Scheme & Syllabus for B.Tech (Computer Science & Engineering) (IInd Year) according to Choice Based Credit System (CBCS)

(Semester IIIrd and Semester IVth)

For Session 2019-20 onwards
(Batch 2018-2019 onwards)

Department of Computer Science & Engineering
School of Engineering & Technology

CENTRAL UNIVERSITY OF HARYANA
MAHENDERGARH-123031
HARYANA

Scheme –B. Tech. (CSE 2nd Year)
THIRD SEMESTER

CODE	SUBJECT	L	T	P	CREDIT	Theory	Practical	Internal Assessment	Total Marks
BT CS 301	Data Structures and Algorithms	3	1	-	4	70	-	30	100
BT MT 301	Mathematics-III	3	1	-	4	70	-	30	100
BT CS 302	Digital Electronics	3	-	-	3	70	-	30	100
	GEC -I*	3	1	-	4	70	-	30	100
BT HUM 304	Fundamentals of Management	3	-	-	3	70	-	30	100
BT CS 303	Computer Organization & Architecture	3	1	-	4	70	-	30	100
BT CS 304	Data Structures & Algorithms Lab	-	-	2	1.0	-	35	15	50
BT CS 305	Digital Electronics Lab	-	-	2	1.0	-	35	15	50
	Total	18	4	4	**24**	420	70	210	700

*GEC to be taken from other department

List of GEC for other department

CODE	GEC SUBJECTS
BT CS 301	Data Structure & Algorithms
BT CS 303	Computer Organization & Architecture

15/2/2019

FOURTH SEMESTER

CODE	SUBJECT	L	T	P	CREDIT	Theory	Practical	Internal Assessment	Total Marks
BT CS 401	Database Management System (DBMS)	3	1	-	4	70	-	30	100
BT CS 402	Object Oriented Programming using C++ (OOPS)	3	1	-	4	70	-	30	100
BT ECO 507A	Economics	3	-	-	3	70	-	30	100
BT AUD 308A	Environmental Sciences	3	-	-	-	70	-	30	100
BT CS 403	Discrete Structures	3	-	-	3	70	-	30	100
	GEC-II*	3	1	-	4	70	-	30	100
BT CS 404	DBMS Lab	-	-	2	1.0	-	35	15	50
BT CS 405	C++ Programming LAB	-	-	2	1.0	-	35	15	50
	Total	18	3	4	**20**	420	70	210	700

*GEC to be taken from other department

List of GEC for other department

CODE	GEC SUBJECTS
BT CS 402	Object Oriented Programming using C++ (OOPS)

Third Semester
Data Structures & Algorithms

BT CS 301

L - T - P
3 - 1 - 0

UNIT-1

Data structures and Algorithms: an overview: concept of data structure, choice of right data structures, types of data structures, basic terminology Algorithms, how to design and develop an algorithm: stepwise refinement, use of accumulators and counters; algorithm analysis, complexity of algorithms Big-oh notation.

Arrays: Searching Sorting: Introduction, One Dimensional Arrays, **Operations Defined:** traversal, selection, searching, insertion, deletion, and sorting. Multidimensional arrays, address calculation of a location in arrays.

Searching: Linear search, Recursive and Non recursive binary Search.

Sorting: Selection sort, Bubble sort, Insertion sort, Merge sort, Quick sort, Shell sort, Heap sort

UNIT-2

Stacks and queues: Stacks, array representation of stack, Applications of stacks. Queues, Circular queues, array representation of Queues, Deque, priority queues, Applications of Queues.

Pointers: Pointer variables, Pointer and arrays, array of pointers, pointers and structures, Dynamic allocation.

Linked Lists: Concept of a linked list,. Circular linked list, doubly linked list, operations on linked lists. Concepts of header linked lists. Applications of linked lists, linked stacks, linked Queues.

UNIT-3

Trees: Introduction to trees, binary trees, representation and traversal of trees, operations on binary trees, types of binary trees, threaded binary trees, B Trees, Application of trees.

Graphs: Introduction, terminology, 'set, linked and matrix' representation, Graph traversal techniques: BFS, DFS, operations on graphs, Minimum spanning trees, Applications of graphs

UNIT-4

Introduction to file handling, Data and Information, File concepts, File organization, files and streams, working with files. AVL trees, Sets, list representation of sets, applications of sets.

Text Books:

1 A.M. Tenenbaum, Langsam, Moshe J. Augentem , "Data Structures using C", Pearson
2 Reema Thareja ,"Data Structures using C", OUP India Pub.

Reference Books:

1 A.V. Aho, J.E. Hopcroft and T.D. Ullman,"Data Structures and Algorithms", Original edition, Addison-Wesley.
2 Ellis Horowitz & Sartaj Sahni,"Fundamentals of Data Structure", OBS Pub.
3 Introduction to Computers Science -An algorithms approach, Jean Paul Tremblay, Richard B. Bunt, T.M.H.
4 Lipschutz Seymour ."Data Structures with C (Schaum's Outline Series)", Mcgraw Hill

Mathematics-III

BT MT 301 L-T-P 3-1-0

UNIT-1

Multivariable Calculus (Differentiation): Limit, continuity and differentiability of functions of two or more variables. Partial differentiation, Euler's theorem on homogeneous functions, change of variables, chain rule, Taylor's theorem (two variables), approximate calculations, Jacobian, maxima and minima of two independent variables, Lagrange's method of multipliers.

UNIT-2

Ordinary differential Equations of first order and first degree: Exact, Reducible to Exact, linear and Bernoulli's equations. Linear differential Equations of higher order with constant coefficients, Cauchy – Euler Differential Equation, Legendre Linear Equation, Method of variation of parameters, Method of undetermined coefficient. Application of ordinary differential equations.

UNIT-3

Laplace Transform: Definition and existence of Laplace transform, Properties of Laplace Transform and formulae, Unit Step function, Dirac Delta function, Heaviside function, Laplace transform of periodic functions. Finding inverse Laplace transform by different methods, convolution theorem. Evaluation of integrals by Laplace transform, solving ODEs by Laplace transforms method.

UNIT-4

Fourier Series: Periodic functions, Fourier series, Euler's formula, Change of intervals, Half range sine and cosine series.

Fourier Transform: Fourier Complex, Sine and Cosine transform, properties and formulae, inverse Fourier transforms, Convolution theorem.

Text Books:

[1] E. Kreyszig, *Advanced Engineering Mathematics*, 10 edition. Hoboken, NJ: Wiley.

[2] B. Ramana, *Higher Engineering Mathematics*, 1st edition. New Delhi: McGraw Hill Education.

[3] B. S. Grewal, *Higher Engineering Mathematics 44th Edition*. Khanna Publishers.

[4] R. K. Jain and S. R. K. Iyengar, *Advanced Engineering Mathematics 4/e)*, Fourth edition edition. Oxford: Narosa Publishing House Pvt. Ltd. - New Delhi.

Reference Books:

[1] J. R. Hass, C. E. Heil, and M. D. Weir, *Thomas' Calculus*, 14 edition. Boston? Pearson.

[2] P. S. Das and C. Vijayakumari, *Engineering Mathematics*, First edition. Place of publication not identified: Pearson Education.

[3] J. Bird, *Higher Engineering Mathematics, 7th ed.*, 7 edition. Abingdon, Oxon ; New York, NY: Routledge.

[4] T. Veerarajan, *Engineering Mathematics*. McGraw Hill Education India Pvt Ltd.

5/20

Digital Electronics

BT CS 302

L- T- P
3 - 0 - 0

UNIT - 1

Binary Systems: Digital Systems, Binary Numbers, Number base conversions, Octal and Hexadecimal Numbers, complements, Signed binary numbers, Binary codes, Binary Storage and Registers, Binary logic.

Boolean Algebra And Logic Gates: Basic Definitions, Axiomatic definition of Boolean Algebra, Basic theorems and properties of Boolean algebra, Boolean functions canonical and standard forms, other logic operations, Digital logic gages, integrated circuits.

UNIT -2

Gate – Level Minimization: The map method, Four-variable map, Five-Variable map, product of sums simplification Don't-care conditions, NAND and NOR implementation other Two-level implementations, Exclusive – Or function, Hardware Description language (HDL).

Combinational Logic: Combinational Circuits, Analysis procedure Design procedure, Binary Adder-Subtractor, Decimal Adder, Binary multiplier, magnitude comparator, Decoders, Encoders, Multiplexers, HDL for combinational circuits.

UNIT -3

Synchronous Sequential Logic: Sequential circuits, latches, Flip-Flops Analysis of clocked sequential circuits, HDL for sequential circuits, State Reduction and Assignment, Design Procedure.

Registers and Counters: Registers, shift Registers, Ripple counters synchronous counters, other counters, HDL for Registers and counters.

UNIT -4

Memory, CPLDs, and FPGAs: Introduction, Random-Access Memory, Memory Decoding, Error Detection and correction Read-only memory, Programmable logic Array programmable Array logic, Sequential Programmable Devices.

Asynchronous Sequential Logic: Introduction, Analysis Procedure, Circuits with Latches, Design Procedure, Reduction of state and Flow Tables, Race-Free state Assignment Hazards, Design Example.

Text Books :

1. JAIN, *MODERN DIGITAL ELECTRONICS 4E*. Tata McGraw-Hill Education.
2. Mano and Ciletti, *Digital Design*. Pearson Education India.
3. Gill Nasib Singh and Dixit J.B., Digital Design and Computer Organisation, University Science Press (An Imprint of Laxmi Publications Pvt. Ltd.), New Delhi.
4. M. M. Morris, *Computer System Architecture 3e (Update) by Pearson*, Third edition. Pearson Education.

Reference Books :

1. H. Taub and D. L. Schilling, *Digital Integrated Electronics*. McGraw-Hill.
2. A. P. Malvino and D. P. Leach, *Digital principles and applications*, McGraw-Hill.
3. W. Stallings, *Computer Organization and Architecture: Designing for Performance (ninth Edition)*.
4. D. A. Patterson and J. L. Hennessy, *Computer Organization and Design: The Hardware/software Interface*. Morgan Kaufmann.

