VEER MADHO SINGH BHANDARI UTTARAKHAND TECHNICAL UNIVERSITY
(Formerly Uttarakhand Technical University, Dehradun Established by Uttarakhand State Govt. wide Act no. 415 of 2005)
Suddhowala, PO-Chandanwadi, Premnagar, Dehradun, Uttarakhand (Website- www.uktech.ac.in)

SYLLABUS

Approved in 13th Meeting of Executive Council held on 27th March 2023 subsequent to the 14th Meeting of Academic Council held on 20th March 2023

(For admission in 2022-23 and onwards)

VEER MADHO SINGH BHANDARI UTTARAKHAND TECHNICAL UNIVERSITY

(Formerly Uttarakhand Technical University, Dehradun Established by Uttarakhand State Govt. wide Act no. 415 of 2005) Suddhowala, PO-Chandanwadi, Premnagar, Dehradun, Uttarakhand

SYLLABUS

For

B.TECH

(Computer Science and Engineering)

2ND Year

Effective From – Session 2023-24

SEMESTER-III													
S. NO.	Subject Codes	Category	Subject	Periods			Evaluation Scheme					Subject Total	Credit
							Sessional Exam			ESE			
				L	T	P	CT	TA	Total	TE	PE		
1	AHT-006/ ECT-033	BSC/ ESC	Advanced Applied Mathematics / Digital Electronics	3	1	0	30	20	50	100		150	4
2	AHT-007/AHT-008	HSC	Technical Communication/ Universal Human Value	2	1	0	30	20	50	100		150	3
3	CST-002	DC	Discrete Structure	3	1	0	30	20	50	100		150	4
4	CST-003	DC	Data Structures and Algorithms	3	1	0	30	20	50	100		150	4
5	CST-004	DC	Object Oriented Programming	3	1	0	30	20	50	100		150	4
6	CSP-003	DLC	Data Structures and Algorithms Lab	0	0	2		25	25		25	50	1
7	CSP-004	DLC	Object Oriented Programming Lab	0	0	2		25	25		25	50	1
8	CSP-005	DLC	Python Programming Lab	0	0	2		25	25		25	50	1
9	CSP-006	DLC	Internship-I/Mini Project-I*	0	0	2			50			50	1
10	CST-005/CST-006	MC	Python Programming/ Cyber Security	2	0	0	15	10	25	50			
11	GP-003	NC	General Proficiency						50				
			Total									950	23
12			Minor Course (Optional)**	3	1	0	30	20	50	100			4
	*The Mini Project-I or Internship-I(3-4weeks) will be conducted during summer break after the II semester and will be assessed during the III semester												
	MOOCs course												

Abbreviations: **L**-No. of Lecture hours per week, **T**-No. of Tutorial hours per week, **P**-No. of Practical hours per week, **CT**- Class Test Marks, **TA**-Marks of teacher's assessment including student's class performance and attendance, **PS**-Practical Sessional Marks, **ESE**-End Semester Examination, **TE**- Theory Examination Marks, PE- Practical External Examination Marks

Minor Courses (Optional) **: Select any subject from Annexure – II from other departments

1 Hr Lecture	**1 Hr Tutorial**	**2 or 3 Hr Practical**
1 Credit	**1 Credit**	**1 Credit**

SEMESTER-IV

S. NO.	Subject Codes	Category	Subject	Periods			Evaluation Scheme					Subject Total	Credit
							Sessional Exam			ESE			
				L	T	P	CT	TA	Total	TE	PE		
1	AHT-006/ ECT-033	BSC/ ESC	Advanced Applied Mathematics / Digital Electronics	3	0	0	30	20	50	100		150	3
2	AHT-007/AHT-008	HSC	Technical Communication/ Universal Human Value	2	1	0	30	20	50	100		150 150	3 4
3	CST-007	DC	Computer Organization and Architecture	3	1	0	30	20	50	100		150	4
4	CST-008	DC	JAVA Programming	3	1	0	30	20	50	100		150	4
5	CST-009	DC	Formal Languages & Automata Theory	3	1	0	30	20	50	100		150	4
6	CSP-007	DLC	Computer Organization and Architecture Lab	0	0	2		25	25		25	50	1
7	CSP-008	DLC	JAVA Programming Lab	0	0	2		25	25		25	50	1
8	CSP-009	DLC	UNIX/LINUX Lab	0	0	2		25	25		25	50	1
9	CST-005/ CST-006	MC	Python Programming/ Cyber Security	2	0	0	15	10	25	50			
10	GP-004	NC	General Proficiency						50				
			Total									900	22
11			Minor Course (Optional)	3	1	0	30	20	50	100			4
		DLC	Internship-II/Mini Project-II*	To be completed at the end of the fourth semester (during the Summer).									
	MOOCs course												

Abbreviations: **L**-No. of Lecture hours per week, **T**-No. of Tutorial hours per week, **P**-No. of Practical hours per week, **CT**- Class Test Marks, **TA**-Marks of teacher's assessment including student's class performance and attendance, **PS**-Practical Sessional Marks, **ESE**-End Semester Examination, **TE**- Theory Examination Marks, PE- Practical External Examination Marks

Minor Courses (Optional) **: Select any subject from Annexure – II from other departments

1 Hr Lecture	**1 Hr Tutorial**	**2 or 3 Hr Practical**
1 Credit	**1 Credit**	**1 Credit**

Advanced Applied Mathematics (AHT-006)

L:T:P:: 3:1:0 **Credits-4**

COURSE OBJECTIVES: The objectives of the course are to:

1. The idea of Laplace transform of functions and their applications.
2. The idea of Fourier transform of functions and their applications.
3. Evaluate roots of algebraic and transcendental equations.
4. Interpolation, numerical differentiation& integration and the solution of differential equations.
5. Acquaintance with statistical analysis and techniques.

COURSE OUTCOMES:

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. Apply the concept of Fourier transform to evaluate engineering problems.
3. Understand to evaluate roots of algebraic and transcendental equations.
4. Solve the problem related interpolation, differentiation, integration and the solution of differential equations.
5. Understand the concept of correlation, regression, moments, skewness and kurtosis and curve fitting.

Module 1: 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.

Module 2: Fourier Transforms: **(8 hours)**

Fourier integral, Fourier sine and cosine integral, Complex form of Fourier integral, Fourier transform,Inverse Fouriertransforms, Convolution theorem, Fourier sine and cosine transform, Applications of Fourier transform to simple one dimensional heat transfer equations.

Module 3: Solution of Algebraic & Transcendental equations and 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.

Module 4: Numerical differentiation & Integration and 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 predicator-corrector method.

Module 5: Statistical Techniques: (8 hours)

Introduction: Measures of central tendency, Moments, 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,10th ed.
2. B.V. Ramana: Higher Engineering Mathematics,McGrawHill.
3. Peter V.O'Neil: Advanced Engineering Mathematics, Cengage Learning,7th ed.
4. B.S. Grewal: Higher Engineering Mathematics, Khanna Publishers, 44th ed.
5. T.Veerarajan: Engineering Mathematics (for semester III), McGrawHill, 3rd ed.
6. R.K. Jain and S.R.K. Iyenger: Advance Engineering Mathematics,Narosa PublishingHouse,Std. ed.
7. P. Kandasamy, K. Thilagavathy, K. Gunavathi: Numerical Methods, S. Chand.
8. S.S. Sastry: Introductory methods of numerical analysis, Prentice Hall India, 5thed.
9. N.P. Bali and Manish Goyal: Computer Based Numerical and Statistical Techniques,Laxmi Publications, 5th ed.
10. J.N. Kapur: Mathematical Statistics, S. Chand &Company.
11. D.N.Elhance,V. Elhance&B.M. Aggarwal: Fundamentals of Statistics,Kitab Mahal.

