstract classes, virtual functions and overriding; Exceptions; Object Oriented I/O; Template functions and classes; Multi- threaded Programming.

This lab based on this course.

Course No. SWE -311

Course Title: *Computer Architecture & Organization*

Number of Credit: 3

Information representation; Measuring performance; Instructions and data access methods: operations and operands of computer hardware, representing instruction, addressing styles; Arithmetic Logic Unit (ALU) design: arithmetic and logical operations, floating point operations, designing ALU; Processor design: datapaths - single cycle and multicycle implementations; Control Unit design - hardware and micro programmed; Hazards; Exceptions; Pipeline: pipelined datapath and control, superscalar and dynamic pipelining; Memory organization: cache, virtual memory; channels; DMA and Interrupts; Buses; Multiprocessors: types of multiprocessors, performance, single bus multiprocessors, multiprocessors connected by network, clusters; Design using MSI and LSI components; Design of memory subsystem using SRAM and DRAM; Design of various components of a computer: ALU, memory and control unit - hardwired and micro programmed, Microprocessor based designs. Computer bus standards. Design using special-purpose controllers.

Course No. SWE- 312

Course Title: *Theory of Computing*
Number of Credit: 3

Language theory; Finite automata: deterministic finite automata, nondeterministic finite automata, equivalence and conversion of deterministic and nondeterministic finite automata, pushdown automata. Context-free languages; Context-free grammars; Turing Machines: basic machines, configuration, computing with turing machines, combining turing machines; Undecidability.

Automata Theory: The study of abstract machines (Finite Automata, Turing Machines) and the formal languages they can recognize.

Computability Theory: Identifying which problems can be solved by an algorithm and which are "undecidable" (e.g., the Halting Problem).

Complexity Theory: Classifying problems by the resources (time and memory) required to solve them, famously centered on the P vs. NP question.

Course No. SWE-313
Course Title: *.NET Programming with Lab*
Number of Credit: 4

Develop complex programs using the .NET framework classes. Understand threading and thread-related issues and implementations. Introduction to Windows forms, Advanced Windows forms, Develop applications that use custom events and delegates. Create programs that run on mobile platforms (phones and PDAs). Develop programs that use MDI forms. Create custom controls. Use GDI+ to draw controls. Develop programs using the C# programming language. Develop programs using the Visual Basic .NET programming language. Create multi-language programs. Use .NET class libraries for collections, input and output, and encryption.

Course No. SWE -321
Course Title: *Data Communication with Lab*
Number of Credit: 3

This course introduces the fundamental concepts and standards of data communication and computer networks. Topics include data communication architectures and models, network standards, and communication protocols with emphasis on the OSI reference model. Network topologies and examples of data communication networks are discussed.

Data Transmission:
Analog and digital data transmission, signal spectrum and bandwidth, transmission impairments, channel capacity, and performance limitations. Transmission media including twisted-pair cables, coaxial cables, optical fiber, and wireless communication. Wireless transmission concepts

covering the electromagnetic spectrum, radio, microwave, infrared, mobile and satellite communication systems. Overview of digital communication technologies including ISDN, Broadband ISDN (B-ISDN), and Asynchronous Transfer Mode (ATM).

Data Encoding and Modulation:
Digital data and digital signaling techniques including NRZ-L, NRZ-I, bipolar AMI, Manchester, and differential Manchester encoding. Digital data with analog signaling methods such as ASK, FSK, PSK, and QPSK, along with performance considerations. Analog data with digital signaling using Pulse Code Modulation (PCM).

Data Communication Techniques:
Asynchronous and synchronous transmission methods. Error detection techniques including parity checking and cyclic redundancy check (CRC). Serial communication standards such as RS-232 (EIA-232) and V.24 interface standards.

Data Link Control and Multiplexing:
Flow control mechanisms and error control protocols including Stop-and-Wait ARQ, Go-Back-N ARQ, and Selective Repeat ARQ. High-Level Data Link Control (HDLC). Multiplexing techniques including Frequency Division Multiplexing (FDM), Synchronous Time Division Multiplexing (TDM), and Statistical Time Division Multiplexing.