Fundamentals of Management

BT HUM 304

L-T-P
3-0-0

Unit-1
Meaning of management, Definitions of Management, Characteristics of management, Importance of Management; Management as Art, Science and Profession; Development of Management thoughts-- Fayol's principles of Management, Taylors Scientific Management, Elton Mayo's Human Relations School, System's Approach to Management; Principles of Management, Management Processes- Planning, Organizing, Staffing, Leading and Controlling; Delegation and Decentralization.

Unit-2
Production Management : Definition, Objectives, Functions and Scope, Production Planning and Control; its significance, stages in production planning and control. Brief introduction to the concepts of material management, inventory control; its importance and various methods.

Unit-3
Marketing Management - Definition of marketing, Marketing concept, objectives & Functions of marketing. Marketing Research - Meaning; Definition; objectives; Importance; Limitations; Process. Advertising - meaning of advertising, objectives, functions, criticism.

Unit-4
An Introduction of Financial Management, Objectives of Financial Management, Functions and Importance of Financial Management, Role of Financial Manager. Brief Introduction to the concept of capital structure and various sources of finance.

Text Books:

1. Robins S.P. and Couiter M., Management, Prentice Hall India, 10th ed.
2. Stoner JAF, Freeman RE and Gilbert DR, Management, 6th ed., Pearson Education.
3. Tripathy PC & Reddy PN, Principles of Management, Tata McGraw Hill.

Reference Books:

1. Pandey I. P., Financial Management (11th ed.). New Delhi: Vikas Publishing House.
2. Kotler, P., Keller K. L. Marketing Management (5th ed.). Pearson.
3. Robbins, S. P., DeCenzo, D., Agarwal, M. N., & Bhattacharyya, S. Essentials of Management (6 ed.). New Delhi: Pearson Edcuation.

BT CS 303 Computer Organization & Architecture L-T-P 3 -1 -0

UNIT-1

Introduction: Historical overview, economic trends, underlying technologies, Data Representation-Data Types, Complements, Fixed-Point Representation, Floating-Point Representation, Error Detection and Correction, Addition, Subtraction, Multiplication and Division algorithms and hardware.

Register Transfer and Micro operations: Register transfer language, Inter-Register Transfer, Arithmetic Micro-operations, Logic and Shift micro-operations Language, Control functions.

UNIT-2

Arithmetic Logic Unit: Arithmetic, logic and shift micro operations, Constructing an arithmetic logic shift unit

Basic Computer Architecture and Design: Computer registers, Computer Instructions-Instruction Set Completeness, Classifying Instruction Set Architecture, Basic steps of Instruction Execution, Hardwired Control, Micro programmed Control, Horizontal and Vertical Microprogramming, Interrupts.

Central Processing Unit: General Register Organization, Stack Organized CPU, Instruction Formats, Addressing Modes, Data Transfer and Manipulation, RISC Vs CISC.

UNIT-3

Pipelining: Parallel and pipeline Processing, Pipeline Control, Pipeline Implementations, Conflicts Resolution, and Pipeline Hazards, Vector Processing, and Array Processors.

Memory Organization: Memory Systems: principle of locality, principles of memory hierarchy Caches, associative memory, main memory, Virtual memory, Paging and Segmentation, Memory Interleaving.

UNIT-4

Input Output Organization: I/O performance measures, types and characteristics of I/O devices, I/O Modes-Programmed I/O, Interrupt Initiated I/O and DMA.Buses: connecting I/O devices to processor and memory, interfacing I/O devices to memory, processor, and operating system.

Parallel Computers: Classification, SIMD, MIMD Organizations, Connection Networks, Data Flow Machines, and Multithreaded Architectures.

Text Books :

1. D. A. Patterson and J. L. Hennessy, *Computer Organization and Design MIPS Edition: The Hardware/Software Interface*, Newnes.
2. C. Hamacher and Z. Vranesic, *Computer Organization 5th Edition*. .
3. Gill Nasib Singh and Dixit J.B., Digital Design and Computer Organisation, University Science Press (An Imprint of Laxmi Publications Pvt. Ltd.), New Delhi.

Reference Books :

1. W. Stallings, *Computer Organization and Architecture: Designing for Performance*, Pearson-Prentice Hall.
2. Zacker, *Networking: The Complete Reference*, Tata McGraw-Hill Education.
3. J. P. Hayes, *Computer Architecture and Organization*, WCB/McGraw-Hill

Data Structure & Algorithms Lab

BT CS 304

L - T -P
0 - 0 - 2

1. Write a program to search an element in a two-dimensional array using linear search.
2. Using iteration & recursion concepts write programs for finding the element in the array using Binary Search Method
3. Write a program to perform following operations on tables using functions only a) Addition b) Subtraction c) Multiplication d) Transpose
4. Using iteration & recursion concepts write the programs for Quick Sort Technique
5. Write a program to implement the various operations on string such as length of string concatenation, reverse of a string & copy of a string to another.
6. Write a program for swapping of two numbers using 'call by value' and 'call by reference strategies.
7. Write a program to implement binary search tree. (Insertion and Deletion in Binary search Tree)
8. Write a program to create a linked list & perform operations such as insert, delete, update, reverse in the link list
9. Write the program for implementation of a file and performing operations such as insert, delete, update a record in the file.
10. Create a linked list and perform the following operations on it
    a) add a node b) Delete a node
11. Write a program to simulate the various searching & sorting algorithms and compare their timings for a list of 1000 elements.
12. Write a program to simulate the various graph traversing algorithms.
13. Write a program which simulates the various tree traversal algorithms.

9/20

BT CS 305 **Digital Electronics Lab** **L-T-P**
0- 0- 2

1. Study of TTL gates – AND, OR, NOT, NAND, NOR, EX-OR, EX-NOR.
2. Design & realize a given function using K-maps and verify its performance.
3. Design of half adder and full adder using NAND gates.
4. To verify the operation of multiplexer & Demultiplexer.
5. To verify the operation of comparator.
6. To verify the truth tables of S-R, J-K, T & D type flip flops.
7. Set up R-S & JK flip flops using NAND Gates.
8. To verify the operation of bi-directional shift register.
9. To design & verify the operation of 3-bit synchronous counter.
10. To design and verify the operation of synchronous UP/DOWN decade counter using J K flip-flops & drive a seven-segment display using the same.
11. To design and verify the operation of asynchronous UP/DOWN decade counter using J K flip-flops & drive a seven-segment display using the same.
12. Study of MUX & DeMUX Circuits and ICs

NOTE: At least ten experiments are to be performed; atleast seven experiments should be performed from above list. Remaining three experiments may either be performed from the above list or designed & set by the concerned institution as per the scope of the syllabus.

Fourth Semester

Database Management Systems L-T-P 3- 1- 0

BT CS 401

UNIT-1

Introduction: An overview of database management system, database system Vs file system, Database system concepts and architecture, data models schema and instances, data independence and data base language and interfaces, Data definitions language, DML, Overall Database Structure.

Data Modeling using the Entity Relationship Model:
ER model concepts, notation for ER diagram, mapping constraints, keys, Concepts of Super Key, candidate key, primary key, Generalization, aggregation, reduction of an ER diagrams to tables, extended ER model.

UNIT-2

Relational data Model and Language: Relational data model concepts, integrity constraints: entity integrity, referential integrity, Keys constraints, Domain constraints, relational algebra, relational calculus, tuple and domain calculus.

Introduction to SQL: Characteristics of SQL. Advantage of SQL. SQL data types and literals. Types of SQL commands. SQL operators and their procedure. Tables, views and indexes. Queries and sub queries. Aggregate functions. Insert, update and delete operations. Joins, Unions, Intersection, Minus, Cursors in SQL.

UNIT-3

Data Base Design & Normalization: Functional dependencies, normal forms, first, second, third normal forms, BCNF, inclusion dependences, loss less join decompositions, normalization using FD, MVD, and JDs, alternative approaches to database design.

Transaction Processing Concepts: Transaction system, Testing of serializability, Serializability of schedules, conflict & view serializable schedule, recoverability, Recovery from transaction failures, log based recovery, checkpoints, deadlock handling.

UNIT-4

Crash Recovery: Failure classification, recovery concepts based on deferred update, recovery concepts based on intermediate update, shadow paging, check points, on-line backup during database updates

Concurrency Control Techniques: Concurrency control, locking Techniques for concurrency control, Time stamping protocols for concurrency control, validation based protocol, multiple granularity, Multi version schemes, Recovery with concurrent transaction.

Text Books:

1. A.Silberschatz, H.F. Korth and S. Sudarshan,"Database System Concepts" 6th edition, McGraw-Hill, International Edition.
2. Bipin Desai,"Introduction to Database Management system" Revised Edition, Galgotia Pub.
3. Shyamkant B. Navathe,"Fundamental of Database systems" 7th Edition, Pearson.

17/20

Reference Books:

1. R.Elmasri and S.B. Navathe,"Fundamentals of Database Systems" 3rd edition, Addision-Wesley, Low Priced Edition.
2. C.J. Date,"An Introduction to Database Systems", 8th edition, Pearson.
3. C.J.Date and S. Swamynathan, "An Introduction to database system",8th Edition, Pearson Education.
4. Shio Kumar Singh,"Database System Concepts, designs and applications", 2nd Edition, Pearson Education.

Object Oriented Programming using C++

L-T-P
3-1-0

BT CS 402

UNIT-1

Object oriented thinking: Need for OOP Paradigm. Procedural programming vs object oriented programming, object oriented concepts.

Functions: Main function, function prototyping, inline functions, reference variables, call by reference, Defaults arguments, Function overloading, Math library functions.

UNIT-2

Class: Difference between C structure and class, specifying a class. Defining member functions: inside and outside class, scope resolution operator, Array within a class, array of objects, Static data members and member functions, Object as function arguments, Returning objects, Friend function, memory allocation for objects, Pointer to Members, pointer to object, This pointer local classes.