DIGITAL ELECTRONICS (ECT-033)

L:T:P:: 3:1:0 **Credits-04**

COURSE OBJECTIVES: The objectives of the course are to:

1. Understand the basics of digital electronics.
2. Understand the basics of Logic family.
3. Apply the knowledge of digital electronics to construct various digital circuits.
4. Analyze the characteristics and explain the outputs of digital circuits.
5. Evaluate and asses the application of the digital circuits.
6. Understand the design flow of VLSI Circuits

COURSEOUTCOMES: After completion of the course student will be able to:

1. Understand the Boolean algebra and minimization of digital functions.
2. Design and implement various combinational circuits.
3. Design and implement various sequential circuits.
4. Understand the digital logic families, semiconductor memories.
5. Design the digital circuits using VHDL

UNIT 1: **MINIMIZATION OF LOGIC FUNCTIONS**: Review of logic gate and Boolean algebra, DeMorgan's Theorem, SOP & POS forms, canonical forms, don't care conditions, K-maps up to 6 variables, Quine-McClusky's algorithm, X-OR & X-NOR simplification of K-maps, binary codes, code conversion.

UNIT 2: **COMBINATIONAL CIRCUITS**: Combinational circuit design, half and full adders, subtractors, serial and parallel adders, code converters, comparators, decoders, encoders, multiplexers, de-multiplexer, parity checker, driver &multiplexed display, BCD adder, Barrel shifter and ALU.

UNIT 3: **SEQUENTIAL CIRCUITS**: Building blocks like S-R, JK and master-slave JK FF, edge triggered FF, ripple and synchronous counters, shift registers, finite state machines, design of synchronous FSM, algorithmic state machines charts, designing synchronous circuits like pulse train generator, pseudo random binary sequence generator, clock generation

UNIT 4: **LOGIC FAMILIES & SEMICONDUCTOR MEMORIES**: TTL NAND gate, specifications, noise margin, propagation delay, fan-in, fan-out, tri-state TTL, ECL, CMOS families and their interfacing, memory elements, concept of programmable logic devices like FPGA, logic implementation using programmable devices.

UNIT 5: **VLSI DESIGN FLOW**: Design entry: schematic, FSM & HDL, different modelling styles in VHDL, data types and objects, dataflow, behavioral and structural modelling, synthesis and simulation VHDL constructs and codes for combinational and sequential circuit

BOOKS:

1. Mano, Digital electronics,TMH,2007.
2. Malvino, Digital Principle and applications, TMH,2014.
3. Jain, Modern digital electronics, PHI, 2012.
4. Tocci, Digital Electronics, PHI,2001.
5. W.H.Gothmann, “Digital Electronics-An introduction to theory and practice”,PHI,2nd edition,20

Technical Communication (AHT-007)

L:T:P:: 2:1:0 **Credits-03**

COURSE OBJECTIVES: The objectives of the course are:

1. Produce technical documents that use tools commonly employed by engineering and computer science professionals.
2. Communicate effectively in a professional context, using appropriate rhetorical approaches for technical documents, adhering to required templates, and complying with constraints on document format.
3. Clarify the nuances of phonetics, intonation and pronunciation skills.
4. Get familiarized with English vocabulary and language proficiency.

COURSE OUTCOMES: On successful completion of the course, the student will be able to:

1. Students will be enabled to **understand** the nature and objective of Technical Communication relevant for the work place as Engineers.
2. Students will **utilize** the technical writing for the purposes of Technical Communication and its exposure in various dimensions.
3. Students would imbibe inputs by presentation skills to **enhance** confidence in face of diverse audience.
4. Technical communication skills will **create** a vast know-how of the application of the learning to promote their technical competence.
5. It would enable them to **evaluate** their efficacy as fluent & efficient communicators by learning the voice-dynamics.

Unit -1 Fundamentals of Technical Communication:
Technical Communication: Introduction, Features; Distinction between General and Technical Communication; The flow of Communication: Downward; upward, Lateral or Horizontal; Barriers to Communication, Importance of communication

Unit - II Forms of Technical Communication:
Technical Report: Definition & importance; Thesis/Project writing: structure & importance; synopsis writing: Methods; Technical research Paper writing: Methods & style; Seminar & Conference paper writing; 7 Cs of effective business writing: concreteness, completeness, clarity, conciseness, courtesy, correctness, consideration.

Unit - III Technical Presentation: Strategies & Techniques

Presentation: Forms; interpersonal Communication; Class Room presentation; style;method, Public Speaking: method; Techniques: Clarity of substance; emotion; Humour; Modes of Presentation; Overcoming Stage Fear: Confident speaking; Audience Analysis & retention of audience interest; Methods of Presentation: Interpersonal; Impersonal; Audience Participation: Quizzes & Interjections

Unit - IV Technical Communication Skills

Interview skills; Group Discussion: Objective & Method; Seminar/Conferences Presentation skills: Focus; Content; Style; Argumentation skills: Devices: Analysis; Cohesion & Emphasis; Critical thinking; Nuances, exposition, narration and description

Unit - V Kinesics & Voice Dynamics:

Kinesics: Definitions; importance; Features of Body Language; Voice Modulation: Quality, Pitch; Rhythm; intonation, pronunciation, articulation, vowel and consonants sounds

Reference Books

1. Technical Communication – Principles and Practices by Meenakshi Raman & Sangeeta Sharma, Oxford Univ. Press, 2007, New Delhi.
2. Business Correspondence and Report Writing by Prof. R.C. Sharma & Krishna Mohan, Tata McGraw Hill & Co. Ltd., 2001, New Delhi.
3. Practical Communication: Process and Practice by L.U.B. Pandey; A.I.T.B.S. Publications India Ltd.; Krishan Nagar, 2014, Delhi.
4. Modern Technical Writing by Sherman, Theodore A (et.al); Apprentice Hall; New Jersey; U.S.
5. A Text Book of Scientific and Technical Writing by S.D. Sharma; Vikas Publication, Delhi.
6. Skills for Effective Business Communication by Michael Murphy, Harward University, U.S. Business Communication for Managers by Payal Mehra, Pearson Publication, Delhi.

UNIVERSAL HUMAN VALUES (AHT-008)

L:T:P:: 2:1:0 **Credits-03**

COURSE OBJECTIVES: The objectives of the course are to:

1. Development of a holistic perspective based on self- exploration about themselves (human being), family, society and nature/existence.
2. Understanding (or developing clarity) of the harmony in the human being, family, society and nature/existence.
3. Strengthening of self-reflection.
4. Development of commitment and courage to act.

COURSE OUTCOMES: On successful completion of the course, the student will be able to:

1. Expected to become more aware of themselves, and their surroundings (family, society, nature)
2. Become more responsible in life, and in handling problems with sustainable solutions, while keeping human relationships and human nature in mind.
3. Have better critical ability.
4. Become sensitive to their commitment towards what they have understood (human values, human relationship and human society).
5. Able to apply what they have learnt to their own self in different day-to- day settings in real life, at least a beginning would be made in this direction.

COURSE TOPICS: The course has 28 lectures and 14 practice sessions in 5 modules:

Module 1: Introduction - Value Education

Universal human values; self exploration, natural acceptance an experimental validation; Human aspirations, right understanding, relationship and physical facility, current scenario; Understanding and living in harmony at various levels.

