



**Uttarakhand Technical University, Dehradun**  
**Scheme of Examination as per AICTE Flexible Curricula**  
**Evaluation Scheme & Syllabus for B. Tech Second Year**

**W.E.F. Academic Session 2019-20**  
**II Year (III & IV SEMESTER)**

**Bachelor of Technology (B. Tech.)**  
**[Computer Science & Engineering]**

Uttarakhand Technical University, Dehradun  
As per AICTE model curriculum

[W.E.F. Academics Session: 2019-20]  
B. Tech. II Year (Computer Science & Engineering)

**Uttarakhand Technical University, Dehradun**  
**New Scheme of Examination as per AICTE Flexible Curricula**  
**Bachelor of Technology (B.Tech.) [Computer Science and Engineering]**  
**W.E.F. Academic Session – 2019-20**

**III Semester**

S. No.	Subject Code	Category	Subject Name	Maximum Marks Allotted					Total Marks	Contact Hours per week			Total Credits
				Theory			Practical			L	T	P	
				End Sem.	Mid Sem.	Quiz / Assignment	End Sem.	Lab work & Sessional					
1.	BCET301	ESC	Energy & Environmental Engineering	100	30	20	-	-	150	3	1	-	4
2.	BCST-302	DC-1	Discrete Structure	100	30	20	-	-	150	3	1	-	4
3.	BCST-303 BCSP 303	DC-2	Data Structure	100	30	20	30	20	200	3	1	2	5
4.	BECT-303 BECP 303	DC-3	Digital Electronics	100	30	20	30	20	200	3	1	2	5
5.	BCST-305 BCSP 305	DC-4	Object Oriented Programming & Methodology	100	30	20	30	20	200	3	1	2	5
6.	BCSP-306	DLC-3	Computer Workshop (Using Python)	-	-	-	30	20	50	-	-	2	1
7.	BASP 107	DLC-1	Evaluation of Internship-I completed at I year level	-	-	-	-	50	50			2	1
8.	BASP 307	DLC-1	90 hrs Internship based on using various software's- internship -II	To be completed anytime during Third/fourth semester. Its evaluation/credit to be added in fifth semester.									
Total				500	150	100	120	130	1000	15	5	10	25
NSS/NCC													

**IV Semester**

S. No	Subject Code	Category	Subject Name	Maximum Marks Allotted					Total Marks	Contact Hours per week			Total Credits
				Theory			Practical			L	T	P	
				End Sem.	Mid Sem. Exam.	Quiz/ Assignment	End Sem.	Term work Lab Work & Sessional					
1.	BAST 401	BSC	Mathematics- III	100	30	20	-	-	150	3	1	-	4
2.	BCST 402 BCSP 402	DC	Database Management Systems	100	30	20	30	20	200	3	1	2	5
3.	BCST 403 BCSP 403	DC	Software Engineering	100	30	20	30	20	200	3	-	2	4
4.	BCST 404 BCSP 404	DC	Computer Org. & Architecture	100	30	20	30	20	200	3	1	2	5
5.	BCST 405 BCSP 405	DC	Theory of Automata and Formal Languages	100	30	20	30	20	200	3	1	2	5
6.	BCSP 406	DLC	Programming Practices (Introduction to MATLAB)	-	-	-	30	20	50	-	-	4	2
7.	BCSP 407	DLC	90 hrs Internship based on using various software's –Internship -II	To be completed anytime during fourth semester. Its evaluation/credit to be added in fifth semester.									3
Total				500	150	100	150	100	1000	15	4	8	25
8.	BCST 408	MC	Cyber Security	Non-credit course									
NSS/NCC													

\*A minimum of 2 hours per week should be allotted for the Virtual Lab along with the slot fixed for the conventional lab classes.  
MST: Minimum of two mid semester tests to be conducted.

1 Hr Lecture	1 Hr Tutorial	2 Hr Practical
1 Credit	1 Credit	1 Credit

## Semester -III

BCET 301	Energy and Environmental Engineering	3L:1T:0P	4 Credits
----------	--------------------------------------	----------	-----------

### Course Objectives:

The objective of this course is to apply knowledge of mathematics, science, technology and engineering appropriate to energy science and engineering degree discipline and to enhance the understanding of conventional and non-conventional energy sources and its relationship with the ecology and environment. More precisely the objectives are:

1. Use mathematical or experimental tools and techniques relevant to the energy and energy-related environmental disciplines along with an understanding of their processes and limitations.
2. Equip the students with knowledge and understanding of various possible mechanisms about renewable energy projects
3. To produce graduates strong in understanding on energy resources, technologies and systems, energy management fundamentals, and capable in innovative technological intervention towards the present and potential future energy.
4. To identify, formulate and solve energy and energy-related environmental problems by pursuing development of innovative technologies that can generate clean and sustainable energy to address energy scarcity and combat pollution and climate change.

### Course Outcomes

1. Apply advanced level knowledge, techniques, skills and modern tools in the field of Energy and Environmental Engineering.
2. Distinguish the different energy generation systems and their environmental impacts.
3. Respond to global policy initiatives and meet the emerging challenges with sustainable technological solutions in the field of energy and environment.

### Detailed Content

#### Unit I:

**Introduction to Energy Science** - Introduction to energy systems and resources; Introduction to Energy, sustainability & the environment, Global Energy Scenario: Role of energy in economic development. Indian Energy Scenario: Introduction to Energy resources & Consumption in India. Common terminologies

#### Unit II

**Energy Sources** - Overview of energy systems, sources, transformations, efficiency, and storage. Fossil fuels (coal, oil, oil-bearing shale and sands, coal gasification) - past, present & future, Remedies & alternatives for fossil fuels - biomass, wind, solar, nuclear, wave, tidal and hydrogen; Sun as Source of Energy, Availability of Solar Energy, Nature of Solar Energy, Solar Energy & Environment. Various Methods of using solar energy. Commercial and noncommercial forms of energy, Fossil fuels, Renewable sources including: Nuclear Energy, Hydel Energy, Storage of Hydrogen, Hydrogen Production, Hydrogen Energy Geothermal, Tide and Wave Energy, Bio-fuels in India.