Switching and Packet Communication:
Circuit switching and packet switching principles. Space-division and time-division switching. Single-node networks and digital PBX systems. Virtual circuits and datagram services, routing concepts, traffic control, and packet-switched networks including the X.25 standard.

Laboratory Component:
The laboratory sessions reinforce theoretical concepts through hands-on experiments involving data transmission, signal encoding, error detection, multiplexing, and basic network configuration and analysis. Students gain practical experience with communication protocols, performance measurement, and network simulation or hardware-based experiments.

Course No. SWE- 322
Course Title: *Software Security*
Number of Credit: 3

Syllabus: system security requirements; new technologies used to build software (Networking, Distributed Systems, Mobile Code, and Commercial off-the-shelf components), Security Policy- Risk Assessment, Identify assets, Identify potential enemies, Identify needed solutions, Password Policies, Administrator Responsibilities, User Responsibilities, Email Policies, Internet Policies, Disaster Recovery; Security Requirements; Unifying Security and System Models, Formal Security Models, Multilevel Security Model, Advancing Security Models; Security Engineering; Software Piracy, Piracy Economics; Dynamic Decryption of Code; Watermarking; Code Partitioning; Attacker Cost Models; Black box Approaches; Grey-box Verification;

Course No. SWE-323
Course Title: *Systems Analysis & Design*
Number of Credit: 3

System concept and information system environment; system development; Role of system analyst; system planning and initial investment; information gathering; tools of structured analysis; feasibility study; cost/ benefit analysis; process and stages of system design; input/output form design; file organization and database design; system testing and quality assurance; implementation and software maintenance.

Modern Methodologies: Heavy emphasis on Agile, Scrum, and Object-Oriented Analysis (OOA).

Visual Modeling: Practical training in UML (Unified Modeling Language) and CASE tools to create system blueprints.

Core Skills: Focuses on Requirement Engineering (defining what a system must do), database design, and user interface (UI) planning.

Interdisciplinary Nature: Combines computer science with business strategy, project management, and emerging fields like AI integration.

Course No. SWE -331
Course Title: *Object-oriented Software Development (Lab-Based)*
Number of Credit: 3

Object-Oriented Software Development (OOSD) courses focus on building modular, reusable, and maintainable code through a "learning-by-doing" approach. These courses are structured to bridge the gap between theoretical computer science and professional software engineering.

Fundamental Concepts: Students master the "A-PIE" principles: Abstraction, Polymorphism, Inheritance, and Encapsulation.

Practical Lab Sessions: Students spend significant time in laboratories applying concepts through live coding, problem-solving portfolios, and weekly programming challenges.

Modeling & Design: Beyond writing code, labs emphasize software architecture using UML (Unified Modeling Language) to design class diagrams and system structures before implementation.

Industry Tools: Students gain experience with professional tools such as IDEs (e.g., IntelliJ, Eclipse), version control (Git), and automated testing frameworks (JUnit).

This lab is based on this course.

Course No. SWE-332
Course Title: ***Software Engineering Project-II (Web Programming)***
Number of Credit: 3

This course provides students with the opportunity to undertake a **supervised software engineering project** focused on **web programming**. Students apply theoretical knowledge and practical skills to design, develop, and evaluate a small-scale web-based software system using programming languages such as **Java, Python, or C#**.

Key Components:

- **Project Planning and Specification:** Defining project objectives, scope, requirements, and deliverables. Preparation of an **Independent Study Contract** outlining learning objectives, project overview, and assessment criteria.

- **Software Development:** Application of web programming concepts, software engineering methodologies, and best practices in the development of web applications. Use of appropriate frameworks, libraries, and development environments.

- **Project Activities:** Combination of literature review, system design, implementation, testing, and documentation. Emphasis on applying object-oriented programming, modular design, and web technologies (front-end and back-end integration).