Constructor and destructor: Constructor, Types of constructors: Default, Parameterized and Copy constructor, Constructor overloading, constructor with default parameter, Dynamic initialization of objects, Destructor

UNIT-3

Operator overloading and Type Conversion: Defining operator overloading, overloading unary and binary operator, Data Conversion: Basic to User Defined , User defined to basic, Conversion from one user-defined to other.

Inheritance and polymorphism: Base class, derived class, visibility modes, derivation and friendship, Types of inheritance, Containership, virtual function binding, pure virtual functions, Abstract class, pointer to derived class.

UNIT-4

Console IO operations: C++ stream classes, Unformatted IO operations, formatted IO operations, managing output with manipulators.

Working with files: Classes for file stream operations, opening and closing files, detection of File opening modes, file Pointers, Error handling during file operations, command line arguments. Templates: Class template, class template with parameter, function template, function template with parameter.

Text Books and Reference Books:

1. Bjrane Stroustrup, "C++ Programming language",3rd edition, Pearson education Asia
2. Lafore R."Object oriented Programming in C++",4th Ed. Techmedia, New Delhi.
3. Yashwant Kenetkar,"Let us C++",1stEd.,Oxford University Press.
4. B.A. Forouzan and R.F. Gilberg,CompilerScience,"A structured approach using C++" Cengage Learning, New Delhi.

13/20

BT- ECO-507A | Economics | L-T-P 3-0-0

Unit-1

Introduction to Economics

Definition of Economics-various definitions, circular flow of economic activity, Production possibility curve Economic laws and their nature. Relation between Science, Engineering, Technology and Economics. Concepts and measurement of utility, Law of Diminishing Marginal Utility, Law of equi-marginal utility - its practical application and importance, the concept of equilibrium

Unit-2

Market Demand

Meaning of Demand, Individual and Market demand schedule, Law of demand, shape of demand curve, Elasticity of demand, measurement of elasticity of demand, factors effecting elasticity of demand, practical importance & applications of the concept of elasticity of demand, the indifference curve theory, consumers surplus

Unit-3

Various Concepts of Cost

Objective of business firm, Meaning of production and factors of production; Law of variable proportions, Returns to scale, Internal and External economics and diseconomies of scale. Various concepts of cost - Fixed cost, variable cost, average cost, marginal cost, money cost, real cost opportunity cost. Shape of average cost, marginal cost, total cost etc. in short run and long run.

Unit-4

Types of Market

Meaning of Market, Types of Market - Perfect Competition, Monopoly, Oligoply, Monoplistic Competition (Main features of these markets) Supply and Law of Supply, Role of Demand & Supply in Price Determination and effect of changes in demand and supply on prices.

Nature and characteristics of Indian economy (brief and elementary introduction), national income concept, Privatization - meaning, merits and demerits, Balance of payment, Globalization of Indian economy - merits and demerits. Elementary Concepts of VAT, WTO, GATT & TRIPS agreement, IMF, World Bank.

Text Books:

1. P.N. Chopra,"Principles of Economics", (Kalyani Publishers).
2. K.K. Dewett," Modern Economic Theory" (S.Chand).
3. Stonier and Hague,"A Text Book of Economic Theory", (Longman's Landon)

Reference Books:

1. M.L. Jhingan," Micro Economic Theory", (S.Chand)
2. H.L. Ahuja,"Micro Economic Theory", (S.Chand)
3. S.K. Mishra,"Modern Micro Economics", (Pragati Publications) .
4. Mishra &Puri ,"Indian Economy".

Environmental Sciences

BT AUD 308A

L-T-P

3- 0- 0

UNIT-1

Introduction to Environmental Science and Natural Resources

The multidisciplinary nature of Environmental Studies. Definition, scope and importance, need for public awareness

Renewable and non-renewable resources: Land resources: Land as a resource, land degradation, soil erosion and desertification. Forest resources: Use and over-exploitation, deforestation, case studies. Water resources: Use and over-utilization of surface and ground water

UNIT-2

Ecosystems, Biodiversity and its Conservation

Concept of an ecosystem. Structure and function of an ecosystem. Energy flow in the ecosystem. Food chains, food webs and ecological pyramids.

Definition: genetic, species and ecosystem diversity. Bio-geographical classification of India. Hot-spots of biodiversity. Threats to biodiversity. Endangered and endemic species of India. Conservation of biodiversity.

UNIT-3

Environmental Pollution, Environment policies & laws

Definition, Causes, effects and control measures of: (a) Air pollution (b) Water pollution (c) Soil pollution (d) Marine pollution (e) Noise pollution (f) Nuclear hazards. Solid waste management. Pollution case studies.

Environment Protection Act. Air (Prevention and Control of Pollution) Act. Water (Prevention and Control of Pollution) Act. Wildlife Protection Act. Forest Conservation Act. Issues involved in enforcement of environmental legislation. Public awareness.

UNIT-4

Human Population and Environment and Fieldwork

Human population growth. Impacts on environment, human health and welfare. Environmental Movements: Chipko, silent valley, Bishnois of Rajasthan.

Visit to a local area to document environmental assets—river/forest/grassland/hill/ mountain. Visit to a local polluted site—Urban/Rural/Industrial/Agricultural.Study of common plants, insects, birds. Study of simple ecosystems—pond, river, hill slopes, etc.

Text Books and Reference books:

1. Cunningham, W.P. and Saigo, B.W., Environmental Science. W.M.C. Brown Publishers, New York, USA.
2. Enger, D.E. and Smith B.F., Environment Science–A Study of Interrelationships. W.M.C. Brown Publishers, New York, USA.
3. Gupta, P.K., Elements of Biotechnology, Rastogi Publications, Meerut.

15/20

4. Negi, B.S., Geography of Resources, Kedar Nath Ram Nath, Meerut.
5. Odum, E.P., Fundamentals of Ecology, Natraj Publishers, Dehradun.
6. Kaushik A and Kaushik C P. Perspectives in Environmental Studies, New age International Publishers, New Delhi.
7. Rastogi, V.B., Environmental Biology and Biochemistry, Kedar Nath Ram Nath, Meerut and Delhi.

Discrete Structures

L-T-P
3- 0-0

BT CS 403

UNIT-1

Set Theory: Algebra of sets. Classes of sets. Power Sets. Multi sets. Cartesian Product. Representation of relations. Types of relation. Equivalence relations and partitions. Partial ordering relations.

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

UNIT-2

Propositional Calculus: Basic operations: AND (^), OR (v), NOT (~), Implication and bi-implication. Truth value of a compound statement. propositions. tautologies. contradictions. Universal and Existential quantifiers, methods of proof. Mathematical Induction, Propositional logic, Hypothesis and Inference.

Techniques of Counting: Permutations with and without repetition, Combination. Pigeonhole principle.

UNIT-3

Algebric Structures: Definition and examples of a monoid, Semigroup, Groups and rings. Homomorphism, Isomorphism and Automorphism, Subgroups and Normal subgroups, Cyclic groups, Integral domain and fields, Cosets, Lagrange's theorem.

UNIT-4

Graphs: Introduction to graphs, Directed and Undirected graphs, Homomorphic and Isomorphic graphs. Subgraphs. Cut points and Bridges, Multigraph and Weighted graph. Paths and circuits. Shortest path in weighted graphs, Eurelian path and circuits. Hamilton paths and circuits, Planar graphs, Euler's formula, Coloring graph problem, Bipartite graphs, Travelling salesman problem, Trees, Spanning trees.

Text Books:

1 B. Kolman, R. Busby, and S. C. Ross, *Discrete Mathematical Structures*, 6 edition. Upper Saddle River, NJ: Pearson.

2 C.L. Liu. and Mohapatra, *Elements of Discrete Mathematics: A Computer Oriented Approach*, 4th edition. New York: Mc Graw Hill India.

3 B. Ram, *Discrete Mathematics*, 1 edition. Pearson.

4 J.-P. Tremblay and R. Manohar, *Discrete Mathematical Structures with Applications to Computer Science*, 1 edition. New Delhi, India: McGraw Hill Education.

17/20

Reference Books:

1. J. L. Gersting, *Mathematical Structures for Computer Science*, 6th Edition edition. New York: W. H. Freeman.
2. S. Lipschutz, M. L. Lipson, and V. H. Patil, *Discrete Mathematics*, Revised Third edition. McGraw Hill Education.
3. R. Johnsonbaugh, *Discrete Mathematics*, 7 edition. Upper Saddle River, N.J: Pearson.
4. A. Doerr and K. Levasseur, *Applied Discrete Structures for Computer Science*. Subsequent edition. New York: Macmillan Coll Div.