#### Unit III

Energy Efficiency and Conservation - Introduction to clean energy technologies and its importance in sustainable development; Carbon footprint, energy consumption and sustainability; introduction to the economics of energy; How the economic system determines production and consumption; linkages between economic and environmental

outcomes; How future energy use can be influenced by economic, environmental, trade, and Research policy.

#### **Unit IV**

**Energy & Environment** - Environment: Introduction, Multidisciplinary nature of environmental studies- Definition, scope and importance, Need for public awareness. Ecosystem: Concept, Energy flow, Structure and function of an ecosystem. Food chains, food webs and ecological pyramids, Forest ecosystem, Grassland ecosystem, Desert ecosystem and Aquatic ecosystems, Ecological succession. Environmental Pollution: Definition, Cause, effects and control measures of - Air pollution, Water pollution, Soil pollution, Marine pollution, Noise pollution, Thermal pollution and Nuclear hazards, solid waste Management.

#### **Unit V**

**Environmental Protection and Ethics** - Environmental Protection- Role of Government Initiatives by Non-governmental Organizations (NGO) Environmental Education. Ethics and moral values Objectives of ethics, Professional and Non-professional ethics Sustainable Development of the ecology and environment Codes of ethics and their limitations

#### **Suggested reading material:**

1. Schaeffer, John. 2007. Real Goods Solar Living Sourcebook: The Complete Guide to Renewable Energy Technologies and Sustainable Living (30th anniversary edition). Gaia.
2. Boyle, Godfrey, Bob Everett, and Janet Ramage (eds.) 2004. Energy Systems and Sustainability: Power for a Sustainable Future. Oxford University Press, 619 pages (ISBN: 0-19-926179-2)
3. Energy Management Principles: C.B.Smith (Pergamon Press)
4. Renewable Sources of Energy and Conversion Systems: N.K.Bansal and M.K.Kleeman.
5. Energy Management: W.R.Murphy, G.Mckay (Butterworths)
6. Ristinen, Robert A. Kraushaar, Jack J. AKraushaar, Jack P. Ristinen, Robert A. (2006) Energy and the Environment, 2nd Edition, John Wiley
7. Ravindranath, N. H., & Hall, D. O. (1995). Biomass, energy and environment: a developing country perspective from India. Oxford University Press.
8. Popp, D., Newell, R. G., & Jaffe, A. B. (2010). Energy, the environment, and technological change. In Handbook of the Economics of Innovation (Vol. 2, pp. 873-937). North-Holland.

<b>BCST 302</b>	<b>Discrete Structure</b>	<b>3L:1T: 0P</b>	<b>4 credits</b>
-----------------	---------------------------	------------------	------------------

### **Course Objectives:**

Throughout the course, students will be expected to demonstrate their understanding of Discrete Mathematics by being able to do each of the following:

1. Use mathematically correct terminology and notation.
2. Construct correct direct and indirect proofs.
3. Use division into cases in a proof.
4. Use counterexamples.
5. Apply logical reasoning to solve a variety of problems.

### **Course Outcomes:**

1. For a given logic sentence express it in terms of predicates, quantifiers, and logical connectives
2. For a given a problem, derive the solution using deductive logic and prove the solution based on logical inference
3. For a given a mathematical problem, classify its algebraic structure
4. Evaluate Boolean functions and simplify expressions using the properties of Boolean algebra
5. Develop the given problem as graph networks and solve with techniques of graph theory.

### **Unit 1: Sets, relations and functions:**

Basic operations on sets, Cartesian products, disjoint union (sum), and power sets. Different types of relations, their compositions and inverses. Different types of functions, their compositions and inverses.

### **Unit 2: Propositional Logic:**

Syntax and semantics, proof systems, satisfiability, validity, soundness, completeness, deduction theorem, etc. Decision problems of propositional logic. Introduction to first order logic and first order theory.

### **Unit 3: Partially ordered sets:**

Complete partial ordering, chain, lattice, complete, distributive, modular and complemented lattices. Boolean and pseudo Boolean lattices.

### **Unit 4: Algebraic Structures:**

Algebraic structures with one binary operation – semigroup, monoid and group. Cosets, Lagrange’s theorem, normal subgroup, homomorphic subgroup. Congruence relation and quotient structures. Error correcting code. Algebraic structures with two binary operations- ring, integral domain, and field. Boolean algebra and Boolean ring (Definitions and simple examples only).

### **Unit 5: Introduction to Counting:**

Basic counting techniques – inclusion and exclusion, pigeon-hole principle, permutation, combination, summations. Introduction to recurrence relation and generating functions.

### **Unit 6: Introduction to Graphs:**

Graphs and their basic properties – degree, path, cycle, subgraph, isomorphism, Eulerian and Hamiltonian walk, trees.

**Textbooks/References:**

1. C. L. Liu, Elements of Discrete Mathematics, 2nd Ed., Tata McGraw-Hill,2000.
2. R. C. Penner, Discrete Mathematics: Proof Techniques and Mathematical Structures, World Scientific,1999.
3. R. L. Graham, D. E. Knuth, and O. Patashnik, Concrete Mathematics, 2nd Ed., Addison-Wesley,1994.
4. K. H. Rosen, Discrete Mathematics and its Applications, 6th Ed., Tata McGraw-Hill, 2007.
5. J. L. Hein, Discrete Structures, Logic, and Computability, 3rd Ed., Jones and Bartlett, 2010.
6. N. Deo, Graph Theory, Prentice Hall of India,1974.
7. S. Lipschutz and M. L. Lipson, Schaum's Outline of Theory and Problems of Discrete Mathematics, 2nd Ed., Tata McGraw-Hill,1999.
8. J. P. Tremblay and R. P. Manohar, Discrete Mathematics with Applications to Computer Science, Tata McGraw-Hill,1997.

**Course Objectives:**

1. Understand and remember algorithms and its analysis procedure.
2. Introduce the concept of data structures through ADT including List, Stack, Queues.
3. To design and implement various data structure algorithms.
4. To introduce various techniques for representation of the data in the real world.
5. To develop application using data structure algorithms.
6. Compute the complexity of various algorithms.