- **Evaluation and Assessment:** Continuous supervision and feedback from faculty. Assessment based on project deliverables, technical report, code quality, demonstration of the working application, and oral presentation.

Laboratory / Practical Component:
Students engage in hands-on programming and implementation using **Java (Spring, JSP, Servlets), Python (Django, Flask), or C# (.NET, ASP.NET)**. Activities include coding, debugging, version control, database integration, and deployment of web applications. Emphasis is placed on applying **software engineering principles** in a real-world web development context.

Course No. SWE -333
Course Title: Desktop and Web Programming with Lab
Number of Credit: 4

This course introduces the **technologies, concepts, and techniques** required for developing **modern desktop and web applications**. Students gain practical and theoretical knowledge in designing, implementing, and deploying cross-platform software systems using contemporary tools and frameworks.

Desktop Application Development:

- Building cross-platform desktop software using **C#/.NET** or **Java/JavaFX**.
- Object-oriented design principles, GUI programming, event handling, and data persistence.
- Integration with databases and external APIs.

Web Application Development:

- Fundamentals of web servers, including **PWS** and **IIS**.
- Server-side programming with **Active Server Pages (ASP)** and database connectivity.
- Introduction to CGI programming with **Perl** and **PHP**.
- Modern web frameworks including **React**, **Node.js**, and **Python (Django/Flask)**.
- Concepts in e-commerce application development, RESTful APIs, and client-server communication.

Laboratory Component:

Hands-on lab sessions are **mandatory**, providing experience in:

- Designing, coding, and debugging desktop and web applications.
- Database management using **SQL and NoSQL systems**.
- Deployment and testing of live applications.
- Applying software engineering best practices, version control, and collaborative development.

Learning Approach:

The course emphasizes **practical, project-based learning** with integration of theoretical concepts. Students develop real-world applications while following software development life cycles, ensuring understanding of both **design and implementation** aspects.

Course No. SWE-412
Course Title: *Management Information System*
Number of Credit: 3

This course provides the principles, theories, concepts, applications, strategies, and issues in the technology and information management that are essential for business leaders as well as entrepreneurs. This course provides the basic knowledge and enhances understanding in information management. It is a highly practical course that stresses the strategic importance of information and knowledge in the business context. This course increases confidence and understanding in IT-related matters; develops skills in analyzing real-world cases, particularly in the context of business strategy, competitors, and the industry. This course also incorporates relevant topics for business leaders or entrepreneurs to enhance decision-making and problem-solving skills; from the structured day-to-day operational decisions to the unstructured strategic decisions. By the end of this course, participants will be able to fully understand the importance of information in business and how information management is critical for creating competitive firms, managing global corporations, and providing useful products and services to customers. Ultimately, this course provides practical insights on how and why value is created by businesses for sustainability.

Apply fundamental concepts and tools of technology and information management in work tasks and organizations, Analyze business-related issues in the context of information systems in global business today, Assess global business environment and utilize technology and information for organizational sustainability, Utilize technology and information management to attain strategic business objectives, and Develop skills in technology and information management for personal development and managerial decision-making.

Course No. SWE-413
Course Title: Software *Engineering and Cyber Laws*
Number of Credit: 3

This course provides a comprehensive overview of the **legal, ethical, and regulatory frameworks** relevant to software engineering and information technology. Students develop an understanding of **professional responsibilities, intellectual property, and cyber law** in both national and international contexts.

Key Topics:

- **Sources and Classification of Law:** Overview of legislation, regulations, and case law relevant to software engineering and IT professionals.
- **Professional Practice and Ethics:** Engineering legislation, codes of ethics, professional registration, disciplinary processes, and negligence.
- **Contract and Employment Law:** Principles of contract law, employment agreements, liability, and compliance in software development and IT projects.
- **Intellectual Property (IP) Law:** Patents, copyrights, trademarks, industrial designs, and their management. Procedures for filing and enforcement of IP rights. Commercialization and licensing of intellectual property. University policies and students' IP rights.
- **Environmental and Safety Laws:** Legal requirements affecting software and IT industries, including compliance and risk management.
- **Cyber Law and Internet Regulation:** Legal aspects of electronic commerce, data privacy, cybersecurity, and digital transactions. Emerging trends in international cyber law.