Module 2: Harmony in the Human Being

Understanding human being, needs of self(I) and body; body as an instrument of 'I'; characteristics and activities of 'I' and harmony in 'I'; harmony of I with the Body: Sanyam and Health, Physical needs an prosperity; Programs to ensure Sanyam and Health.

Module 3: Harmony in the Family and Society

Values in human-human relationship; nine universal values in relationships; justice, truth, respect, trust; Difference between intention and competence; Respect and differentiation, Harmony in society: resolution, prosperity, fearlessness and coexistence; Universal harmonious order in society.

Module 4: Harmony in the Nature and Existence

Harmony in the nature. Four orders of nature; existence as co-existence, harmony at all levels of existence.

Module 5: Harmony in the Professional Ethics

Natural acceptance of human values**,** Definitiveness of Ethical Human Conduct**;** Basis for Humanistic Education, Humanistic Constitution and Humanistic Universal Order; Competence in professional ethics; Case studies; transition from the present state to Universal Human Order: at individual level and societal level.

TEXT BOOK

1. Human Values and Professional Ethics by R R Gaur, R Sangal, G P Bagaria, Excel Books, New Delhi, 2010

REFERENCE BOOKS

1. Jeevan Vidya: Ek Parichaya, A Nagaraj, Jeevan Vidya Prakashan, Amarkantak, 1999.
2. Human Values, A.N. Tripathi, New Age Intl. Publishers, New Delhi, 2004.
3. The Story of Stuff (Book).
4. The Story of My Experiments with Truth - by Mohandas Karam chand Gandhi.
5. Small is Beautiful - E. F Schumacher.
6. Slow is Beautiful - Cecile Andrews
7. Economy of Permanence - J C Kumarappa
8. Bharat Mein Angreji Raj - PanditSunderlal
9. Rediscovering India - by Dharampal
10. Hind Swaraj or Indian Home Rule - by Mohandas K. Gandhi
11. India Wins Freedom - Maulana Abdul Kalam Azad
12. Vivekananda - Romain Rolland (English)
13. Gandhi - Romain Rolland (English)

DISCRETE STRUCTURE (CST-002)

L:T:P:: 3:1:0 **Credits-04**

COURSE OBJECTIVES: The objectives of the course are to:

1. To introduce several Discrete Mathematical Structures to serve as tools in the development of theoretical computer science.
2. Transform a given problem into a combination of several simpler statements, reach at a solution and prove it logically.
3. Enhance the ability to reasoning and presenting the mathematically accurate argument.
4. Apply the abstract concepts of graph theory in the modelling and solving of non-trivial.

COURSE OUTCOMES: On successful completion of the course, the student will be able to:

1. Develop new models to represent and interpret the data.
2. Apply knowledge of mathematics, probability & statistics, graph theory and logics.
3. Interpret statements presented in disjunctive normal form and determine their validity by applying the rules and methods of propositional calculus.
4. Reformulate statements from common language to formal logic using the rules of propositional and predicate calculus.
5. Apply graph theory in solving computing problems.

Unit 1- Set Theory: Introduction to set theory, set operations, Algebra of Sets, Combination of sets, Duality, Finite and infinite sets, Classes of sets, Power sets, Multi sets, Cartesian Product, Representation of relations, Types of relation, Binary relation, Equivalence relations and partitions, Mathematics Induction.

Function and its types: Composition of function and relations, Cardinality and inverse relations, Functions, logic and proofs injective, surjective and bijective functions.

Unit 2- Propositional Calculus: Basic operations; AND(^), OR(v), NOT(~), True value of a compound statement, propositions, tautologies, and contradictions. Partial ordering relations and lattices.
Lattice theory: Partial ordering, posets, lattices as posets, properties of lattices as algebraic systems, sublattices, and some special lattices.

Unit 3-Combinations: The Basic of Counting**,** Pigeonhole Principles, Permutations and Combinations, Principle of Inclusion and Exclusion.

Recursion and Recurrence Relation: linear recurrence relation with constant coefficients, Homogeneous solutions, Particular solutions, and Total solution of a recurrence relation using generating functions.

Unit 4- Algebraic Structures: Definition, elementary properties of Algebraic structures, examples of a Monoid, sunmonoid, semigroup, groups and rings, Homomorphism, Isomorphism and automorphism, Subgroups and Normal subgroups, Cyclic groups, Integral domain and fields, Rings, Division Ring.

Unit 5- Graphs and Trees: Introduction to graphs, Directed and undirected graphs, Homomorphic and Isomorphic graphs, Subgraphs, cut points and bridges, Multigraph and Weighted graphs, Paths and circuits, Shortest path in a weighted graph, Eulerian path and circuits, Hamilton paths and circuits, Planar graphs, Euler's formula, Trees, Rooted trees, Spanning trees and cut-sets, Binary trees and its traversals.

TEXTBOOKS:

1. Discrete and combinatorial mathematics-An applied introduction-5th edition- Ralph P. Grimaldi, Pearson Education.
2. Discrete Mathematics for Computer Scientists & Mathematicians, J.L. Mott. A. Kandel, T.P. Baker, Prentice Hall.

REFERENCE BOOKS:

1. Discrete mathematical with graph theory, edgar G. Goodaire, 3rd Edition, Pearson Education.
2. Discrete Mathematics and its Applications, Kenneth H. Rosen, Fifth Edition. TMH.
3. Mathematical foundations of computer science-Dr S. Chandra sekharaiah-Prism books Prv. Lt.
4. Discrete mathematical structures Theory and applications-malik & Sen.
5. Logic and Discrete Mathematics, Grass Mann & Trembley, Person Education.
6. Discrete mathematical structures with applications to Comp. Science- J. P. Tremblay and R. Manohar, Tata-McGraw-Hill publications.
7. Elements of DISCRETE MATHEMATICS – A computer-oriented Approach – C L Liu, D P Mohapatra. Third Edition, Tata McGraw Hill

DATA STRUCTURES AND ALGORITHMS (CST-003)

L:T:P:: 3:1:0 **Credits-04**

Course Objectives: The objectives of this course are to:

1. Introduce the fundamentals of Data Structures, Abstract concepts and how these concepts are useful in problem-solving.
2. Analyze step by step and develop algorithms to solve real-world problems.
3. Implement various data structures, viz. Stacks, Queues, Linked Lists, Trees and Graphs.
4. Understand various searching & sorting techniques

Course Outcomes: On successful completion of the course, the student will be able to:

1. Compare functions using asymptotic analysis and describe the relative merits of worst-case, average-case, and best-case analysis.
2. Become familiar with a variety of sorting algorithms and their performance characteristics (e.g., running time, stability, space usage) and be able to choose the best one under a variety of requirements.
3. Understand and identify the performance characteristics of fundamental algorithms and data structures and be able to trace their operations for problems such as sorting, searching, selection, operations on numbers, and graphs.
4. Solve real-world problems using arrays, stacks, queues, and linked lists.
5. Become familiar with the major graph algorithms and their analyses. Employ graphs to model engineering problems when appropriate.

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 Queues: Simple Queue, Circular Queue, Priority Queue; Operations on each type 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 the 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 complexity analysis.

Unit 4-Trees and Graphs: 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.