**Course Outcomes:**

1. Select appropriate data structures as applied to specified problem definition.
2. Implement operations like searching, insertion, and deletion, traversing mechanism etc.on various data structures.
3. Students will be able to implement Linear and Non-Linear data structures.
4. Implement appropriate sorting/searching technique for given problem.
5. Design advance data structure using Non-Linear data structure.
6. Determine and analyze the complexity of given Algorithms.

**Unit 1:****Introduction:**

Basic Terminologies: Elementary Data Organizations, Data Structure, Operations: insertion, deletion, traversal etc.; Analysis of an Algorithm, Asymptotic Notations, Time-Space trade off. Searching: Linear Search and Binary Search Techniques and their complexity analysis.

**Unit 2:**

Stacks and Queues: ADT Stack and its operations: Algorithms and their complexity analysis, Applications of Stacks: Expression Conversion and evaluation-corresponding algorithms and complexity analysis. ADT queue, Types of Queue: Simple Queue, Circular Queue, Priority Queue; Operations on each types of Queues: Algorithms and their analysis.

**Unit 3:**

Linked Lists: Singly linked lists: Representation in memory, Algorithms of several operations: Traversing, Searching, Insertion into, Deletion from linked list; Linked representation of Stack and Queue, Header nodes, Doubly linked list: operations on it and algorithmic analysis; Circular Linked Lists: all operations their algorithms and the complexity analysis.

**Trees:** Basic Tree Terminologies, Different types of Trees: Binary Tree, Threaded Binary Tree, Binary Search Tree, AVL Tree; Tree operations on each of the trees and their algorithms with complexity analysis. Applications of Binary Trees. B Tree, B+ Tree: definitions, algorithms and analysis.

**Unit 4:**

**Sorting and Hashing:** Objective and properties of different sorting algorithms: Selection Sort, Bubble Sort, Insertion Sort, Quick Sort, Merge Sort, Heap Sort; Performance and Comparison among all the methods, Hashing.

**Graph:** Basic Terminologies and Representations, Graph search and traversal algorithms and complexity analysis.

**List of Programs:**

1. Implementation of Array ADT and String ADT
2. Programs for Stack, Queues and Circular Queues using Arrays
3. Program to convert an Infix Expression into Postfix and Postfix Evaluation
4. Program to implement a Singly Linked List
5. Programs to implement Stack & Queues using Linked Representation
6. Programs implement Double Linked List and Circular Linked List
7. Program for Polynomial Arithmetic using Linked List
8. Program to implement Hashing
9. Programs to implement Insertion Sort, Selection Sort, Heap Sort, and Shell Sort
10. Program to implement Quick Sort and Merge Sort
11. Programs to implement Tree Traversals on Binary Trees and Graphs Search Methods
12. Programs to implement operations on AVL Trees and Splay Trees

**Suggested books:**

1. “Fundamentals of Data Structures”, Illustrated Edition by Ellis Horowitz, Sartaj Sahni, Computer Science Press.
2. Data Structures, Schaum’s Series

**Suggested reference books:**

1. Algorithms, Data Structures, and Problem Solving with C++”, Illustrated Edition by Mark Allen Weiss, Addison-Wesley Publishing Company
2. Data Structures Using C and C++, Aaron Tenenbaum



<b>BECT 303 &amp; BECP 303</b>	<b>Digital Electronics</b>	<b>3L:1T: 2P</b>	<b>5 credits</b>
--------------------------------	----------------------------	------------------	------------------

**COURSE OBJECTIVE(s):**

The objectives of this course are to:

1. Introduce the concept of digital and binary systems
2. Be able to design and analyze combinational logic circuits.
3. Be able to design and analyze sequential logic circuits.
4. Understand the basic software tools for the design and implementation of digital circuits and systems.
5. Reinforce theory and techniques taught in the classroom through experiments and projects in the laboratory.

**COURSE OUTCOME(s):**

1. After successful completion of the course student will be able to
2. Develop a digital logic and apply it to solve real life problems.
3. Analyze, design and implement combinational logic circuits.
4. Classify different semiconductor memories.
5. Analyze, design and implement sequential logic circuits.
6. Analyze digital system design using PLD.
7. Simulate and implement combinational and sequential circuits.

**Course Contents:**

**Unit 1:**

**Number Systems Binary Codes:** Number System and its arithmetic, conversion between bases, Boolean algebra, Canonical form, SOP & POS forms, Minimization of Boolean Functions: K Map (upto 5 variables), Quine-Mcclusky method, Error detection & correcting codes, Hamming codes, Binary codes.

**Unit 2:**

**Combinational Logic Circuits:** Introduction to Combinational Circuits, Analysis and Design Procedure, Binary Adder, Subtractor, Parallel Adder/Subtractor, Carry Look Ahead Adder, Decoder, Encoder, Priority Encoder, Digital Multiplexer, Magnitude Comparator. Programmable Logic Devices, PLA & PAL.

**Unit 3:**

**Sequential Circuits Fundamentals:** Basic Architectural Distinctions between Combinational and Sequential circuits, Flip Flops: SR, JK, D and T Type, Timing and Triggering Consideration, JK Master Slave, Excitation Table of all Flip Flops, Conversion from one type of Flip-Flop to another.

**Counters:** Asynchronous and Synchronous Counters, Design of Up Counters, Design of Down Counters, Mod Counter, Lock-Out table, Self-Starting Counter.

**Unit 4:**

**Finite State Machine:** Mealy and Moore machines, State Table, State Diagram, Reduction of State Table, FSM Design Steps, Counter Design Using FSM.

**Logic Families:** Classification of Logic Families, Parameters: Propagation Delay, Power Dissipation, Fan-in, Fan-out, Noise Margin. TTL Family, TTL output configurations, ECL Family, IIL Family, MOS Family. Logic gate design using TTL and MOS.

**Unit 5:**

**Hazard, Fault Detection:** Hazard and Fault Detection, Static Hazards, Dynamic Hazards, Determination of Hazards in Combinational Circuits. Fault Detection Using Fault Table and Path Sensitizing Methods.

**Memories:** Sequential Access Memories, Random Access Memories, RAM, ROM, PROM, EPROM, EEPROM, Static and Dynamic RAM cells using nMOS CMOS, Memory Size Expansion.