Database Management System Lab

L-T-P
0- 0- 2

BT CS 404

1. Write the queries for Data Definition and Data Manipulation language.
2. Write SQL queries using Logical operators (=,<,>,etc.).
3. Write SQL queries using SQL operators (Between.... AND, IN(List), Like, ISNULL and also with negating expressions).
4. Write SQL query using character, number, date and group functions.
5. Write SQL queries for Relational Algebra (UNION, INTERSECT, and MINUS, etc.).
6. Write SQL queries for extracting data from more than one table (Equi-Join, Non-Equi-Join , Outer Join)
7. Write SQL queries for sub queries, nested queries.
8. Write programs by the use of PL/SQL.
9. Concepts for ROLL BACK, COMMIT & CHECK POINTS.
10. Create VIEWS, CURSORS, and TRIGGRS & write ASSERTIONS.
11. Create FORMS and REPORTS.
12. Creation, altering and droping of tables and inserting rows into a table (use constraints while creating tables) examples using SELECT command.
13. Queries (along with sub Queries) using ANY, ALL, IN, EXISTS, NOTEXISTS, UNION, INTERSET, Constraints. Example:- Select the roll number and name of the student who secured fourth rank in the class.
14. Queries using Aggregate functions (COUNT, SUM, AVG, MAX and MIN), GROUP BY, HAVING and Creation and dropping of Views.
15. Queries using Conversion functions (to_char, to_number and to_date), string functions (Concatenation, lpad, rpad, ltrim, rtrim, lower, upper, initcap, length, substr and instr), date functions (Sysdate, next_day, add_months, last_day, months_between, least, greatest, trunc, round, to_char, to_date)
16. i) Creation of simple PL/SQL program which includes declaration section, executable section and exception –Handling section (Ex. Student marks can be selected from the table and printed for those who secured first class and an exception can be raised if no records were found)
ii) Insert data into student table and use COMMIT, ROLLBACK and SAVEPOINT in PL/SQL block.
17. Develop a program that includes the features NESTED IF, CASE and CASE expression. The program can be extended using the NULLIF and COALESCE functions.
18. Program development using WHILE LOOPS, numeric FOR LOOPS, nested loops using ERROR Handling, BUILT –IN Exceptions, USE defined Exceptions, RAISE- APPLICATION ERROR.
20. Program development using creation of stored functions, invoke functions in SQL Statements and write complex functions.
21. Program development using creation of package specification, package bodies, private objects, package variables and cursors and calling stored packages.
22. Develop programs using features parameters in a CURSOR, FOR UPDATE CURSOR, WHERE CURRENT of clause and CURSOR variables.

19/20

BT CS 405 **C++ Programming Lab** **L-T-P**
0- 0- 2

1. Write a program to read a matrix of size m x n form the keyboard and display the same using function.
2. Program to make the use of inline function.
3. Write a function power () which raise a number m to a power n. The function takes double value of m and integer value of n and returns the result. Use a default value of n is 2 to make the function to calculate squares when this argument is omitted.
4. Program to show that the effect of default arguments can be alternatively achieved by overloading.
5. Write a class ACCOUNT that represents your bank account and then use it.
The class should allow you to deposit money, withdraw money, calculate interest,
send you a message if you have insufficient balance.
6. Write a class STRING that can be used to store strings, add strings, equate string, output strings.
7. Create the class TIME to store time in hours and minutes. Write a friend function to add two TIME objects.
8. Create two classes DM and DB. DM stores the distance in meter and centimeters and DB stores the distance in feet and inches. Write a program two add object of DM with the object of DB class.
9. Write a program to create an abstract class named Shape that contains an empty method named number Of Sides ( ).Provide three classes named Trapezoid, Triangle and Hexagon such that each one of the classes inherits the class Shape. Each one of the classes contains only the method number Of Sides ( ) that shows the number of sides in the given geometrical figures.
10. Program to demonstrate the concept of:
a. Default constructor
b. Parameterized constructor
c. Copy constructor
d. Constructor overloading
11. Program to demonstrate the concept of destructor.
12. Program to show multiple inheritance
13. Program to show multilevel inheritance
14. Program to show hybrid inheritance
15. Program to show the concept of containership.
16. Program to overload unary operator.
17. Program to overload binary operator
18. Program to show the concept of run time polymorphism using virtual function.
19. Program to work with formatted and unformatted IO operations.
20. Program to read the name and roll numbers of students from keyboard and write them into a file and then display it.
21. Program to copy one file onto the end of another, adding line numbers
22. Write a function template for finding the minimum value contained in an array.

Scheme and Syllabus for B. Tech. (Computer Science and Engineering) (3rd Year)

(Semester 5th and Semester 6th)

W.e.f. Session 2020-21

Department of Computer Science and Engineering
School of Engineering and Technology

CENTRAL UNIVERSITY OF HARYANA
MAHENDERGARH-123031
HARYANA

Bunay Kr Ray Sangeeta

Scheme B.Tech. (CSE-3rd Year)

Fifth Semester

Code	Subject	L	T	P	Credit
BT CS 501A	Analysis and Design of Algorithms	3	1	0	4
BT CS 502A	Theory of Computation	3	1	0	4
BT CS 503A	Computer Network	3	0	0	3
BT CS 504A	Microprocessor and Interfacing	3	0	0	3
BT CS 505A	Analysis and Design of Algorithms Laboratory	0	0	2	1
BT CS 506A	Computer Network Laboratory	0	0	2	1
BT CS 507A	Microprocessor and Interfacing Laboratory	0	0	2	1
BT CS 508A	Summer Training *	0	2	0	2
	DCEC-1				
BT CS 521	Programming with Python	3	0	0	3
BT CS 522	Mobile Application Development	3	0	0	3
BT CS 523	Programming Concept using Java	3	0	0	3
	DCEC-1 Laboratory				
BT CS 526	Programming with Python Laboratory	0	0	2	1
BT CS 527	Mobile Application Development Laboratory	0	0	2	1
BT CS 528	Programming Concept using Java Laboratory	0	0	2	1
	Total	15	4	8	23

* Students will undergo six weeks summer training during summer vacation after the 4th semester and its evaluation will be carried out in 5th semester

Scheme B.Tech. (CSE-3rd Year)

Sixth Semester

Code	Subject	L	T	P	Credit
BT CS 601A	Principle of Operating System	3	0	0	3
BT CS 602A	Compiler Design	3	0	0	3
BT CS 603A	Software Engineering	3	0	0	3
BT CS 604A	Data Warehousing and Data Mining	3	0	0	3
BT CS 605A	Principle of Operating System Laboratory	0	0	2	1
BT CS 606A	Software Engineering Laboratory	0	0	2	1
BT CS 607A	Seminar	0	2	0	2
	DCEC-2				
BT CS 621	Computer Graphics	3	0	0	3
BT CS 622	Unix and Linux Programming	3	0	0	3
	DCEC-2 Laboratory(As per DCEC-2)				
BT CS 626	Computer Graphics Laboratory	0	0	2	1
BT CS 627	Unix and Linux Programming Laboratory	0	0	2	1
	DCEC-3				
BT CS 631	Stochastic Processes and Queueing Theory	3	0	0	3
BT CS 632	Wireless Communication	3	0	0	3
BT CS 633	Distributed Systems	3	0	0	3
BT CS 634	Unified Modeling Language and Agile Technology	3	0	0	3
	Total	18	2	6	23

* Students will undergo six weeks summer training during summer vacation after the 6th semester and its evaluation will be carried out in 7th semester

Analysis and Design of Algorithm
Course Code: BT CS 501A

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

Course Level Learning Outcomes

After successful completion of the course, the student will be able to:

- analyze the time and space complexity of algorithm
- develop systematically an algorithm for a problem
- apply the algorithms in solving real world problems

Unit-1

Introduction: Algorithms, Analysis of Algorithms, Design of Algorithms, Complexity of Algorithms, Asymptotic Notations, Growth of function, Recurrences and their solution methods.

Divide and Conquer: General method, Binary Search, Exponentiation problem, Merge Sort, Quick Sort, Selection Sort, Strassen's Matrix Multiplication algorithms and analysis of algorithms for these problems.

Unit-2

Greedy Method: General method, Knapsack Problem, Job sequencing with deadlines, Minimum Spanning trees, Single source shortest path and analysis of these algorithms.

Unit-3

Dynamic Programming: General method, 0/1 Knapsack problem, Optimal BST, All Pairs shortest path, Traveling Salesman Problem, Longest common subsequence (LCS).

Back Tracking: General method, 8 Queen's problem, Graph Coloring, Hamiltonian cycles and analysis of these problems.

Unit-4

Primality testing, Integer factorization, Randomized algorithms, Probabilistic algorithms. String Matching algorithms: Rabin Karp, KMP, Boyer Moore. Introduction to problem classes – NP, NPC, NP-Hard.

Suggested Readings:

1. Aho, A.V., Hopcroft, J.E. and Ullman, J.D., *The Design and Analysis of Algorithms*, 2nd Edition, Pearson, 2009.
2. Cormen, T.H., Leiserson, C.E., Rivest, R.L. and Stein, C., *Introduction to Algorithms*, 3rd Edition, PHI, 2010.
3. Goodrich, M.T., and Tamassia, R., *Algorithm Design: Foundation, Analysis and Internet Examples*, 1st Edition, John Wiley & Sons, 2006.
4. Horowitz, E., and Sahni, S., *Fundamental of Computer Algorithms*, 2nd Edition, Universities Press, 2008.
5. Kleinberg, J., and Tardos, E., *Algorithm Design*, 1st Edition, Pearson, 2013.

Theory of Computation
Course Code: BT CS 502A

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

Course Level Learning Outcomes

After successful completion of the course, the student will be able to:

- relate practical problems to languages, automata, and computability
- demonstrate an increased level of mathematical sophistication
- apply mathematical and formal techniques for solving problems

Unit -1: Finite Automata and Regular Languages

Finite automata, DFA, NFA, Equivalence of NDFA and DFA, FA with null moves, Myhill-Nerode Theorem and minimization of FA.

FA with outputs: Moore and mealy Machines, Equivalence of Moore and Mealy machines.

Regular Expressions, Regular languages and FA, Arden theorem: Equivalence of FA and Regular Expressions, Closure properties of regular sets, Pumping Lemma for Regular Sets, Applications of the pumping lemma.

Unit 2: Chomsky classification of Grammars

Unrestricted languages, Context sensitive languages, Context free languages and regular languages, Relation between classes of languages, Parse trees, Ambiguity in CFG, Reduced forms, Removal of useless Symbols and unit production, Chomsky Normal Form (CNF), Greibach Normal Form (GNF).

Unit 3: Pushdown Automata and Turing machines

Basic Structure of PDA, Acceptance by PDA, PDA and Context Free Languages, Design of PDA corresponding to a grammar, LL(k) parsing, LBA.

Basic structure and working of Turing Machine, Nondeterministic and deterministic Turing Machine, Role of TM as language recognizer, Space and time complexity of TM, binary coding of TM, Universal Turing Machine, halting problem of TM, Multitrack and Multitape TM.

Unit 4: Undecidability and Advancement in Automata Theory

Decidability: Decidable languages, decidable problems concerning Context free languages.