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

Unit 5-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: Symbol table, Hashing Functions, Collision-Resolution Techniques

TEXTBOOKS:

1. An Introduction to Data Structures with Applications. by Jean-Paul Tremblay & Paul G. Sorenson Publisher-Tata McGraw Hill.
2. Ritika Mehra, Data Structures Using C, Pearson Education.
3. Data Structures using C & C++ -By Ten Baum Publisher – Prentice-Hall International**.**

REFERENCE BOOKS:

1. Schaum's Outlines Data structure Seymour Lipschutz Tata McGraw Hill 2nd Edition.
2. Fundamentals of Computer Algorithms by Horowitz, Sahni, Galgotia Pub. 2001 ed.
3. Fundamentals of Data Structures in C++-By Sartaj Sahani.
4. Data Structures: A Pseudo-code approach with C -By Gilberg&Forouzan Publisher-Thomson Learning.

OBJECT ORIENTED PROGRAMMING (CST-004/CSO-053)

L:T:P:: 3:1:0 **Credits-04**

COURSE OBJECTIVES: The objectives of this course are to:

1. Provide flexible and powerful abstraction.
2. Allow programmers to think the problem in terms of the structure rather than in terms of structure of the computer.
3. Decompose the problem into a set of objects.
4. Objects interact with each other to solve the problem.
5. Create new type of objects to model elements from the problem space

COURSE OUTCOMES: On successful completion of the course, the student will be able to:

1. Recognize features of object-oriented design such as encapsulation, polymorphism, inheritance, and composition of systems based on object identity.
2. Apply some common object-oriented design patterns.
3. Specify simple abstract data types and design implementations using abstraction functions to document them.
4. Design a convenient way for the handling problems using templates and use simple try-catch blocks for Exception Handling.
5. Manage I/O streams and File I/O oriented interactions.

Unit 1- Object Oriented Programming Concepts: Classes and Objects, Methods and Messages, Abstraction and Encapsulation, Inheritance, Abstract Classes, Polymorphism. Introduction to C++: Classes and Objects, Structures and Classes, Unions and Classes, Friend Functions, Friend Classes, Inline Functions, Static Class Members, Scope Resolution Operator, Nested Classes, Local Classes, Passing Objects to Functions, Returning objects, object assignment. Arrays, Pointers, References, and the Dynamic Allocation Operators: Arrays of Objects, Pointers to Objects, Type Checking, this Pointer, Pointers to Derived Types, Pointers to Class Members, References, Dynamic Allocation Operators.

Unit 2- Function Overloading and Constructors: Function Overloading, Constructors, parameterized constructors, Copy Constructors, Overloading Constructors, Finding the Address of an Overloaded Function, Default Function Arguments, Function Overloading and Ambiguity. Operator overloading: Creating member Operator Function, Operator Overloading Using Friend Function, Overloading New and Delete, Overloading Special Operators, Overloading Comma Operator.

Unit 3- Inheritance and Polymorphism: Inheritance: Base-Class Access Control, Inheritance and Protected Members, Inheriting Muitiple Base Classes, Constructors, Destructors and Inheritance, Granting Access, Virtual Base Classes. Polymorphism: Virtual Functions, Virtual Attribute and Inheritance, Virtual Functions and Hierarchy, Pure Virtual

Functions, Early vs. Late Binding, Run-Time Type ID and Casting Operators: RTTI, Casting Operators, Dynamic Cast.

Unit 4- Templates and Exception Handling: Templates: Generic Functions, Applying Generic Functions, Generic Classes, The type name and export Keywords, Power of Templates, Exception Handling: Fundamentals, Handling Derived Class Exceptions, Exception Handling Options, Understanding terminate() and unexpected(), uncaught_exception () Function, exception and bad_exception Classes, Applying Exception Handling.

Unit 5- I/O System Basics: Streams and Formatted 1/O. File I/O: File Classes, File Operations. Namespaces: Namespaces, std Namespace. Standard Template Library: Overview, Container Classes, General Theory of Operation, Lists, string Class, Final Thoughts on STL.

TEXTBOOKS:

1. Object Oriented Programming with C++ by E. Balagurusamy, McGraw-Hill Education (India).
2. ANSI and Turbo C++ by Ashoke N. Kamthane, Pearson Education

REFERENCE BOOKS:

1. Big C++ - Wiley India.
2. C++: The Complete Reference- Schildt, McGraw-Hill Education (India).
3. C++ and Object Oriented Programming – Jana, PHI Learning.
4. Object Oriented Programming with C++ - Rajiv Sahay, Oxford.
5. Mastering C++ - Venugopal, McGraw-Hill Education (India)

DATA STRUCTURES AND ALGORITHMS LAB (CSP-003)

L:T:P:: 0:0:2 **Credits-01**

Course Objectives: The objectives of this course are to:

1. Analyse step by step development of algorithms to solve real-world problems.
2. Implement various data structures, viz. Stacks, Queues, Linked Lists, Trees and Graphs.
3. Understand various data searching & sorting techniques.

Course Outcomes: On successful completion of the course, the student will be able to:

1. Develop programs using dynamic memory allocation and linked list ADT.
2. Apply Stack and Queue to solve problems.
3. Implement the concept of hashing in real-time dictionaries.
4. Identify and implement suitable data structures for the given problem.
5. Solve real-world problems by finding the minimum spanning tree and the shortest path algorithm.

LIST OF EXPERIMENTS:

1. Write programs to implement the following using an array.
   a) Stack ADT
   b) Queue ADT
2. Write programs to implement the following using a singly linked list.
   a) Stack ADT
   b) Queue ADT
3. Write a program to implement the deque (double-ended queue) ADT using a doubly linked list.
4. Write a program to perform the following operations:
   a) Insert an element into a binary search tree.
   b) Delete an element from a binary search tree.
   c) Search for a key element in a binary search tree.
5. Write a program to implement circular queue ADT using an array.
6. Write a program to implement all the functions of a dictionary (ADT) using hashing.
7. Write a program to perform the following operations on B-Trees and AVL-trees:
   a) Insertion.
   b) Deletion.
8. Write programs for implementing BFS and DFS for a given graph.
9. Write programs to implement the following to generate a minimum cost-spanning tree:

a) Prim's algorithm.

b) Kruskal's algorithm.

10. Write a program to solve the single source shortest path problem.

(Note: Use Dijkstra's algorithm).

11. Write a program that uses non-recursive functions to traverse a binary tree in:

a) Pre-order.

b) In-order.

c) Post-order.

12. Write programs for sorting a given list of elements in ascending order using the following sorting methods:

a) Quick sort.

b) Merge sort.

OBJECT ORIENTED PROGRAMMING LAB (CSP-004)

L:T:P:: 0:0:2 **Credits-01**

COURSE OBJECTIVES: The objectives of this course are to:

1. Build software development skills using C++programming for real-world applications.
2. Understand and apply the concepts of classes, packages, interfaces, List, exception handling and file processing.
3. Develop applications using event handling.

COURSE OUTCOMES: On successful completion of the course, the student will be able to:

1. Design object-oriented programs with static members and friend functions using C++.
2. Implement C++ programs with operator overloading and type conversions.
3. Develop class templates for various data structures like stack, queue and linked list.
4. Create classes with necessary exception handling
5. Construct simple test applications using polymorphism.

LIST OF EXPERIMENTS

1. Design C++ classes with static members, methods with default arguments, and friend functions. (For example, design matrix and vector classes with static allocation, and a friend function to do matrix-vector multiplication).
2. Implement Matrix class with dynamic memory allocation and necessary methods. Give proper constructor, destructor, copy constructor, and overloading of the assignment operator.
3. Implement complex number class with necessary operator overloading and type conversions such as integer to complex, double to complex, complex to double etc.
4. Overload the new and delete operators to provide a custom dynamic allocation of memory.
5. Develop C++ class hierarchy for various types of inheritances.
6. Design a simple test application to demonstrate dynamic polymorphism and RTTI.
7. Develop a template of the linked-list class and its methods.
8. Develop templates of standard sorting algorithms such as bubble sort, insertion sort and quick sort.
9. Design stack and queue classes with necessary exception handling.