**List of Experiments:**

1. Introduction to Digital Electronics lab- nomenclature of digital ICS.
2. Implementation of the given Boolean function using logic gates in both sop and pos forms.
3. Verification of state tables of RS, JK, T and D flip-flops using NAND & NOR gates.
4. Implementation and verification of decoder/de-multiplexer and encoder using logic gates.
5. Implementation of 4x1 multiplexer using logic gates.
6. Implementation of 4-bit parallel adder using 7483 IC.
7. To design and verify operation of half adder and full adder.
8. To design & verify the operation of magnitude comparator.
9. Design and verify the 4-bit synchronous counter.
10. Design and verify the 4-bit asynchronous counter.

**Textbooks/References:**

1. M. Morris Mano and M. D. Ciletti, Digital Fundamental, 4<sup>th</sup>, Edition, Pearson.
2. Digital Systems: Principles and Design, Raj Kamal, Pearson.
3. Fundamentals of Digital Circuits A. Anand Kumar PHI 3rd Edition, 2014.
4. Digital Fundamental, Thomas L Floyd, 11<sup>th</sup> Edition

<b>BCST 305 &amp; BCSP 305</b>	<b>Object Oriented Programming &amp; Methodology</b>	<b>3L:1T: 2P</b>	<b>5 credits</b>
------------------------------------	--	------------------	------------------

### **Objectives of the course:**

The course will introduce standard tools and techniques for software development, using object-oriented approach, use of a version control system, an automated build process, an appropriate framework for automated unit and integration tests.

### **Course Outcomes:**

After taking the course, students will be able to:

1. Specify simple abstract data types and design implementations, using abstraction functions to document them.
2. Recognise features of object-oriented design such as encapsulation, polymorphism, inheritance, and composition of systems based on object identity.
3. Name and apply some common object-oriented design patterns and give examples of their use.
4. Design applications with an event-driven graphical user interface.

### **Course contents:**

#### **Unit I**

**Object Modelling:** Objects and classes, links and association, generalization and inheritance, aggregation, abstract class, multiple inheritance, meta data, candidate keys, constraints. **Dynamic Modelling:** Events and states, operations, nested state diagrams and concurrency, advanced dynamic Modelling concepts, a sample dynamic model.

#### **Unit II**

**Functional Modelling:** Data flow diagram, specifying operations, constraints, a sample functional model. OMT (object Modelling techniques) methodologies, examples and case studies to demonstrate methodologies, comparisons of methodologies, SA/SD, JSD.

#### **Unit III**

**Java Programming:** Introduction, Operator, Data types, Variables, Methods & Classes, Multithread Programming, I/O, Java Applet.

#### **Unit IV**

**Java Library:** String Handling, Input / Output exploring Java.io, Networking, Exception Handling, Event Handling, Introduction to AWT, Working with window, Graphics, AWT Controls, Layout Manager and Menus, Images.

#### **Unit V**

##### **Software Development using Java:**

Java Swing, Migrating from C++ to java, Application of java, JDBC.

### **Experiments**

1. To write a Java program to print HELLO INDIA.
2. To write a java program that takes in command line arguments as input and print the number of arguments.
3. To write a java program find the division of student.
4. To write a program implements the concept of inheritance.

5. To write a java program method overloading.
6. To write a java program for method over riding.
7. To write a java program exception handling.
8. To write a java program to run applet for drawing various shapes.
9. To write a java program to design a login using JFrame.
- 10.To write a java program to validate the logging details of user using JDBC concept.
- 11.To write a Java program Insertion sort.
- 12.To write a Java program merge sort.
- 13.To write a Java program first n prime numbers.

***Text Books:***

1. Herbert Schildt, “The Complete Reference: Java”, TMH, 7th Edition.
2. E. Balagurusamy, “Programming in JAVA”, TMH, 4th Edition.
3. James Rumbaugh et al, “Object Oriented Modelling and Design”, PHI
4. Barbara Liskov, *Program Development in Java*, Addison-Wesley, 2001

***References:***

1. Bjarne Stroustrup, “C++ Programming Language”, Addison Wesley, 3rd Edition.
2. E. Balagurusamy, “Object Oriented Programming with C++”, TMH, 2008.

<b>BCSP-306</b>	<b>Computer Workshop (Python Programming)</b>	<b>0L:0T:2P</b>	<b>1 credit</b>
-----------------	---	-----------------	-----------------

1. Installation & working with IDE
2. How to declare and use variables and operators
3. Programming using Basic Libraries (Numpy, Pandas, SK Learn etc)
4. To write a Python program to print HELLO INDIA.
5. To write a Python program that takes in command line arguments as input and print the number of arguments.
6. To write a Python program find the division of student.
7. To write a program implements Fibonacci series.
8. To write a Python program for factorial.
9. To write a Python program to use of functions.
10. To write a Python program to implement list.
11. To write a Python program to implement tuples.
12. To write a Python program Insertion sort.
13. To write a Python program merge sort.
14. To write a Python program first n prime numbers.
15. Implementation of Data Science concepts using Python

## Semester - IV

<b>BAST 401</b>	<b>Mathematics – III</b>	<b>3L-1T-0P</b>	<b>4 Credits</b>
-----------------	--------------------------	-----------------	------------------

**Prior Learning:** Students Should have the knowledge of Mathematics I and Mathematics II

### Course Objective:

The objective of this course is to familiarize the students with Laplace Transform, Fourier Transform, techniques in numerical methods & some statistical techniques. It aims to present the students with standard concepts and tools at B.Tech first year to superior level that will provide them well towards undertaking a variety of problems in the concern discipline.

The students will learn:

- The idea of Laplace transform of functions and their applications.
- The idea of Fourier transform of functions and their applications.
- To evaluate roots of algebraic and transcendental equations.
- Interpolation, differentiation, integration and the solution of differential equations.
- The basic ideas of statistics including measures of central tendency, correlation, regression and their properties.

### COURSE OUTCOMES(s):

At the end of this course, the students will be able to:

1. Remember the concept of Laplace transform and apply in solving real life problems.
2. Understand the concept of Fourier transform to evaluate engineering problems
3. Understand to evaluate roots of algebraic and transcendental equations.
4. Understand interpolation, differentiation, integration and the solution of differential equations.
5. Understand the concept of correlation, regression, moments, skewness and kurtosis and curve fitting.