Recursive and recursive enumerable languages, Post Correspondence Problem, A simple undecidable problem (PCP), Primitive recursive functions.

Suggested Readings:

1. Cohen, D.I., *Introduction to Computer Theory*, 2nd Edition, Wiley India, 2007.
2. Hopcroft, J.E., Motwani, R. and Ullman, J.D., *Introduction to Automata Theory Language and Computation*, Pearson Education, 2008.
3. Martin, J.C., *Introduction to Languages and The Theory of Computation*, 4th Edition, Tata McGraw Hill, 2010.

4. Mishra, K.L.P., and Chandrasekaran, N., *Theory of Computer Science: Automata Language and Computation*, 3rd Edition, Prentice Hall of India, 2006.
5. Nagpal, C.K., *Formal Languages and Automata Theory*, Oxford University Press, 2012.

Computer Network
Course Code: BT CS 503A

L T P **Credits: 03**
3 0 0

Course Level Learning Outcomes

After successful completion of the course, the student will be able to:

- gain insight about basic network theory and layered communication architectures
- provide solutions to various problems in network theory
- conceptualize and design a network stack

Unit-1: Computer Networks, the Internet, and Application Layer

About The Internet its Protocols, the Network Edge, the Network Core, Access Networks and Physical Media (Transmission Media, Guided Media and Unguided Media), Switching technique (Circuit Switching, Message Switching, Packet Switching), Delay and Loss in Packet-Switched Networks, Protocol Layers and Their Service Models (Layered Architecture, function of the layers, OSI & TCP/IP Reference model).

Principles of Application-Layer Protocols, the World Wide Web: HTTP, File Transfer: FTP, Electronic Mail in the Internet: SMTP, the Internet's Directory Service: DNS,DHCP.

Unit-2: Transport Layer

Transport-Layer Services and Principles, Multiplexing and Demultiplexing, Connectionless Transport: UDP, Principles of Reliable of Data Transfer, Connection-Oriented Transport: TCP, Principles of Congestion Control, TCP Congestion Control.

Unit-3: Network Layer and Routing

Introduction and Network Service Model, Internetworking & devices: Repeaters, Hubs, Bridges, Switches, Router, Gateway; Addressing: IPv4 address, classful address, subnetting, classless address, Routing : techniques, static vs. dynamic routing, routing table; Routing algorithms: shortest path algorithm, distance vector routing, link state routing; Hierarchical Routing.

Unit-4: Link Layer and Local Area Networks

The Data Link Layer: Introduction, Services, Flow Control: Stop and wait Protocols, Sliding Window Protocols; Error Detection and Correction, Multiple Access Protocols (Pure ALOHA & Slotted ALOHA, CSMA, CSMA/CD) and LANs, LAN Addresses and ARP, Ethernet, Point-to-Point Protocol, ATM, X.25 and Frame Relay.

Suggested Readings:

1. Forouzan, B.A., *Data Communication and Networking*, 4th Edition, Tata McGraw Hill, 2017.
2. Keshav, S., *An Engineering Approach to Computer Network*, Pearson Education, 2002.
3. Kurose, J., and Ross, K., *Computer Networking – A Top-Down Approach Featuring the Internet*, 7th Edition, Pearson Education, 2016.
4. Stallings, W., *Data and Computer Communications*, 10th Edition, Pearson Education, 2013
5. Tanenbaum, A.S., *Computer Networks*, 5th Edition, Pearson Education, 2010.

Microprocessor and Interfacing
Course Code: BT CS 504A

L T P **Credits: 03**
3 0 0

Course Level Learning Outcomes

After successful completion of the course, the student will be able to:

- design and implement programs on 8086 microprocessor
- design I/O circuits and Memory Interfacing circuits
- implement modular programming

Unit-1: Introduction & Architecture

Overview of 8086 microprocessor. Signals and pin diagram of 8086, 8086 architecture, internal operation, Register structure, ALU, Bus Organization, Physical memory organization, BIU, EU, Minimum mode and Maximum mode, Instruction execution time (Timing Diagram: Fetch Cycle, Execution Cycle).

Unit-2: Assembly Language Programming

Addressing modes, Instruction format, Instruction Types (data transfer instruction, arithmetic instructions, branch instruction, NOP & HLT instructions, flag manipulation instruction, logical instruction, shift and rotate instruction, String instructions), Assembler directions and operators.

Unit-3: Memory Interfacing and Modular Programming

Memory Interfacing (Static RAMs & ROMs), Stacks, Procedures, macros –local labels and nested macros.

Unit-4: I/O Interface and I/O Programming

I/O consideration, programmed I/O block transfer, Serial communication, asynchronous, synchronous, physical, 8251A; Parallel communication: 8255 A, 16-bit bus interface, Basic Interrupt processing, Interrupt and interrupt routines, 8259A Programmable Interrupt Controller, DMA with controller (8237), 8254 Programmable interval Timer (8253/8254), Interfacing to A/D and D/A converters, Stepper motor interfacing.

Suggested Readings:

1. Antonakos, J. L., *The Intel Microprocessor Family H/W and S/W Principles and Applications*, 1st Edition, Cengage Learning, 2007.
2. Bray, B.B., *The Intel Microprocessor 8086/8088-Pentium: Architecture, Programming and Interfacing*, 8th Edition, PHI, 2009.
3. Hall, D.V., *Microprocessors and Interfacing*, Revised 2nd Edition, Tata McGraw Hill, 2010.
4. Liu, Y.C., and Gibson, G.A., *Microcomputer Systems: 8086/8088 Family: Architecture, Programming and Design*, 2nd Edition, PHI, 2006.
5. Ray, A.K., and Bhurchandi, K.M., *Advanced Microprocessors and Peripherals: Architecture, Programming and Interfacing*, 3rd Edition, Tata McGraw Hill, 2006.

Analysis and Design of Algorithm Laboratory
Course Code: BT CS 505A

L T P **Credit: 01**
0 0 2

Course Level Learning Outcomes

After successful completion of the course, the student will be able to:

- apply and implement the algorithm for problem solving
- identify the data structure to develop program for real time applications
- design and develop optimal algorithms

List of Practicals:

1. Implement recursive binary search and linear search and determine the time required to search an element. Repeat the experiment for different values of n, the number of elements in the list to be searched and plot a graph of the time taken versus n.
2. Sort a given set of elements using heap sort method and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n.
3. Sort a given set of elements using merge sort method and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n.
4. Sort a given set of elements using a selection sort method and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n.
5. Implement 0/1 knapsack using dynamic programming.
6. From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm.
7. Sort a given set of elements using a quick sort method and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n.
8. Find the minimum cost spanning tree of a given undirected graph using Kruskal's algorithm.
9. a) Print all the nodes reachable from a given starting node in a digraph using BFS method.

   b) Check whether a given graph is connected or not using the DFS method.

10. Find a subset of a given set S={s1,s2,s3,.....,sn} of n positive integers whose sum is equal to a given positive integer d. For example, if S={1,2,5,6,8} and d=9 there are 2 solutions {1,2,6} and {1,8}. A suitable message is to be displayed if the given problem instance does not have a solution.
11. Find the minimum cost spanning tree of a given undirected graph using Prim's algorithm.
12. Implement N Queens problem using backtracking.

Note: This is only the suggested list of practical. Instructor may frame additional practicals relevant to the course contents.

Computer Network Laboratory
Course Code: BT CS 506A

L T P **Credit: 01**
0 0 2

Course Level Learning Outcomes

After successful completion of the course, the student will be able to:

- invoke analytical studies of Computer Networks through network simulation
- design a network using simulator and configure a real world network
- write a network program and implement network concept like RPC, Client- Server

List of Practicals:

1. To implement date and time display from localhost to server using TCP.
2. To write a client-server application for chat using TCP.
3. Programs to implement error correction and detection.
4. Connection oriented Client server applications with TCP.
5. Connectionless Client server applications with UDP.
6. Study of routing protocols using Cisco Packet Tracer.
7. Implement a chat and mail server.
8. Compare and contrast Network Topologies (Star, Mesh, Ring, Bus) using Cisco Packet Tracer.
9. Implement Configure enterprise network and show different subnet, private network using Cisco Packet Tracer.
10. To write a program to develop a DNS client server to resolve the given hostname.
11. Programs using RPC remote procedure call.

Note: This is only the suggested list of practical. Instructor may frame additional practicals relevant to the course contents.

Microprocessor and Interfacing Laboratory
Course Code: BT CS 507A

L T P **Credit: 01**
0 0 2

Course Level Learning Outcomes

After successful completion of the course, the student will be able to:

- write programs in assembly language using trainer kits/simulator
- interface development kits effectively for the real time applications of various peripheral devices with the processor
- handle arithmetic/logical operations using assembly language programming

List of Practicals:

1 a Write an ALP to move block of data without overlap.
  b Write an ALP to move block of data with overlap.
  c Write an ALP to interchange a block of data.
2 a Write an ALP to add 2 Multibyte numbers.
  b Write an ALP to subtract two Multibyte numbers.
  c Write an ALP to multiply two 16-bit numbers.
  d Write an ALP to perform the conversion from BCD to binary.
3 a Write an ALP to separate odd and even numbers.
  b Write an ALP to separate positive and negative numbers.
  c Write an ALP to check bitwise palindrome or not.
4 a Write an ALP to find largest number from a given array.
  b Write an ALP to find smallest number from a given array.
  c Write an ALP to sort a given set of 16bit unsigned integers into ascending order using bubble/insertion sort algorithm.
5 a Write an ALP to search a character in a string.
  b Write an ALP to check whether a given string is palindrome or not.
6 a Write an ALP to read a character from keyboard.
  b Write an ALP to read buffered input from the keyboard using dos interrupts.
  c Write an ALP to display single character.
7 a Scan 4*4 keyboard for key closure and display the corresponding key code.
  b Write an ALP for Seven segment LED display through 8255 (PCI based).
8 Program to rotate the Stepper motor in Clock-Wise direction(8steps).