Learning Approach:

The course combines **lectures, case studies, and practical discussions**. Students analyze real-world scenarios involving IP, contracts, and cyber law, and examine ethical dilemmas in software engineering practice.

Course No. SWE-421

Course Title: *Multimedia Programming with Lab*

Number of Credit: 4

This course introduces the **concepts, tools, and techniques for multimedia programming**, combining **creative design with technical implementation**. Students learn to develop interactive digital media applications integrating text, graphics, audio, video, and animation.

Key Topics:

- **Programming Fundamentals for Multimedia:** Variables, control structures, loops, functions, and object-oriented concepts applied to media programming.
- **Graphics and Animation:** Drawing shapes, manipulating pixels, creating animations, and handling interactive graphical elements.
- **Audio and Video Processing:** Importing, processing, and playing audio and video streams. Basic sound synthesis, mixing, and manipulation.
- **Integration of Multimedia Components:** Combining graphics, audio, and video into cohesive interactive applications, such as educational software, games, and simulations.
- **Tools and Languages:** Hands-on experience using **MATLAB, C++, Python**, or specialized multimedia programming environments and libraries.

Laboratory Component:
Laboratory sessions provide **practical, project-based experience**, where students implement multimedia applications. Activities include coding, debugging, testing interactive programs, processing images and audio, and integrating multiple media elements into functioning software. Emphasis is placed on **both technical proficiency and creative design skills**.

Learning Approach:
The course emphasizes **experiential learning**, encouraging students to develop **interactive applications** that demonstrate integration of **artistic creativity and programming expertise**.

Course No. SWE - 422
Course Title: *Numerical Analysis with Lab*
Number of Credit: 3

Numerical solution of systems of linear and non-linear equations. Numerical optimization techniques including direct search methods and gradient-based methods. Matrix computations with emphasis on eigenvalue and eigenvector determination using direct iteration, inverse iteration, and shift of origin, with applications to vibration and stability problems. Numerical methods for handling banded and sparse matrices arising in large-scale engineering systems.

Numerical solution of ordinary and partial differential equations using finite difference methods for initial-value and boundary-value problems. Phase plane analysis and isocline curves. Series-based methods including Taylor series approximations. Time-stepping methods such as explicit and implicit schemes, including Euler and Runge–Kutta methods. Introduction to numerical errors, convergence, consistency, and stability analysis. Introductory concepts of the method of characteristics and the finite element method.

Computational Laboratory Component (Python):
Implementation of numerical algorithms using Python-based scientific computing tools. Practical use of libraries such as NumPy, SciPy, and Matplotlib for matrix operations, eigenvalue problems, numerical optimization, and solution of differential equations. Emphasis on algorithm implementation, numerical experimentation, visualization of results, performance analysis, and validation of numerical solutions through error and stability assessment.

Course Code: SWE-423
Course Title*: Advanced Database Management System with Lab*
Number of Credit: 4

Query Processing and Optimization; Transactions; Concurrency Control; Recovery System; Database System Architectures; Parallel Databases; Distributed Database; Special Topics: Security and integrity, Standardization, Performance benchmarks, Performance tuning, Time in database, User Interfaces, Active database; Advanced Transaction Processing: Remote backup systems, Transaction-processing monitors, High-performance transaction systems, Long-duration transactions, Real-time transaction systems, Weak level of consistency, Transaction workflows; New Applications: Decision-support systems, Data analysis, Data mining, Data warehousing, Spatial and geographic databases, multimedia databases, Mobility and personal databases, Information-retrieval systems, Distributed information systems, The world wide web; Network Model: Data-structure diagrams, The DBTG CODASYL model; Hierarchical Model: Tree - structure diagrams, The IMS database system.