### Unit 1: Fourier Transforms: (8 hours)

Fourier integral, Fourier Transform, Complex Fourier transform, Inverse Transforms, Convolution Theorem, Fourier sine and cosine transform, Applications of Fourier transform to simple one dimensional heat transfer equations.

### Unit 2: Laplace Transform: (8 hours)

Definition of Laplace transform, Existence theorem, Laplace transforms of derivatives and integrals, Initial and final value theorems, Unit step function, Dirac- delta function, Laplace transform of periodic function, Inverse Laplace transform, Convolution theorem, Application to solve linear differential equations.

### Unit 3: Solution of Algebraic and Transcendental equations & Interpolation (8 hours)

Number and their accuracy, Solution of algebraic and transcendental equations: Bisection method, Iteration method, Newton-Raphson method and Regula-Falsi method. Rate of convergence of these methods (without proof),

Interpolation: Finite differences, Relation between operators, Interpolation using Newton's forward and backward difference formula. Interpolation with unequal intervals: Newton's divided difference and Lagrange's formula.

**Unit 4: Numerical differentiation, Integration & Solution of ODE (8 hours)**

Numerical Differentiation, Numerical integration: Trapezoidal rule, Simpson's 1/3rd and 3/8 rule Runge- Kutta method of fourth order for solving first order linear differential equations. Milne's predictor-corrector method.

**Unit 5: Statistical Techniques (8 hours)**

Introduction: Measures of central tendency, Moments, Moment generating function (MGF) , Skewness, Kurtosis, Curve Fitting : Method of least squares, Fitting of straight lines, Fitting of second degree parabola, Exponential curves. Correlation and Rank correlation, Regression Analysis: Regression lines of y on x and x on y, regression coefficients, properties of regressions coefficients and non-linear regression.

**Reference Books:**

1. E. Kreyszig: Advanced Engineering Mathematics; John Wiley & Sons
2. B.V. Ramana: Higher Engineering Mathematics; Tata McGraw- Hill Publishing Company Limited, New Delhi.
3. Peter V.O' Neil. Advanced Engineering Mathematics, Thomas ( Cengage) Learning
4. B.S. Grewal, Higher Engineering Mathematics, Khanna Publishers, 35th Edition, 2000.
5. T.Veerarajan : Engineering Mathematics (for semester III), Tata McGraw-Hill, New Delhi.
6. R.K. Jain and S.R.K. Iyenger: Advance Engineering Mathematics; Narosa Publishing House, New Delhi.
7. P. Kandasamy, K. Thilagavathy, K. Gunavathi, Numerical Methods, S. Chand & Company, 2nd Edition, Reprint 2012.
8. S.S. Sastry, Introductory methods of numerical analysis, PHI, 4th Edition, 2005.
9. N.P. Bali and Manish Goyal, Computer Based Numerical and Statistical Techniques , Laxmi Publications, Reprint, 2010.
10. J.N. Kapur: Mathematical Statistics; S. Chand & Sons Company Limited, New Delhi.
11. D.N.Elhance,V. Elhance & B.M. Aggarwal: Fundamentals of Statistics; Kitab Mahal Distributers, New Delhi.

<b>BCST 402 &amp; BCSP 402</b>	<b>Database Management System</b>	<b>3L:1T:2P</b>	<b>5 credits</b>
--------------------------------	-----------------------------------	-----------------	------------------

### Course Objectives:

1. To understand the different issues involved in the design and implementation of a database system.
2. To study the physical and logical database designs, database modeling, relational, hierarchical, and network models
3. To understand and use data manipulation language to query, update, and manage a Database
4. To develop an understanding of essential DBMS concepts such as: database security, integrity, concurrency, distributed database, and intelligent database, Client/Server (Database Server), Data Warehousing.
5. To design and build a simple database system and demonstrate competence with the fundamental tasks involved with modeling, designing, and implementing a DBMS.

### Course Outcomes

1. For a given query write relational algebra expressions for that query and optimize the developed expressions
2. For a given specification of the requirement design the databases using ER method and normalization.
3. For a given specification construct the SQL queries for Open source and Commercial DBMS -MYSQL, ORACLE, and DB2.
4. For a given query optimize its execution using Query optimization algorithms
5. For a given transaction-processing system, determine the transaction atomicity, consistency, isolation, and durability.
6. Implement the isolation property, including locking, time stamping based on concurrency control and Serializability of scheduling

### UNIT-1:

**Database system architecture:** Data Abstraction, Data Independence, Data Definition Language (DDL), Data Manipulation Language (DML).

**Data models:** Entity-relationship model, network model, relational and object-oriented data models, integrity constraints, data manipulation operations.

### UNIT-2:

**Relational query languages:** Relational algebra, Tuple and domain relational calculus, SQL3, DDL and DML constructs, Open source and Commercial DBMS - MYSQL, ORACLE, DB2, SQL server.

**Relational database design:** Domain and data dependency, Armstrong's axiom, Normal forms, Dependency preservation, Lossless design.

**Query processing and optimization:** Evaluation of relational algebra expressions, Query equivalence, Join strategies, Query optimization algorithms.

### UNIT-3:

**Transaction processing:** Concurrency control, ACID property, Serializability of scheduling, Locking and timestamp-based schedulers, Multi-version and optimistic Concurrency Control schemes, Database recovery.



**UNIT-4:**

**Database Security:** Authentication, Authorization and access control, DAC, MAC and RBAC models, Intrusion detection, SQL injection.

**UNIT-5:**

**Advanced topics:** Object oriented and object relational databases, Logical databases, Web databases, Distributed databases, Data warehousing and data mining.

**List of Experiments:**

1. Write the queries for Data Manipulation and Data Definition Language.
2. Write SQL queries using logical operations and operators.
3. Write SQL query using group by function.
4. Write SQL queries for group functions.
5. Write SQL queries for sub queries, nested queries.
6. Write programme by the use of PL/SQL.
7. Write SQL queries to create views.
8. Write an SQL query to implement JOINS.
9. Write a query for extracting data from more than one table.
10. Write a query to understand the concepts for ROLL BACK, COMMIT & CHECK POINTS.