Note: This is only the suggested list of practical. Instructor may frame additional practicals relevant to the course contents.

Summer Training
Course Code: BT CS 508A

L T P **Credits: 02**
0 2 0

Course Level Learning Outcomes

After successful completion of the course, the student will be able to:

- get exposure to industry practices
- apply the knowledge to the real world industrial problems
- get hands-on experience on latest industrial equipment/machinery

Methodology

- To identify industries offering internship by students
- To avail the training during summer vacation
- To submit a report based on the work done during internship to the department
- Non-Industry Internship students will take up the internship with a faculty member in the department

Programming with Python
Course Code: BT CS 521

L T P **Credits: 03**
3 0 0

Course Level Learning Outcomes

After successful completion of the course, the student will be able to:

- explain basic principles of Python programming language
- write programs using object oriented concepts
- implement GUI applications

Unit -1

The concept of data types; variables, assignments; immutable variables; numerical types; arithmetic operators and expressions; comments in the program; understanding error messages; Conditions, boolean logic, logical operators; ranges; Control statements: if-else, loops (for, while); short-circuit (lazy) evaluation; Strings and text files; manipulating files and directories, os and sys modules; text files: reading/writing text and numbers from/to a file; creating and reading a formatted file (csv or tab separated); String manipulations: subscript operator, indexing, slicing a string.

Unit -2

Lists, tuples, and dictionaries; basic list operators, replacing, inserting, removing an element; searching and sorting lists; dictionary literals, adding and removing keys, accessing and replacing values; traversing dictionaries; Design with functions: hiding redundancy, complexity; arguments and return values; formal vs actual arguments, named arguments.

Unit -3

Simple Graphics and Image Processing: "turtle" module; simple 2d drawing - colors, shapes; digital images, image file formats, image processing: Simple image manipulations with 'image' module (convert to bw, greyscale, blur, etc.). Classes and OOP: classes, objects, attributes and methods; defining classes; design with classes, data modeling; persistent storage of objects; inheritance, polymorphism, operator overloading (_eq_, _str_, etc.); abstract classes; exception handling, try block

Unit -4

Graphical user interfaces; event-driven programming paradigm; tkinter module, creating simple GUI; buttons, labels, entry fields, dialogs; widget attributes - sizes, fonts, colors layouts, nested frames.

Suggested Readings:

1. Rossum, G.V., and Drank, F.L., *An Introduction to Python*, 1st Edition, Network Theory Ltd, 2011.
2. Lambert, K., *Fundamentals of Python: First Programs*, 1st Edition, Cengage Learning, 2012.
3. Lutz, M., *Programming Python*, 4th Edition, O'Reilly, 2011.
4. Padmanabhan, T.R., *Programming with Python*, 1st Edition, Springer, 2016.
5. *The Python Tutorial Online Book*, Python Software Foundation (http://docs.python.org/3/tutorial/index.html)

Mobile Application Development
Course Code: BT CS 522

L T P **Credits: 03**
3 0 0

Course Level Learning Outcomes

After successful completion of the course, the student will be able to:

- comprehend Android platform and its usefulness in application development
- acquire skill set to execute applications in Android based devices
- design and develop deployable Android applications

Unit-1: Introduction to mobile devices

Introduction to Mobile Computing, Introduction to Android Development Environment, Mobile devices vs. desktop devices, ARM and intel architectures, Power Management, Screen resolution, Touch interfaces, Application deployment, App Store, Google Play, Windows Store, Development environments: XCode, Eclipse, VS2012, PhoneGAP, etc.; Native vs. web applications. Factors in Developing Mobile Applications: Mobile Software Engineering, Frameworks and Tools, Generic UI Development, Android User; Graphics and Multimedia: Performance and Multithreading, Graphics and UI Performance, Android Graphics, Mobile Agents and Peer-to-Peer Architecture, Android Multimedia.

Unit-2: Mobile OS Architectures

Comparing and contrasting architectures of all three – Android, iOS and Windows, Underlying OS, Kernel structure and native level programming. Approaches to power management, Security. Android/iOS/Win 8 Survival and basic apps: Building a simple "Hello World" App in all three applications, App-structure, built-in Controls, file access, basic graphics. Platforms and Additional Issues: Development Process, Architecture, Design, Technology Selection, Mobile App Development Hurdles, Testing.

Unit-3: Android/iOS/Win8 Apps

DB access, network access, contacts/photos/etc. Underneath the frameworks: Native level programming on Android, Low-level programming on (jailbroken) iOS, Windows low level APIs. Intents and Services: Android Intents and Services, Characteristics of Mobile Applications, Successful Mobile Development; Storing and Retrieving Data: Synchronization and Replication of Mobile Data, Getting the Model Right, Android Storing and Retrieving Data, working with a Content Provider; Putting It All Together: Packaging and Deploying, Performance Best Practices, Android Field Service App.

Unit-4: Advanced Topics

Power Management: Wake locks and assertions, Low-level OS support, Writing power-smart applications. Augmented Reality via GPS and other sensors: GPS, Accelerometer, Camera. Mobile device security, in depth: Mobile malware, Device protections, iOS "Jailbreaking", Android "rooting" and Windows' "defenestration"; Security and Hacking: Active Transactions, More on Security, Hacking Android.

Suggested Readings:

1. Hardy, B., and Phillips, B., *Android Programming: The Big Nerd Ranch Guide*, 4th Edition, Big Nerd Ranch LLC, 2019.
2. Fling, B., *Mobile Design and Development*, O'Reilly Media Inc, 2009.
3. Crumlish, C., and Malone, E., *Designing Social Interfaces*, 2nd Edition, O'Reilly Media, Inc.
4. Firtman, M., *Programming the Mobile Web*, 2nd Edition, O'Reilly Media Inc, 2013.
5. Ginsburg, S., *Designing The iPhone User Experience: A User-Centered Approach to Sketching and Prototyping iPhone Apps*, Addison-Wesley Professional, 2010.
6. Nurkiewicz, T. and Christensen, B., *Reactive Programming with RxJava*, O'Reilly Media, 2016.
7. Lee, V., Schneider, H., and Schell, R., *Mobile Applications: Architecture, Design and Development*, Prentice Hall, 2004.

Programming Concepts using Java
Course Code: BT CS 523

L T P **Credits: 03**
3 0 0

Course Level Learning Outcomes

After successful completion of the course, the student will be able to:

- identify classes, objects, members of a class and relationships among them needed for a specific problem
- write Java application programs using OOP principles and proper program structuring
- demonstrate the concepts of polymorphism, inheritance and error handling techniques using exception handling

Unit-1: Overview of class, object and methods

The overview of Java's architecture and the architecture of the Java Virtual Machine (JVM).

Classes: Declaring Members (Fields and Methods), Instance Members, Static Members.

Objects: Class Instantiation, Reference Values, and References. Primitive Data Types, Variable Declarations, Initial Values for Variables, Class Declarations, Method Declarations, this reference, Method Overloading, Constructors, The Default Constructor and Constructors overloading. Arrays, Anonymous Arrays, Multidimensional Arrays, The main() Method, Program Arguments.

Unit-2: Inheritance and Packages

Object-Oriented Programming: Single Implementation Inheritance, Overriding Methods, Hiding Members, The Object Reference super, Chaining Constructors Using this() and super().

Packages: Defining Packages, Using Packages, Compiling Code into Packages, Running Code from Packages. Scope Rules, Accessibility Modifiers, Overview of other Modifiers for Members. Operators and Expressions, Overview of Control Flow Statements.

Unit-3: Interfaces and Exception Handling

Interfaces: Defining Interfaces, Abstract Method Declarations, Implementing Interfaces, Extending Interfaces, Interface References, Constants in Interfaces, Polymorphism and Dynamic Method Lookup.

Exception Handling: The try Block, the catch Block, the finally Block, the throw Statement, the throws Clause, Checked and Unchecked Exceptions, Defining New Exceptions.

Unit-4: Fundamental Classes and Multithreading

Fundamental Classes: Overview of the java.lang Package, The Object Class, The Wrapper Classes, The String Class, The StringBuilder and the StringBuffer Classes.

Multithreading: Overview of Threads, the Main Thread, Thread Creation, Synchronization, Thread Transitions. Basics of Event Handling, Graphics Programming using AWT and Swing.

Suggested Readings:

1. Deitel, P., and Deitel, H., *Java How to Program*, 10th Edition, Pearson Education, 2015.
2. Eckel, B., *Thinking in Java*, 4th Edition, Pearson Education, 2006.
3. Horstmann, C.S., and Cornell, G., *Core Java Volume I (Fundamentals)*, 11th Edition, Pearson, 2019.
4. Naughton, P., and Schildt, H., *The Complete Reference Java*, 9th Edition, Tata McGraw Hill, 2017.
5. Sierra, K., and Bates, B., *Head First Java*, 2nd Edition, O'Reilly, 2005.

Programming with Python Laboratory

Course Code: BT CS 526

L	T	P	Credit: 01
0	0	2	

Course Level Learning Outcomes

After successful completion of the course, the student will be able to:

- write, test and debug python programs
- implement conditionals and loops for python programs using object oriented programming concepts
- implement string and function for python programs

List of Practicals:

1. Write python program to print Hello World
2. Write python program to Hello World using string variable
3. Write a python program to store data in list and then try to print them.
4. Write a python program to do basic trim and slice on string.
5. Write python program to print list of numbers using range and for loop
6. Write a python program to store strings in a list and then print them.
7. Write a python program to let the user enter some data in string and then verify data and print welcome to the user.
8. Write python program in which a function is defined and calling that function prints Hello World
9. Write a python program in which a function (with single string parameter) is defined and calling that function prints the string parameters given to the function.
10. Write a python program in which a class is defined, then create an object of that class and call a simple print function defined in the class.

Note: This is only the suggested list of practical. Instructor may frame additional practicals relevant to the course contents.