10. Write a C++ program that randomly generates complex numbers (use previously designed Complex class) and write them two per line in a file along with an operator (+, -, *, or /). The numbers are written to file in the format (a + ib). Write another program to read one line at a time from this file, perform the

PYTHON PROGRAMMING LAB (CSP-005)

L:T:P:: 0:0:2 **Credits-01**

COURSE OBJECTIVES: The objectives of this course are to:

1. Learn and understand Python programming basics and control statements.
2. Illustrate the applications of string handling and regular expressions in building Python programs using functions.
3. Discover the use of supported data structures like lists, dictionaries, and tuples in Python.
4. Understand a range of Object-Oriented Programming and in-depth data and information processing techniques.
5. Apply the concepts of file I/O in python.

COURSE OUTCOMES: On successful completion of the course, the student will be able to:

1. Demonstrate the basic concepts of python programming with the help of data types, operators and expressions, and console input/output.
2. Apply the concept of Control Structures in Python to solve any given problem.
3. Demonstrate operations on built-in container data types (list, tuple, set, dictionary) and strings.
4. Ability to explore python, especially the object-oriented concepts and the built-in objects of Python.
5. Implement the concepts of file handling using packages.

LIST OF PROGRAMS:

Exercise 1 - Basics

a) Running instructions in Interactive interpreter and a Python Script

b) Write a program to purposefully raise Indentation Error and Correct it

Exercise 2 - Operations

a) Write a program to compute distance between two points taking input from the user (Pythagorean Theorem)

b) Write a program add.py that takes 2 numbers as command line arguments and prints its sum.

Exercise - 3 Control Flow

a) Write a Program for checking whether the given number is a even number or not.

b) Using a for loop, write a program that prints out the decimal equivalents of 1/2, 1/3, 1/4, . . . ,1/10

c) Write a program using a for loop that loops over a sequence.

d) Write a program using a while loop that asks the user for a number, and prints a countdown from that number to zero.

Exercise 4 - Control Flow - Continued

A. Find the sum of all the primes below two million. Adding the previous two terms, each new term in the Fibonacci sequence is generated. By starting with 1 and 2, the first 10 terms will be:

B. 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, …
C. By considering the terms in the Fibonacci sequence whose values do not exceed four million,find the sum of the even- valued terms.
D. Linear search and Binary search
E. Selection sort, Insertion sort

Exercise - 5 - DS

a) Write a program to count the numbers of characters in the string and store them in a dictionary data structure

b) Write a program to use split and join methods in the string and trace a birthday with a dictionary data structure.

Exercise - 6 DS - Continued

a) Write a program combine_lists that combines these lists into a dictionary.

b) Write a program to count frequency of characters in a given file. Can you use character frequency to tell whether the given file is a Python program file, C program file or a text file?

Exercise - 7 Files

a) Write a program to print each line of a file in reverse order.

b) Write a program to compute the number of characters, words and lines in a file.

Exercise - 8 Functions

a) Write a function ball_collide that takes two balls as parameters and computes if they are colliding. Your function should return a Boolean representing whether or not the balls are colliding.

Hint: Represent a ball on a plane as a tuple of (x, y, r), r being the radiusIf (distance between two balls centers) <= (sum of their radii), then (they are colliding)

b) Find the mean, median, and mode for the given set of numbers in a list.

Exercise - 9 Functions - Continued

a) Write a function nearly_equal to test whether two strings are nearly equal. Two strings a and bare nearly equal when a single mutation on b can generate a.

b) Write a function dups to find all duplicates in the list.

c) Write a function unique to find all the unique elements of a list.

Exercise - 10 - Functions –Problem-Solving

a) Write a function cumulative_product to compute the cumulative product of a list of numbers.

b) Write a function reverse to reverse a list. Without using the reverse function.

c) Write a function to compute gcd, lcm of two numbers. Each function shouldn't exceed one line.

Exercise - 11–Python Packages

a) Install packages requests, flask and explore them. using (pip)

b) Plot graphs using python and Matplotlib.

c) Data Analysis using numpy and Pandas Libraries

INTERNSHIP-I/MINI PROJECT-I (CSP-006)

L:T:P:: 0:0:2 **Credits-01**

ABOUT INTERNSHIP/ MINI PROJECT

It is an organized method or activity of enhancing and improving engineering students' skill sets and knowledge, which boosts their performance and consequently helps them meet their career objectives. Industrial Training is essential in developing the practical and professional skills required for an Engineer and an aid to prospective employment.

OBJECTIVES OF INTERNSHIP/ MINI PROJECT: The objectives of this course is to:

1. Expose the students to the actual working environment and enhance their knowledge and skill from what they have learned in college.
2. Enhance the good qualities of integrity, responsibility, and self-confidence. Students must follow all ethical values and good working practices.
3. Help the students with the safety practices and regulations inside the industry and to instils the spirit of teamwork and good relationship between students and employees.

COURSE OUTCOMES: On successful completion of the course, the student will be able to:

1. Understand organizational issues and their impact on the organization and employees.
2. Identify industrial problems and suggest possible solutions.
3. Relate, apply and adapt relevant knowledge, concepts and theories within an industrial organization, practice and ethics.
4. Apply technical knowledge in an industry to solve real world problems.
5. Demonstrate effective group communication, presentation, self-management, and report writing skills.

PYTHON PROGRAMMING (CST-005)

L:T:P:: 2:0:0 **Credits-0**

Course Objectives: The objectives of this course are to:

1. Introduce the basic principles and concepts of python programming, and how python programming concepts are useful in problem-solving.
2. Write clear and effective python code.
3. To perform file operations to read and write data in files.
4. To create applications using Python Programming.

Course Outcomes: On successful completion of the course, the student will be able to:

1. Develop essential programming skills in computer programming concepts like data types.
2. Examine Python syntax and semantics and be fluent in the use of Python flow control and functions.
3. Illustrate the process of structuring the data using lists, tuples, and dictionaries.
4. Demonstrate using built-in functions and operations to navigate the file system.
5. Interpret the concepts of modules and user-defined functions in Python.

UNIT – I: Introduction and Syntax of Python Program: Features of Python, Interactive, Object-oriented, Interpreted, platform-independent, Python building blocks -Identifiers, Keywords, Indention, Variables, Comments, Python environment setup – Installation and working of IDE, Running Simple Python scripts to display a welcome message, Python variables.

Python Data Types: Numbers, String, Tuples, Lists, Dictionary. Declaration and use of datatypes, Built-in Functions.

UNIT – II: Python Operators and Control Flow statements: Basic Operators: Arithmetic, Comparison/ Relational, Assignment, Logical, Bitwise, Membership, Identity operators, Python Operator Precedence.

Control Flow: Conditional Statements (if, if...else, nested if), Looping in python (while loop, for loop, nested loops), loop manipulation using continue, pass, break, else.

UNIT – III: **Data Structures in Python: String:** Concept, escape characters, String special operations, String formatting operator, Single quotes, Double quotes, Triple quotes, Raw String, Unicode strings, Built-in String methods.