**REFERENCES:**

1. "Database System Concepts", 6th Edition by Abraham Silberschatz, Henry F. Korth, S. Sudarshan, McGraw-Hill.
2. "Principles of Database and Knowledge – Base Systems", Vol 1 by J. D. Ullman, Computer Science Press.
3. "Fundamentals of Database Systems", 5th Edition by R. Elmasri and S. Navathe, Pearson Education
4. "Foundations of Databases", Reprint by Serge Abiteboul, Richard Hull, Victor Vianu, Addison-Wesley

<b>BCST 403 &amp; BCSP 403</b>	<b>Software Engineering</b>	<b>3L:0T:2P</b>	<b>4 credits</b>
--------------------------------	-----------------------------	-----------------	------------------

**Course Objectives:**

1. To enable the students to apply a systematic application of scientific knowledge in creating and building cost effective software solutions to business and other types of problems.
2. To make the students understand project management concepts & their metrics.
3. To make the students understand requirement engineering and its models (Information, functional, behavioral).
4. Making the students understand to develop quality software, its maintenance & introduce about software reliability.

**Course Outcomes:**

The student will be able to

1. Implement Software life cycle models and have a knowledge of different phases of Software life cycle
2. Identify, formulate, review, estimate and schedule complex software projects using principles of mathematics.
3. Create a bug free software with good design and quality by using appropriate techniques and modern engineering and IT tools.
4. Analyze verification, validation activities, static, dynamic testing, debugging tools and techniques and importance of working in teams.

**Detailed Syllabus:**

**UNIT-1: INTRODUCTION**

Evolving role of software, Software Characteristics, Software crisis, Silver bullet, Software myths, Software process, Personal Software Process (PSP), Team Software Process (TSP), emergence of software engineering, Software process, project and product, Software Process Models: Waterfall Model, Prototype Model, Spiral, Model, RAD Model, Iterative Model, Incremental Model, Aspect-oriented Model, Agile Model.

**UNIT-2: SOFTWARE PROJECT MANAGEMENT**

Project management concepts, Planning the software project, Estimation—LOC based, FP based, Use-case based, empirical estimation COCOMO- A Heuristic estimation techniques, staffing level estimation, team structures, staffing, risk analysis and management.

**UNIT-3: REQUIREMENTS, ANALYSIS AND SPECIFICATION**

Software Requirements engineering, Requirement engineering process, Requirement Engineering Tasks, Types of requirements, SRS. System Modelling: Data Modelling, Functional Modelling and information flow: Data flow diagrams, Behavioral Modelling, The mechanics of structured analysis: Creating entity/ relationship diagram, data flow model, control flow model, the data dictionary.

#### **UNIT-4: SYSTEM DESIGN**

Design principles, the design process; Design concepts: Abstraction, refinement, modularity, software architecture, control hierarchy, structural partitioning, data structure, software procedure, information hiding; Effective modular design: Functional independence, Cohesion, Coupling;

#### **UNIT-5: SOFTWARE TESTING AND MAINTENANCE**

Testing terminology- error, bug/defect/fault, failure, Verification and validation, Test case design, Static testing, Dynamic testing--- Black box testing—Boundary value analysis, White box testing-- basis path testing, Unit testing, Integration testing, Acceptance Testing

#### **UNIT-6: SOFTWARE QUALITY MODELS AND STANDARDS**

Quality concepts, Software quality assurance, SQA activities, Formal approaches to SQA; Statistical software quality assurance; CMM, The ISO 9126 Standard

#### **List of Experiments**

1. Phases in software development project, overview, need, coverage of topics
2. To assign the requirement engineering tasks
3. To perform the system analysis : Requirement analysis, SRS
4. To perform the function oriented diagram : DFD and Structured chart
5. To perform the user's view analysis : Use case diagram
6. To draw the structural view diagram : Class diagram, object diagram
7. To draw the behavioral view diagram : Sequence diagram, Collaboration diagram
8. To draw the behavioral view diagram : State-chart diagram, Activity diagram
9. To draw the implementation view diagram: Component diagram
10. To draw the environmental view diagram : Deployment diagram
11. To perform various testing using the testing tool unit testing, integration testing.

#### **REFERENCES:**

1. Software Engineering – A Practitioner's Approach, Roger S. Pressman, 1996, MGH.
2. Fundamentals of software Engineering, Rajib Mall, PHI
3. Software Engineering by Ian Sommerville, Pearson Edu, 5th edition, 1999, AW,
4. Software Engineering – David Gustafson, 2002, T.M.H
5. Software Engineering Fundamentals Oxford University, Ali Behforooz and Frederick J. Hudson 1995, JW&S,
6. An Integrated Approach to software engineering by Pankaj Jalote , 1991 Narosa

<b>BCST 404 &amp; BCSP 405</b>	<b>Computer Organization &amp; Architecture</b>	<b>3L:1T:2P</b>	<b>5 credits</b>
--------------------------------	---	-----------------	------------------

### **Objectives of the course:**

To expose the students to the following:

1. How Computer Systems work & the basic principles
2. Instruction Level Architecture and Instruction Execution
3. The current state of art in memory system design
4. How I/O devices are accessed and its principles.
5. To provide the knowledge on Instruction Level Parallelism
6. To impart the knowledge on micro programming
7. Concepts of advanced pipelining techniques.

### **Course outcomes**

1. Draw the functional block diagram of a single bus architecture of a computer and describe the function of the instruction execution cycle, RTL interpretation of instructions, addressing modes, instruction set.
2. Write assembly language program for specified microprocessor for computing 16 bit multiplication, division and I/O device interface (ADC, Control circuit, serial port communication).
3. Write a flowchart for Concurrent access to memory and cache coherency in Parallel Processors and describe the process.
4. Given a CPU organization and instruction, design a memory Unit and analyze its operation by interfacing with the CPU.
5. Given a CPU organization, assess its performance, and apply design techniques to enhance performance using pipelining, parallelism and RISC methodology

### **Course Content**

#### **Unit-I**

Register Transfer Language, Bus and Memory Transfers, Bus Architecture, Bus Arbitration, Arithmetic Logic, Shift Microoperation, Arithmetic Logic Shift Unit, Design of Fast address, Arithmetic Algorithms (addition, subtraction, Booth Multiplication), IEEE standard for Floating point numbers.