Mobile Application Development Laboratory
Course Code: BT CS 527

L T P **Credit: 01**
0 0 2

Course Level Learning Outcomes

After successful completion of the course, the student will be able to:

- to gain hands on experience in Android SDK
- design and develop applications in Android based devices
- design and develop deployable Android applications

List of Practicals:

1. a) Build your first simple Hello world application using android studio.
   b) Build Hello world application using Toast in android studio.
2. Build an application that contains a TextView with value "Hello world".
3. a) Add background to textView created in above question 3
   b) Using color.xml values
4. Add margins to TextView in question3
   a) Using hexadecimal value in xml layout
   b) Using dimens.xml values
5. Try Linear Layout that contains 4 buttons as UI Components with following attribute properties set – Layout, Button, Text.
6. Using Relative Layout Design a form that inputs first name , last name , Gender(using radioButton) and Date of Birth (Using Date Picker) with following attribute properties set – Layout, RadioButton, EditText.
7. a). Try Table Layout and design following using EditText
   b). Design Following using Grid Layout and use ImageView and TextView UI Controls for its designing
8. Create a Login page for Online Shopping using any learned Layouts
   a. Create TextView for the title
   b. Add two EditText for username and password
   c. Add Login button
   d. Create checkbox
   e. Create Radio button
9. Create an application to show the lifecycle of an activity. Include the following functions:

A. onCreate B. onStartC. onResumed. D. onPause E. OnStop F. OnRestartG. OnDestroy

10 a). Use Toast and logcat to show when each function gets calledactivity_main.xml

b). Create a calculator App with buttons for each operation using Event Delegation Model

11. Dice Roller : On Button click Application will display any random number between 1 to 6 OnClick Method
12. Create an application that lists the states of India using ListView.

Note: This is only the suggested list of practical. Instructor may frame additional practicals relevant to the course contents.

Programming Concept using Java Laboratory
Course Code: BT CS 528

L T P **Credit: 01**
0 0 2

Course Level Learning Outcomes

After successful completion of the course, the student will be able to:

- write, test and debug java programs
- implement conditionals and loops for java programs using object oriented programming concepts
- implement function and handle exception function for python programs

List of Practicals:

1. WAP to use various data types and type casting in Java.
2. WAP for if-else looping and break-continue statements in Java.
3. WAP for implementing For and While loop in Java.
4. WAP to implement constructors and inheritance in Java.
5. Implement Method overloading and overriding using Java.
6. WAP to implement garbage collection in Java.
7. Implement try-catch using Java program.
8. Implement exception handling using Java.
9. Use Java programming for accessing and performing operations on file.
10. Implement Fibonacci series and string reversal using Java.

Note: This is only the suggested list of practical. Instructor may frame additional practicals relevant to the course contents.

Principle of Operating Systems
Course Code: BT CS 601A

L T P **Credits: 03**
3 0 0

Course Level Learning Outcomes

After successful completion of the course, the student will be able to:

- comprehend the techniques used to implement the process manager
- comprehend virtual memory abstractions in operating systems
- design and develop file system interfaces

Unit-1

Overview of operating systems: Batch, iterative, time sharing, multiprocessor, distributed, cluster and real-time systems, Unix system introduction and commands, Operating system structure. Processes: states, synchronisation and scheduling. Deadlocks and deadlock handling. Memory management. Input-output management. Disk devices and file systems.

Unit-2

Inter-process Communication Race Conditions, Critical Section, Mutual Exclusion, Hardware Solution, Strict Alternation, Peterson's Solution, The Producer Consumer Problem, Semaphores, Event Counters, Monitors, Message Passing. Classical IPC Problems: Reader's & Writer Problem, Dining Philosopher Problem etc., Scheduling, Scheduling Algorithms.

Unit-3

File concept, Access methods, File types, File operation, Directory structure, File System structure, Allocation methods (contiguous, linked, indexed), Free-space management (bit vector, linked list, grouping), directory implementation (linear list, hash table), efficiency & performance.

Unit-4

Security & Protection Security Environment, Design Principles Of Security, User Authentication, and Protection Mechanism: Protection Domain, Access Control List.

Suggested Readings:

1. Dhamdhere, D.M., *Operating Systems: A Concept Based Approach*, 3rd Edition, Tata McGraw Hill, 2017.
2. Dietel, H. M., *An Introduction to Operating System*, 5th Edition, Pearson, 2009.
3. Silberschatz, A., and Galvin, P.B., *Operating System Concepts*, 10th Edition, Wiley, 2018.
4. Stalling, W., *Operating Systems: Internals and Design Principles*, 9th Edition, Pearson, 2018.
5. Tanenbaum, A.S., *Modern Operating System*, 4th Edition, Pearson, 2014.

Compiler Design
Course Code: BT CS 602A

L T P **Credits: 03**
3 0 0

Course Level Learning Outcomes

After successful completion of the course, the student will be able to:

- apply the knowledge of lexical tool to develop a scanner & parser
- design and develop software system for backend of the compiler
- comprehend and adapt to new tools and technologies in compiler design

Unit-1

Compiler structure: Analysis-Synthesis model of compilation, Various phases of a compiler, Tool based approach to Compiler Construction.

Lexical analysis: Interface with input, Parser and Symbol table, Token, Lexeme and Patterns. Difficulties in Lexical Analysis. Error reporting. Implementation. Regular definition, Transition diagrams, LEX.

Unit-2

Syntax analysis: CFGs, Ambiguity, Associativity, Precedence, Top Down Parsing, Recursive Descent Parsing, Transformation on the grammars, Predictive Parsing, Bottom Up Parsing, Operator Precedence grammars, LR parsers (SLR, LALR, LR), YACC.

Syntax directed definitions: Inherited and Synthesized Attributes, Dependency Graph, Bottom Up and Top Down Evaluation of Attributes, L- and S-Attributed Definitions.

Unit-3

Type checking: Type System, Type Expressions, Structural and Name Equivalence of types, Type Conversion, Overloaded Functions and Operators.

Run time system: Storage Organization, Activation Tree, Activation Record, Parameter Passing, Symbol Table, Dynamic Storage Allocation.

Unit-4

Intermediate code generation: Intermediate Representations, Translation of Declarations, Assignments, Control Flow, Boolean Expressions and Procedure Calls.

Code generation and instruction selection: Issues, Basic Blocks and Flow Graphs, Register Allocation, Code Generation, DAG representation of programs, Code Generation from DAGS, Peep-Hole Optimization, Code Generators, Specifications of machine.

Suggested Readings:

1. Aho, A.V., Sethi, R. and Ullman, J.D., *Compilers: Principles, Techniques and Tools*, 2nd Edition, Pearson, 2008.
2. Appel, A.W., *Modern Compiler Implementation in C*, 1st Edition, Cambridge University Press, 2004.
3. Dhamdhere, D.M., *Compiler Construction – Principles & Practice*, 2nd Edition, Macmillan India, 2008.
4. Fischer, C.N., and Leblanc, R.J., *Crafting a Compiler*, 1st Edition, Pearson, 2011.
5. Holub, A.I., *Compiler Design in C*, 1st Edition, PHI, 1992.

Software Engineering
Course Code: BT CS 603A

L T P **Credits: 03**
3 0 0

Course Level Learning Outcomes

After successful completion of the course, the student will be able to:

- enhance the software project management skills
- comprehend the systematic methodologies involved in SE
- design and develop a software product in accordance with SE principles

Unit-1

Problem domain, software engineering challenges, software engineering approach, Software process, characteristics of software process, software development process models, other processes. Software requirements, problem analysis, requirements specification, functional specification with use cases, validation, matrices.

Unit-2

Role of software architect, architecture views, component and connector view, architecture style for C & C view, discussion and evaluating architectures. Effort estimation, project scheduling and staffing, software configuration management plan, quality assurance plan, risk management, project monitoring plan.

Unit-3

Design principles, design notation and specification, structured design methodology, verification, metrics.OO concepts, design concept, design methodology, metrics. Detailed design and PDL, verification, Metrics and their scope, Qualities of a good Software metrics, classification of metrics, Cost estimation models COCOMO, Quality attributes, SQA, Quality Standards, ISO 9000 and CMM.

Unit-4

Programming principles and guidelines, coding process, refactoring, verification, metrics. Testing fundamentals, black-box testing, white-box testing, testing process, defect analysis and prevention, metrics - reliability estimation. Types of CASE tools, advantages and components of CASE tools, Unified Modeling Language (UML)

Suggested Readings:

1. Aggarwal, K.K., and Singh, Y., *Software Engineering*, 3rd Edition, New Age Publications, 2008.
2. Gill, N.S., *Software Engineering*, 1st Edition, Khanna Publications, 2003.
3. Jalote, P., *An Integrated Approach to Software Engineering*, 2nd Edition, Springer, 2013.
4. Mall, R., *Fundamentals of Software Engineering*, 4th Edition, PHI, 2014.
5. Pender, T., *UML Bible*, 1st Edition, Wiley, 2003.
6. Pressman, R. S., *Software Engineering: A Practitioner's Approach*, 9th Edition, Tata McGraw Hill, 2014.
7. Sommerville, I., *Software Engineering*, 10th Edition, Pearson, 2015.

Data Warehousing and Data Mining
Course Code: BT CS 604A

L T P **Credits: 03**
3 0 0

Course Level Learning Outcomes

After successful completion of the course, the student will be able to:

- comprehend the various architectures and its application with data mining
- design and develop data mining algorithms to analyze raw real world data
- monitor and analyze to predict online digital activities

Unit-1: Data Warehousing, Business Analysis and OLAP

Basic Concepts, Data Warehousing Components, Building a Data Warehouse, Database Architectures for Parallel Processing, Parallel DBMS Vendors, Multidimensional Data Model, Data Warehouse Schemas for Decision Support, Concept Hierarchies, Characteristics of OLAP Systems, Typical OLAP Operations, OLAP and OLTP.