Lists: Defining lists, accessing values in lists, deleting values in lists, updating lists, Basic List Operations, and Built-in List functions.

Tuples: Accessing values in Tuples, deleting values in Tuples, and updating Tuples, Basic Tuple operations, and Built-in Tuple functions.

Sets: Accessing values in Set, deleting values in Set, and updating Sets, Basic Set operations, Built-in Set functions.

Dictionaries: Accessing values in Dictionary, deleting values in Dictionary, and updating Dictionary, Basic Dictionary operations, Built-in Dictionaries functions.

UNIT – IV: Python Functions, modules, and Packages: Use of Python built-in functions (e.g., type/data conversion functions, math functions etc.),

user-defined functions: Function definition, Function call, function arguments and parameter passing, Return statement, **Scope of Variables:** Global variable and Local Variable.

Modules: Writing modules, importing modules, importing objects from modules, Python built-in modules (e.g., Numeric, mathematical module, Functional Programming Module), Packages.

UNIT – V: File Handling: Opening files in different modes, accessing file contents using standard library functions, Reading, and writing files, closing a file, Renaming, and deleting files, File related standard functions.

TEXTBOOKS:

1. Charles R. Severance, “Python for Everybody: Exploring Data Using Python 3”, 1st Edition, CreateSpace Independent Publishing Platform, 2016.
2. Allen B. Downey, "Think Python: How to Think Like a Computer Scientist”, 2nd Edition, Green Tea Press, 2015.
3. Ch Satyanarayana, “Python Programming”, 1st Edition, universities press (India) private limited 2018.

REFERENCE BOOKS:

1. Charles Dierbach, "Introduction to Computer Science Using Python", 1st Edition, Wiley India Pvt Ltd. ISBN-13: 978-8126556014
2. Mark Lutz, “Programming Python”, 4th Edition, O’Reilly Media, 2011.ISBN-13: 978-9350232873
3. Wesley J Chun, “Core Python Applications Programming”, 3rd edition, Pearson Education India, 2015. ISBN-13: 978-9332555365
4. Roberto Tamassia, Michael H Goldwasser, Michael T Goodrich, “Data Structures and Algorithms in Python”,1stEdition, Wiley India Pvt Ltd, 2016. ISBN-13: 978- 8126562176
5. Reema Thareja, “Python Programming using problem-solving approach”, Oxford university press, 2017.

CYBER SECURITY (CST-006)

L:T:P:: 2:0:0 **Credits-0**

Course Objectives: The objectives of this course are to:

1. Familiarize with network security, network security threats, security services, and countermeasures.
2. Be aware of computer security and Internet security.
3. Study the defensive techniques against these attacks.
4. To familiarize with cyber forensics, cybercrimes, and Cyberspace laws.
5. Understand ethical laws of computers for different countries, Offences under cyberspace and the Internet in India.

Course Outcomes: On successful completion of the course, the student will be able to:

1. Understand cyber-attacks and types of cybercrimes, and familiarity with cyber forensics
2. Realize the importance of cyber security and various forms of cyber-attacks and countermeasures.
3. Get familiar with obscenity and pornography in cyberspace and understand the violation of the Right to privacy on the Internet.
4. Appraise cyber laws and how to protect themselves and, ultimately, the entire Internet community from such attacks.
5. Elucidate the various chapters of the IT Act 2008 power of the Central and State Governments to make rules under IT Act 2008.

UNIT – I: Introduction to Cyber Security: Basic Cyber Security Concepts, layers of security, Vulnerability, threat, Harmful acts, the motive of attackers, active attacks, passive attacks, Software attacks, hardware attacks, Spectrum of attacks, Taxonomy of various attacks, IP spoofing, Methods of defense, Security Models, risk management, Cyber Threats-Cyber Warfare, Cyber Crime, Cyber terrorism, Cyber Espionage, etc., CIA Triad

UNIT – II: Cyber Forensics: Introduction to cyber forensic, Historical background of Cyber forensics, Digital Forensics Science, The Need for Computer Forensics, Cyber Forensics and Digital evidence, Forensics Analysis of Email, Digital Forensics Lifecycle, Forensics Investigation, Challenges in Computer Forensics, Special Techniques for Forensics Auditing.

UNIT – III: Cybercrime (Mobile and Wireless Devices): Introduction, Proliferation of Mobile and Wireless Devices, Trends in Mobility, Credit card Frauds in Mobile and Wireless Computing Era, Security Challenges Posed by Mobile Devices, Registry Settings for Mobile Devices, Authentication service Security, Attacks on Mobile/Cell Phones, Mobile Devices: Security Implications for Organizations, Organizational

Measures for

Handling Mobile, Organizational Security Policies and Measures in Mobile Computing Era, Laptops and desktop.

UNIT – IV: Cyber Security (Organizational Implications): Introduction cost of cybercrimes and IPR issues, web threats for organizations, security and privacy implications, social media marketing: security risks and perils for organizations, social computing, and the associated challenges for organizations.

Cybercrime and Cyber terrorism: Introduction, intellectual property in cyberspace, the ethical dimension of cybercrimes, the psychology, mindset and skills of hackers and other cybercriminals.

UNIT – V: Cyberspace and the Law &Miscellaneous provisions of IT Act.: Introduction to Cyber Security Regulations, International Law. The INDIAN Cyberspace, National Cyber Security Policy. Internet Governance – Challenges and Constraints, Computer Criminals, Assets and Threats. Other offences under the Information Technology Act in India, The role of Electronic Evidence and miscellaneous provisions of the IT Act.2008.

TEXTBOOKS:

1. Nina Godbole and SunitBelpure, Cyber Security Understanding Cyber Crimes, Computer Forensics and Legal Perspectives, Wiley.
2. B. B. Gupta, D. P. Agrawal, Haoxiang Wang, Computer and Cyber Security: Principles, Algorithm, Applications, and Perspectives, CRC Press, ISBN 9780815371335, 2018.

REFERENCE BOOKS:

1. Cyber Security Essentials, James Graham, Richard Howard and Ryan Otson, CRC Press.
2. Introduction to Cyber Security, Chwan-Hwa(john) Wu,J. David Irwin, CRC Press T&F Group.
3. Debby Russell and Sr. G.T Gangemi, "Computer Security Basics (Paperback)", 2ndEdition, O' Reilly Media, 2006.
4. Wenbo Mao, "Modern Cryptography – Theory and Practice", Pearson Education, New Delhi, 2006.
5. Cyberspace and Cybersecurity, George Kostopoulos, Auerbach Publications, 2012.
6. Cyber Forensics: A Field Manual for Collecting, Examining, and Preserving Evidence of Computer Crimes, Second Edition, Albert Marcella, Jr., Doug Menendez, Auerbach Publications, 2007.
7. Cyber Laws and IT Protection, Harish Chander, PHI, 2013.

COMPUTER ORGANIZATION AND ARCHITECTURE (CST-007)

L:T:P:: 3:1:0 **Credits-04**

COURSE OBJECTIVES: The objectives of this course are to:

1. Thoroughly understand the basic structure and operation of a digital computer.
2. Study the different communication methods with I/O devices and standard I/O interfaces.
3. Learn the various instruction modes, Addressing modes and RISC and CISC Architecture
4. Study the various memory architecture.

COURSE OUTCOMES: On successful completion of the course, the student will be able to:

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.
2. Given a CPU organization and instruction, design a memory module and analyze its operation by interfacing with the CPU.
3. Design the connection between I/O address from the CPU and the I/O interface.
4. Understand the concept of Pipelining and multiprocessor.
5. Draw a flowchart for concurrent access to memory and cache coherency in parallel processors.