#### **Unit-II**

Control Design: Hardwired & Micro Programmed (Control Unit): Fundamental Concepts (Register Transfers, performing of arithmetic or logical operations, fetching a word from memory, Storing a word in memory), Execution of a complete instruction, Multiple-Bus organization, Hardwired Control, Micro programmed control(Microinstruction, Microprogram sequencing, Wide-Branch addressing, Microinstruction with Next-address field, Prefetching Microinstruction).

#### **Unit-III**

Processor Design: Processor Organization: General register organization, Stack organization, Addressing mode, Instruction format, Data transfer & manipulations, Program Control, Reduced Instruction Set Computer. Input-Output Organization: I/O Interface, Modes of transfer, Interrupts & Interrupt handling, Direct Memory access, InputOutput processor, Serial Communication.

#### **Unit-IV**

Memory Organization: Memory Hierarchy, Main Memory (RAM and ROM Chips), organization of Cache Memory, Auxiliary memory, Cache memory, Virtual Memory, Memory management hardware.

#### **Unit-V**

Parallel Processing, Pipelining- Arithmetic Pipelining, Instruction Pipelining, RISC Pipelining, Vector Processing, Array Processor. Multiprocessor: Characteristic of Multiprocessor, Interconnection Structure, Interprocessor Arbitration, Cache Coherence

#### **List of Experiments**

1. Bread-board implementation of various flip-flops.
2. Bread-board implementation of counters & shift registers.
3. Determination of Delay time and NAND, NOR, Ex-OR, AND & OR Gates.
4. Bread Board Implementation of Flip-Flops.
5. Experiments with clocked Flip-Flop.
6. Design of Counters.
7. Bread Board implementation of counters & shift registers.
8. Implementation of Arithmetic algorithms.
9. Bread Board implementation of Adder/Subtractor (Half, Full)
10. Transfer characteristics of TTL inverters & TTL Schmitt Trigger inverter.
11. Transfer characteristics of CMOS inverters series and CD40 series and
12. Estimation of Gate delay of CD40 series CMOS inverter.
13. Monoshot multivibrators using 74121 and 74123.
14. Clock circuit realization using 555 and CMOS inverter and quartz crystal.
15. Adder/ subtractor operation using IC7483 4 bit/ 8 bit.
16. Demultiplexer / Decoder operation using IC-74138.

#### **Text Book:**

1. Computer Organization, John P.Hayes, McGraw Hill, 3rd Edition.
2. Computer System Architecture, M. Mano, Pearson, 3rd Edition.

#### **Reference Book:**

1. Computer Organization, Vravice, Zaky & Hamacher (TMH Publication)
2. Structured Computer Organization, Tannenbaum(PHI)
3. Computer Organization, Stallings(PHI)
4. R.S.Gaonkar - Microprocessor architecture – Programming and Application with 8085/8080A - Wiley Eastern Limited

<b>BCST 405 &amp; BCSP 405</b>	<b>Theory of Automata &amp; Formal Languages</b>	<b>3L:1T:0P</b>	<b>4 credits</b>
------------------------------------	--	-----------------	------------------

### **Course Objectives**

1. To define mathematical methods of computing devices, called abstract machines, namely Finite Automata, Pushdown Automata, and Turing Machines.
2. To study the capabilities of these abstract machines.
3. To classify machines by their power to recognize languages.
4. Employ finite state machines to solve problems in computing
5. Explain deterministic and non- deterministic machines.
6. Identify different formal language classes and their relationships
7. Design grammars and recognizers for different formal languages
8. Determine the decidability and intractability of computational problems
9. Comprehend the hierarchy of problems arising in the computer sciences.

### **Course Outcomes:**

1. Apply knowledge in designing or enhancing compilers.
2. Design grammars and automata (recognizers) for different language classes.
3. Apply knowledge in developing tools for language processing or text processing.

### **Course Content**

#### **Unit I**

Introduction to defining language, Kleene closures, Arithmetic expressions, defining grammar, Chomsky hierarchy, Finite Automata (FA), Transition graph, generalized transition graph.

#### **Unit II**

Nondeterministic finite Automata (NFA), Deterministic finite Automata (DFA), Construction of DFA from NFA and optimization, FA with output: Moore machine, Mealy machine and Equivalence, Applications and Limitation of FA.

#### **Unit III**

Arden Theorem, Pumping Lemma for regular expressions, Myhill-Nerode theorem, Context free grammar: Ambiguity, Simplification of CFGs, Normal forms for CFGs, Pumping lemma for CFLs, Decidability of CFGs, Ambiguous to Unambiguous CFG.

#### **Unit IV**

Push Down Automata (PDA): Description and definition, Working of PDA, Acceptance of a string by PDA, PDA and CFG, Introduction to auxiliary PDA and Two stack PDA.

#### **Unit V**

Turing machines (TM): Basic model, definition and representation, Language acceptance by TM, TM and Type – 0 grammar, Halting problem of TM, Modifications in TM, Universal TM, Properties of recursive and recursively enumerable languages, unsolvable decision problem, undecidability of Post correspondence problem, Church's Thesis, Recursive function theory.

### List of Experiments

1. Design a Program for creating machine that accepts three consecutive one.
2. Design a Program for creating machine that accepts the string always ending with 101.
3. Design a Program for Mode 3 Machine
4. Design a program for accepting decimal number divisible by 2.
5. Design a program for creating a machine which accepts string having equal no. of 1's and 0's.
6. Design a program for creating a machine which count number of 1's and 0's in a given string.
7. Design a Program to find 2's complement of a given binary number.
8. Design a Program which will increment the given binary number by 1.
9. Design a Program to convert N DFA to DFA.
10. Design a Program to create PDA machine that accept the well-formed parenthesis.
11. Design a PDA to accept WCWR where w is any string and WR is reverse of that string and C is a Special symbol.
12. Design a Turing machine that's accepts the following language  $L = \{a^n b^n c^n \mid \text{where } n > 0\}$ .
13. Introduction, Setup & Installation of JFLAP software/relevant software for Theory of Automata & Formal Languages.
14. Create DFA to NFA and vice versa using JFLAP software/relevant software for Theory of Automata & Formal Languages.
15. Convert Regular Expression into NFA using JFLAP software/relevant software for Theory of Automata & Formal Languages.