Course No. SWE- 424
Course Title: *Artificial Intelligence with Lab*
Number of Credit: 4

Syllabus: Survey of concepts in artificial intelligence. Knowledge representation, search and control techniques. AI machines and features of Python language. Problem representation, search and constraint propagation, rule chaining, frame inheritance, inference and learning in intelligent systems; systems for general problem solving, game playing, expert consultation, concept formation and natural languages processing; recognition, understanding and translation. Machine Learning Approaches. Use of heuristic vs. algorithmic programming; cognitive simulation vs. machine intelligence; study of some expert systems such as robotics and understanding. Solving problem in AI languages.

This lab is based on this course.

Course No. SWE-425
Course Title: *Telecommunication Engineering with Lab*
Number of Credit: 4

Data Communications, telephony, and microwave transmission, high efficiency transmission of analog and transmission paths, radio link systems, earth station criteria, and facsimile communications, FDMA, TDMA, and CDMA, associated standard hierarchies and applications.

Signals & Systems Lab: Hardware implementation of signal processing.

Antennas & Microwaves Lab: Designing and testing physical radio frequency (RF) components.

Networking Lab: Configuring physical routers, switches, and software-defined networks (SDN).

Optical Communications Lab: Experiments with fiber optics and laser transmission.

This lab is based on this course.

Course Code: SWE-426
Course Title: *Distributive Computing & Networking Security with Lab*
Number of Credit: 4

Distributed systems, operating system structure, process management, interaction between system components (processes, devices, and processors), mutual exclusion, concurrent programming, semaphores and monitors, inter-process communication, distributed systems, crash resilience and persistent data, deadlock, transaction processing.

Security challenges and requirements; security management; symmetric key cryptography (+ DES); public key cryptography (+ RSA); one-way hash functions and digital signatures; secret key distribution (Diffie-Hellman key exchange); public key infrastructure (X.509); network authentication protocols (Kerberos); electronic mail security (PGP); IP security (IPSec V4 V6); web security (SSL, SET); system security; and selected topics (emerging technologies).

This lab is based on this course.

Course No. SWE- 435
Course Title: *Business communication*
Number of Credit: 3

This course aims to help you develop the proficiency needed to succeed in today's technologically enhanced workplace by focusing on the development of professional oral and written communication skills. Having good oral communication and writing skills is imperative in today's workplace. In the past, business people may have written a couple business letters a month, but now they can receive and send hundreds of email messages weekly. Their writing skills are showcased in every message they send. As well, with the arrival of videoconferencing, large-scale meetings are possible with increased frequency. Through this course, you will improve your: memo, letter, email, report, policy and proposal writing; presentation skills; participation in and chairing of meetings; and job application strategies.

Course No. SWE-411
Course Title: *Computer Network with Lab*
Number of Credit: 4

A Computer Network Lab provides hands-on experience with network setup, configuration, protocol analysis, troubleshooting, and security using real hardware (routers, switches, servers) and software (Wireshark, Cisco Packet Tracer, NS2) to understand everything from basic LANs to complex protocols like TCP/IP, HTTP, DNS, routing, and network programming. These labs bridge theory with practice, enabling students to build, manage, and test modern network infrastructures.

Course No. SWE- 439

Course Title: *Project/Thesis (Internship included)*
Number of Credit: 3

This course represents the capstone of the undergraduate program and integrates an industry internship with an academic project or thesis. Students undertake an independent project involving problem identification, literature review, system design or theoretical analysis, implementation, experimentation, and evaluation under academic supervision.

The internship component provides practical exposure to real-world software engineering, information technology, or applied computing environments. Students apply theoretical knowledge, professional practices, and ethical standards in an organizational setting. The academic project or thesis is closely aligned with the internship or an approved research problem and requires the application of appropriate methodologies, tools, and technologies.

The course culminates in the submission of a written project report or thesis and an oral presentation or defense. Emphasis is placed on independent learning, research skills, technical writing, and professional communication.