Unit 1- Functional Blocks of a Computer: CPU, Memory, Input-Output Subsystems, Control Unit. Instruction Set Architecture of a CPU – Registers, Instruction Execution Cycle, RTL Representation and Interpretation of Instructions, Addressing Modes, Instruction Set. Case Study – Instruction Sets of Some Common CPUs, RISC and CISC Architecture.

Unit 2- Basic Processing Unit: Signed Number Representation, Fixed Point Arithmetic, Addition and Subtraction of Signed Numbers, Multiplication of Positive Numbers, Signed Operand Multiplication Algorithm, Booth Multiplication Algorithm, division algorithm, floating point numbers and its arithmetic operation. Fundamental Concepts: Execution of a Complete Instruction, Multiple Bus Organization, Hardwired Control, Micro Programmed Control.

Unit 3- Peripheral Devices and their Characteristics: Input-Output Subsystems, I/O Device Interface, I/O Transfers– Program Controlled, Interrupt Driven and DMA, Software Interrupts and Exceptions, Programs and Processes – Role of Interrupts in Process State Transitions, I/O Device Interfaces – SCII, USB.

Unit 4- Pipelining& Multiprocessor: Basic Concepts of Pipelining, Throughput and Speedup, Instruction Pipeline, Pipeline Hazards, Introduction to Parallel Processors, Symmetric Shared Memory and Distributed Shared Memory Multiprocessors, Performance Issues of Symmetric and Distributed Shared Memory, Synchronization.

Unit 5- Memory Organization: Basic Concepts, Concept of Hierarchical Memory Organization, Main Memory: RAM, ROM, Speed, Size and cost, Cache Memory and its Mapping, Replacement Algorithms, Write Policies, Virtual Memory, Memory Management Requirements, Associative Memory, Secondary storage devices.

TEXTBOOKS:

1. William Stallings, Computer Organization and architecture, 11th edition (2022), Pearson Education.
2. David A. Patterson and John L. Hennessy "Computer Organization and Design: The Hardware/Software Interface" , 5th Edition, Elsevier.
3. M. Morris Mano, "Computer System Architecture", Third Edition, Pearson Education.

REFERENCE BOOKS:

1. Microprocessor Architecture, Programming, and Applications with the 8085 -Ramesh S. Gaonkar Pub: Penram International.
2. CarlHamacher" Computer Organization and EmbeddedSystems", 6th Edition, McGraw Hill HigherEducation.
3. Miles Murdoccaa and Vincent Heuring"Computer Architecture and Organization: An integrated Approach" 2nd edition,Wiley Publication.

JAVA PROGRAMMING (CST-008)

L:T:P:: 3:1:0 **Credits-04**

COURSE OBJECTIVES: The objectives of this course are to:

1. Understand Object Oriented Programming concepts and basic characteristics of Java.
2. Know the principles of packages, inheritance and interfaces.
3. Define exceptions and use I/O streams.
4. Develop a java application with threads and generics classes
5. Design and build simple Graphical User Interfaces.

COURSE OUTCOMES: On successful completion of the course, the student will be able to:

1. Write Java programs with properly designed constants, variables, objects, methods and reusability functionality
2. Learn how and where to implement interface and exception-handling concepts.
3. Write multi-threaded programming concepts for concurrency control based applications.
4. Construct GUI based JAVA enterprise applications
5. Develop web applications using JDBC, RMI and Servlet methodologies.

Unit 1- Java Basics and Inheritance: The Genesis of Java, Overview of Java, Data Types, Variables, and Arrays, Operators, Control Statements, Introducing Classes, Methods and Classes,Type Casting, String Handling, Abstract Class, Method overriding.

Inheritance: Basics, Using Super, Creating a Multilevel Hierarchy, Problem with Multiple Inheritance.

Unit 2- Packages, Interfaces and Exception Handling: Packages- Packages, Access Protection, Importing Packages,

Interfaces- Definition and Implementations,

Exception Handling- Types, Try and Catch, Throw and Finally statements.

Unit 3- Multi Threading and File Handling: Multithreaded Programming, Thread Life Cycle Creating Threads, Creating Multiple Threads, Thread Priorities, Synchronization, Inter Thread Communication, Suspending, Resuming and Stopping Threads.

File Handling: I/O Basics, Reading Console Input, Writing Console output, I/ O Classes and Interfaces.

Unit 4- Applets, Event Handling and AWT: Applet Basics, Applet Architecture, Applet Display Methods, Passing parameters to Applets,

Event Handling: Delegation Event Model, Event Classes, Event Listener Interfaces,
AWT: Working with Windows, Graphics, Colors and Fonts, Using AWT Controls, Layout Managers and Menus.

Unit 5- JDBC, RMI And Servlets: JDBC-JDBC Architecture, The Structured Query Language, JDBC Configuration, Executing SQL, RMI Architecture, A simple client/server application using RMI, **Servlets-** Life cycle of a Servlet, Servlet packages ,Handling HTTP Requests and Responses.

TEXTBOOKS:

1. Herbert Schildt, ―Java The complete reference, 8th Edition, McGraw Hill Education, 2011.
2. Cay S. Horstmann, Gary cornell, ―Core Java Volume –I Fundamentals, 9th Edition, Prentice Hall, 2013.

REFERENCES:

1. Paul Deitel, Harvey Deitel, ―Java SE 8 for programmers, 3rd Edition, Pearson, 2015.
2. Steven Holzner, ―Java 2 Black book, Dreamtech press, 2011.
3. Timothy Budd, ―Understanding Object-oriented programming with Java, Updated Edition, Pearson Education, 2000.

FORMAL LANGUAGES & AUTOMATA THEORY (CST-009)

L:T:P:: 3:1:0 **Credits-04**

COURSE OBJECTIVES: The objectives of this course are to:

1. Introduce the student to the concepts of theory of computation in computer science.
2. Acquire insights into the relationship among formal languages, formal grammars, and automata.
3. Learn to design automats and Turing machine.

COURSE OUTCOMES: On successful completion of the course, the student will be able to:

1. Apply the knowledge of automata theory, grammars & regular expressions for solving the problem.
2. Analyze the give automata, regular expression & grammar to know the language it represents.
3. Design Automata & Grammar for pattern recognition and syntax checking.
4. Distinguish between decidability and un-decidability of problems.
5. Identify limitations of some computational models and possible methods of proving them.

Unit 1- Introduction: Alphabets, Strings and Languages; Automata and Grammars, Deterministic finite Automata (DFA)- Formal Definition, Simplified notation: State transition graph, Transition table, Language of DFA, Nondeterministic finite Automata (NFA), NFA with epsilon transition, Language of NFA, Equivalence of NFA and DFA, Minimization of Finite Automata, Distinguishing one string from other, Myhill-Nerode Theorem

Unit 2- Regular Expressions: Definition, Operators of regular expression and their precedence, Algebraic laws for Regular expressions, Kleen's Theorem, Regular expression to FA, DFA to Regular expression, Arden Theorem, Non Regular Languages, Pumping Lemma for regular Languages. Application of Pumping Lemma, Closure properties of Regular Languages, Decision properties of Regular Languages, FA with output: Moore and Mealy machine, Equivalence of Moore and Mealy Machine, Applications and Limitation of FA.

Unit 3- Context-free languages and pushdown automata: Context-free grammars (CFG) and languages (CFL), Chomsky and Greibach normal forms, nondeterministic pushdown automata (PDA) and equivalence with CFG, parse trees, ambiguity in CFG, pumping lemma for context-free languages, deterministic pushdown automata, closure properties of CFLs.