### Text Books:

1. Hopcroft, Ullman, "Introduction to Automata Theory, Language and Computation", Nerosa Publishing House, 3rd Edition
2. K.L.P. Mishra and N.Chandrasekaran, "Theory of Computer Science(Automata, Languages and Computation)", PHI, 3rd Edition

### Reference Books:

1. Martin J. C., "Introduction to Languages and Theory of Computations", TMH
2. Papadimitrou, C. and Lewis, C.L., "Elements of theory of Computations", PHI
3. Cohen D. I. A., "Introduction to Computer theory", John Wiley & Sons
4. Kumar Rajendra, "Theory of Automata (Languages and Computation)", PPM

<b>BCSP 406</b>	<b>Programming Practices (Introduction to MATLAB)</b>	<b>0L:0T:4P</b>	<b>2 Credits</b>
-----------------	---	-----------------	------------------

### Course Objectives:

1. The course is intended to assist undergraduates in learning the basics of programming in general and programming MATLAB in particular.
2. Basics of programming in MATLAB will be covered, with the goal of having students become comfortable enough to continue learning MATLAB and other programming languages on their own.

### Course Outcomes:

At the end of the course, students will be able to

1. Use MATLAB for programming purposes
2. Learn and explore MATLAB further on their own
3. Use this learning experience to learn other programming languages.

### UNIT 1: INTRODUCTION

*Data types and variables:* Introduction to MATLAB, Data Types, Inter-conversion of Data types, MATLAB Variables, Keywords and Constant, Session Command. *MATLAB Operators and Operations:* Operators (Arithmetic, Relational, Logical, Bitwise), Set Operations, Operator Precedence, Mathematical Functions.

### UNIT 2: PROGRAMMING IN MATLAB

*Script and Function:* Decision Making, Loops, branches, Functions, Working on Script File (Creating, Saving and Executing), MATLAB I/O, Formatted I/O Method.

### UNIT 3: ARRAYS AND GRAPHICS

*Matrices and Arrays:* Introduction to Matrices, Operations on Arrays/Matrices, Manipulations of Arrays/Matrices, Expansion of Matrix Size, Reduction of Matrices/Arrays order, *Graphics:* Introduction to plot, Basic 2-D Plots (Style options, Labels, Axis control, etc.), specialized 2-D Plots, drawing multiple plots. Using MATLAB for fractals and chaos and Conway game of life

### UNIT 4: FILE HANDLING AND DEBUGGING

*File Handling:* Introduction to file handling, working on files, accessing of Text File, Saving/ Loading MATLAB Variables, reading data without opening file, reading and writing Excel. *Debugging:* Introduction to debugging, Break points, debugger, stepping, watching variable values, debugging commands.

### REFERENCES:

1. Delores M. Etter, David C. Kuncicky, Holly Moore, “*Introduction to MATLAB 7.0*”, Pearson, 2013.
2. Rudra Pratap, “*Getting Started with MATLAB*”, OXFORD University Press, 2010.
3. Agam Kumar Tyagi, “*MATLAB and Simulink for Engineers*”, University Press, 2012.

### WEB REFERENCES

<https://ocw.mit.edu/courses/mathematics/18-s997-introduction-to-matlab-programming-fall-2011/syllabus/>



<b>BCST 408</b>	<b>Cyber Security</b>	<b>Non- Credit Course</b>
-----------------	-----------------------	---------------------------

**Course Objectives:**

1. Understand the basic concept of Cyber Security.
2. Understand the basic concept of Viruses.
3. Understand the basic concept of Digital Attacks.
4. Understand the basic concept of Phishing.
5. Understand the basic concept of Cyber Law.

**Course Outcomes:**

After the completion of this course the student will able to:

1. Know about various attacks and viruses in cyber systems
2. Know about how to prevent digital attacks
3. Know about how to prevent Phishing Attacks
4. Know about how to do secure transactions

**UNIT-1**

Introduction to information systems, Types of information Systems, Development of Information Systems, Introduction to information security, Need for Information security, Threats to Information Systems, Information Assurance, Cyber Security, and Security Risk Analysis.

**UNIT-2**

Application security (Database, E-mail and Internet), Data Security Considerations-Backups, Archival Storage and Disposal of Data, Security Technology-Firewall and VPNs, Intrusion Detection, Access Control.

Security Threats -Viruses, Worms, Trojan Horse, Bombs, Trapdoors, Spoofs, E-mail viruses, Macro viruses, Malicious Software, Network and Denial of Services Attack, Security Threats to E-Commerce- Electronic Payment System, e- Cash, Credit/Debit Cards. Digital Signature, public Key Cryptography.

**UNIT-3**

Developing Secure Information Systems, Application Development Security, Information Security Governance & Risk Management, Security Architecture & Design Security Issues in Hardware, Data Storage & Downloadable Devices, Physical Security of IT Assets, Access Control, CCTV and intrusion Detection Systems, Backup Security Measures.

**UNIT-4**

Security Policies, Why Policies should be developed, WWW policies, Email Security policies, Policy Review Process-Corporate Policies-Sample Security Policies, Publishing and Notification Requirement of the Policies.

Information Security Standards-ISO, IT Act, Copyright Act, Patent Law, IPR. Cyber Laws in India; IT Act 2000 Provisions, Intellectual Property Law: Copy Right Law, Software License, Semiconductor Law and Patent Law.

**References:**

1. Charles P. Pfleeger, Shari Lawerance Pfleeger, “Analysing Computer Security ”, Pearson Education India.
2. V.K. Pachghare, “Cryptography and information Security”, PHI Learning Private Limited, Delhi India.
3. 3.Dr. Surya Prakash Tripathi, Ritendra Goyal, Praveen kumar Shukla ,”Introduction to Information Security and Cyber Law” Willey Dreamtech Press.
4. Schou, Shoemaker, “ Information Assurance for the Enterprise”, Tata McGraw Hill. 5. CHANDER, HARISH,” Cyber Laws And It Protection ” , PHI Learning Private Limited ,Delhi ,India