Unit 4- Context-sensitive languages: Context-sensitive grammars (CSG) and languages, linear bounded automata and equivalence with CSG. Turing machines: The basic model for Turing machines (TM), Turing- recognizable (recursively enumerable) and Turing-decidable (recursive) languages and their closure properties, variants of Turing machines, nondeterministic TMs and equivalence with deterministic TMs, unrestricted grammars and equivalence with Turing machines, TMs as enumerators.

Unit 5- Types of Turing machine: Turing machines and halting Problem

Undecidability: Undecidability, A Language that is Not Recursively Enumerable, An Undecidable Problem That is RE, Undecidable Problems about Turing Machines, Recursive languages, Properties of recursive languages, Post's Correspondence Problem, Modified Post Correspondence problem, Other Undecidable Problems, Counter machines.

TEXTBOOKS:

1. Introduction to Automata Theory, Languages, and Computation, 3nd Edition, John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, Pearson Education.
2. Theory of Computer Science – Automata languages and computation, Mishra and Chandrashekaran, 2nd edition, PHI.

REFERENCE BOOKS:

1. Introduction to Languages and The Theory of Computation, John C Martin, TMH.
2. Introduction to Computer Theory, Daniel I.A. Cohen, John Wiley.
3. A Textbook on Automata Theory, P. K. Srimani, Nasir S. F. B, Cambridge University Press.
4. Introduction to the Theory of Computation, Michael Sipser, 3rd edition, Cengage Learning.
5. Introduction to Formal languages Automata Theory and Computation Kamala Krithivasan, Rama R, Pearson.

COMPUTER ORGANIZATION AND ARCHITECTURE LAB (CSP-007)

L:T:P:: 0:0:2 **Credits-01**

COURSE OBJECTIVES: The objectives of this course are to:

1. Understanding the behaviour of Logic Gates, Adders, Decoders, Multiplexers and Flip-Flops.
2. Understanding the behaviour of ALU, RAM, STACK and PROCESSOR from working modules and the modules designed by the student as part of the experiment.

COURSE OUTCOMES: On successful completion of the course, the student will be able to:

1. Recognize basic logic gates with IC chips.
2. Design combinational circuits using IC Chips.
3. Connect the theory of computer organization with hardware.
4. Implement the concept of adders
5. Apply fundamentals of digital design and extend the learning to design sequential circuits.

LIST OF EXPERIMENTS

1. Implementing HALF ADDER, FULL ADDER using basic logic gates.
2. Implementing Binary -to -Gray, Gray -to -Binary code conversions.
3. Implementing 3-8 line DECODER and Implementing 4x1 and 8x1 MULTIPLEXERS.
4. Verify the excitation tables of various FLIP-FLOPS.
5. Design of an 8-bit Input/ Output system with four 8-bit Internal Registers.
6. Design of an 8-bit ARITHMETIC LOGIC UNIT.
7. Design the data path of a computer from its register transfer language description.
8. Design the control unit of a computer using either hardwiring or microprogramming based on its register transfer language description.
9. Write an algorithm and program to perform matrix multiplication of two n * n matrices on the 2-D mesh SIMD model, Hypercube SIMD Model or multiprocessor system.
10. Study of Scalability for Single board Multi-board, multi-core, multiprocessor using Simulator.

JAVA PROGRAMMING LAB (CSP-008)

L:T:P:: 0:0:2 **Credits-01**

COURSE OBJECTIVES: The objectives of this course are to:

1. Write the program using abstract classes.
2. Write programs for solving real world problems using java collection framework
3. Write multithreaded program.

COURSE OUTCOMES: On successful completion of the course, the student will be able to:

1. Develop programs using object-oriented concepts, exception handling and multi-threading.
2. Demonstrate java features such as Inheritance, Interfaces, Polymorphism for different scenarios
3. Demonstrate java features such as Abstract class and method overriding
4. Design and implement data driven applications and assign responsibilities.
5. Develop web application using JDBC and Servlets

LIST OF EXPERIMENTS

1. Develop a java program to find the sum of odd and even numbers in an array.
2. Develop a java program to print the prime numbers between n1 to n2 using class, objects and methods.
3. Develop a program for calculating the age of a person and display the age in the form of years, months and days.
4. Demonstrate a program for method overloading. Consider the different types of transaction modes used for transferring money. (Credit card, Debit card, Net banking etc).
5. Create a Abstract class and calculate the area of different shapes by overriding methods.
6. Develop a Library application using multiple inheritances. Consider Book, Magazines and Journals as base classes and Library as derived class. In the Book class, perform the operations like Search Book, Issue Book, Return Book, Renew Book, and Fine Calculation. In the Magazines and Journals classes, perform issue and return operations.
7. Develop a program for banking application with exception handling. Handle the exceptions in following cases:
   a) Account balance <1000
   b) Withdrawal amount is greater than balance amount
   c) Transaction count exceeds 3
   d) One day transaction exceeds 1 lakh.
8. Create a student database and store the details of the students in a table. Perform the SELECT, INSERT, UPDATE and DELETE operations using JDBC connectivity.

9. Design a login page using servlets and validate the username and password by comparing the details stored in the database.

10. Mini project (Anyone)

(Front End: Java, Back End: Oracle, define classes for the application and assign responsibilities)

a) Central Library OPAC Engine

b) ATM Banking

c) Online Shopping

d) E-Ticketing System

e) Student Information Management System

f) City Info Browser

g) E-mail Server

UNIX/LINUX PROGRAMMING LAB (CSP-009)

L:T:P:: 0:0:2 **Credits-01**

COURSE OBJECTIVES: The objectives of this course are to:

1. Describe the basic file system in Linux and its file attributes.
2. Appraise different filters, process handling, regular expressions and network handling features using suitable commands.
3. Summarize different Linux commands to write Shell Programs.

COURSE OUTCOMES: On successful completion of the course, the student will be able to:

2. Demonstrate the basic knowledge of Linux commands and file-handling utilities by using a Linux shell environment.
3. Evaluate the concept of shell scripting programs by using AWK and SED commands.
4. Use tracing mechanisms for debugging.
5. Compile source code into an object and executable modules.
6. Use advanced network tools.

LIST OF EXPERIMENTS

1. Study of Unix/Linux general purpose utility command list (man, who, cat, cd, cp, ps, ls, mv, rm, mkdir, rmdir, echo, more, date, time, kill, history, chmod, chown, finger, pwd, cal, logout, shutdown etc.), vi editor, .bashrc, /etc/bashrc and environment variables.
2. Write a shell script program to: a) display list of user currently logged in; b) to copy contents of one file to another.
3. Write a program using sed command to print duplicated lines of Input.
4. Write a grep/egrep script to find the number of words character, words and lines in a file.
5. Write an awk script to: a). develop a Fibonacci series; b) display the pattern of given string or number.
6. Write a shell script program to a) display the process attributes; b) change priority of processes; c) change the ownership of processes; d)to send back a process from foreground ; e) to retrieve a process from background ; f) create a Zombie process
7. Write a program to create a child process and allow the parent to display “parent” and the child to display “child” on the screen
8. Write a makefile to compile a C program.
9. Study to execute programs using gdb to utilize its various features like breakpoints, conditional breakpoints. Also write a shell script program to include verbose and xtrace debug option for debugging
10. Study to use ssh, telent, putty, ftp, ncftp and other network tools.