Unit-2: Data Mining- Introduction

Introduction to Data Mining Systems, Knowledge Discovery Process, Data Mining Techniques, Issues, Applications, Data Objects and attribute types, Statistical description of data, Data Preprocessing, Cleaning, Integration, Reduction, Transformation and Discretization, Data Visualization, Data similarity and dissimilarity measures.

Unit-3: Data Mining – Frequent Pattern Analysis

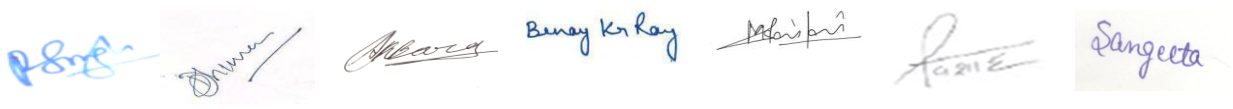
Mining Frequent Patterns, Associations and Correlations, Mining Methods, Pattern Evaluation Method, Pattern Mining in Multilevel, Multi-Dimensional Space, Constraint Based Frequent Pattern Mining, Classification using Frequent Patterns

Unit-4: Classification and Clustering

Decision Tree Induction, Bayesian Classification, Rule Based Classification, Classification by Back Propagation, Support Vector Machines, Lazy Learners, Model Evaluation and Selection, Techniques to improve Classification Accuracy, Clustering Techniques, Cluster analysis, Partitioning Methods, Hierarchical Methods, Density Based Methods.

Suggested Readings:

1. Berson, A., and Smith, S.J., *Data Warehousing, Data Mining & OLAP*, 1st Edition, Tata McGraw Hill, 2017.
2. Han, J., Pei, J., and Kamber, M., *Data Mining Concepts and Techniques*, 3rd Edition, Elsevier, 2011.
3. Margaret H.D., *Data Mining Introductory and Advanced Topics*, 3rd Impression, Pearson.
4. Pujari, A.K., *Data Mining Techniques Paperback*, 2nd Edition, Universities Press, 2010.
5. Soman, K.P., Diwakar, S., and Ajay, V., *Insight into Data Mining Theory and Practice*, 1st Edition, PHI, 2009.
6. Witten, I.H., and Frank, E., *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd Edition, Elsevier, 2016.



Principle of Operating System Laboratory
Course Code: BT CS 605A

L T P **Credit: 01**
0 0 2

Course Level Learning Outcomes

After successful completion of the course, the student will be able to :

- make use of tools for solving synchronization problems
- compare and contrast various CPU scheduling algorithms
- understand the differences between segmented and paged memories

List of Practicals:

1. Simulate the following CPU scheduling algorithms
    a. Round Robin
    b. SJF
    c. FCFS
    d. Priority
2. Simulate all file allocation strategies
    a. Sequential
    b. Indexed
    c. Linked
3. Simulate MVT and MFT memory management techniques.
4. Simulate all File Organization Techniques
    a. Single level directory
    b. Two level
    c. Hierarchical
    d. DAG
5. Simulate Bankers Algorithm for Dead Lock Avoidance
6. Simulate Bankers Algorithm for Dead Lock Prevention
7. Simulate all page replacement algorithms
    a. FIFO
    b. LRU
    c. LFU
8. Simulate Paging technique of memory management.

Note: This is only the suggested list of practicals. Instructor may frame additional practicals relevant to the course contents.

Software Engineering Laboratory
Course Code: BT CS 606A

L T P **Credit: 01**
0 0 2

Course Level Learning Outcomes

After successful completion of the course, the student will be able to :

- acquire the generic software development skill through various stages of SDLC
- implement and analyze the quality of software through software development with various protocol based environment
- develop, maintain and evaluate large-scale software systems

List of Practicals:

1. Writing problem statements for systems of relevance.
2. To perform requirement analysis and develop SRS documents.
3. To draw basic DFDs.
4. Implement Use case diagram.
5. Implementing a Class and Object diagram.
6. To draw state-charts and activity diagrams.
7. To draw Sequence and Collaboration diagrams.
8. Program to Unit testing.
9. Implement integration testing.
10. To draw component and deployment diagrams.

Note: This is only the suggested list of practical. Instructor may frame additional practicals relevant to the course contents.

Seminar
Course Code: BT CS 607A

L T P **Credit: 02**
0 2 0

Course Level Learning Outcomes

After successful completion of the course, the student will be able to :

- improve in oral communication skills through presentation
- improve in understanding research papers and prepare presentation material
- improve in analytical and reasoning ability and technical writing

Methodology

- To choose the area of interest
- To identify current literatures
- To choose state of the art survey paper/research paper
- To consult and get confirmed with departmental faculty
- To prepare the PPT
- To present as per schedule drawn by the department
- To prepare a technical report and submit to the department

Computer Graphics
Course Code: BT CS 621

L T P **Credits: 03**
3 0 0

Course Level Learning Outcomes
After successful completion of the course, the student will be able to:

- configure the various computer graphics hardware and display technologies.
- implement various 2D and 3D objects transformation techniques.
- apply 2D and 3D viewing technologies into the real world applications

Unit-1

Introduction to Computer Graphics: What is Computer Graphics, Computer Graphics Applications, Computer Graphics Hardware and software, Two dimensional.

Graphics Primitives: Points and Lines, Line drawing algorithms: DDA, Bresenham's; Circle drawing algorithms: Using polar coordinates, Bresenham's circle drawing, midpoint circle drawing algorithm; Filled area algorithms: Scanline: Polygon filling algorithm, boundary filled algorithm.

Unit-2

Two/Three Dimensional Viewing: The 2-D viewing pipeline, windows, viewports, window to viewport mapping; Clipping: point, clipping line (algorithms): 4 bit code algorithm, Sutherland-cohen algorithm, parametric line clipping algorithm (Cyrus Beck).

Polygon clipping algorithm: Sutherland-Hodgeman polygon clipping algorithm. Two-dimensional transformations: transformations, translation, scaling, rotation, reflection, and composite transformation.

Three-dimensional transformations: Three-dimensional graphics concept, Matrix representation of 3-D Transformations, Composition of 3-D transformation.

Unit-3

Viewing in 3D: Projections, types of projections, the mathematics of planar geometric projections, coordinate systems.

Hidden surface removal: Introduction to hidden surface removal. The Z- buffer algorithm, scan line algorithm, area sub-division algorithm.

Unit-4

Image Compression & Standards: Making still images; editing and capturing images; scanning images; computer color models; color palettes; vector drawing; 3D drawing and rendering; JPEG-objectives and architecture; JPEG-DCT encoding and quantization, JPEG statistical coding, JPEG predictive lossless coding; JPEG performance; overview of other image file formats as GIF, TIFF, BMP, PNG etc.

Suggested Readings:

1. Foley, J.D., Van, F.D., Van Dam, A., Feiner, S.K., Hughes, J.F., Angel, E., and Hughes, J., *Computer Graphics Principles and Practices*, 3rd Edition, Addison Wesley, 2013.

2. Gomes, J., Velho, L., and Sousa, M.C., *Computer Graphics Theory and Practice*, 1st Edition, CRC Press, 2012.
3. Hearn, D., and Baker, M. P., *Computer Graphics*, 2nd Edition, Pearson, 2002.
4. Rogers, D. F., *Procedural Elements for Computer Graphics*, 2nd Edition, Tata McGraw Hill, 2001.
5. Watt, A., *Fundamentals of 3Dimensional Computer Graphics*, Addision Wesley, 1999.

Unix and Linux Programming
Course Code: BT CS 622

L T P **Credits: 03**
3 0 0

Course Level Learning Outcomes

After successful completion of the course, the student will be able to :

- understand the basic set of commands and utilities in Linux/Unix systems.
- learn the important Linux/Unix library functions and system calls
- understand the inner workings of Unix-like operating systems.

Unit- 1: Linux Startup and Shell Programming

User accounts, accessing Linux- starting and shutting process, Logging in and Logging out, Command line, simple command, Unix file system: Linux/Unix files, inodes and structure and file system related commands, Shell as command processor, shell variables, creating command substitution, scripts, functions, conditionals, loops, customizing

Unit-2: Regular Expressions and Filters

Introducing regular expressions patterns, syntax, character classes, quantifiers, introduction to egrep, sed.

Unit-3: The C/C++ Environment

The C compiler, vim editor, compiler options, managing projects, memory management, use of make files, dependency calculations, memory management- dynamic and static memory, building and using static and dynamic libraries, using ldd, soname, dynamic loader, debugging with gdb.

Unit-4: Processes in Linux

Processes, starting and stopping processes, initialization processes, rc and init files, job control- at, batch, cron, time, network files, security, privileges, authentication, password administration, archiving, Signals and signal handlers, Linux I/O system.

Suggested Readings:

1. Das, S., *Your Unix- The Ultimate Guide*, 4th Edition, Tata McGraw Hill, 2017.
2. Forouzan, B.A., and Gilberg, R.F., *Unix and Shell Programming*, 1st Edition, Cengage Learning, 2009.
3. Goerzen, J., *Linux Programming Bible*, 1st Edition, IDG Books, New Delhi, 2000.
4. Matthew, N., *Beginning Linux Programming*, Richard Stones, 4th Edition, Wrox-Shroff, 2007.
5. Venkateshmurthy, M.G., *Introduction to Unix and Shell Programming*, 1st Edition, Pearson, 2005.
6. Welsh, M., and Kaufmann, L., *Running Linux*, 4th Edition, O'Reilly & Associates, 2002.

Computer Graphics Laboratory
Course Code: BT CS 626

L T P **Credit: 01**
0 0 2

Course Level Learning Outcomes

After successful completion of the course, the student will be able to:

- draw geometric primitives
- write a program to display 3D objects as a 2D display using perspective transformation
- implement basic transformations on objects using

List of Practicals:

1. Write a program for 2D line drawing as Raster Graphics Display.
2. Write a program for circle drawing as Raster Graphics Display.
3. Write a program for polygon filling as Raster Graphics Display
4. Write a program for line clipping.
5. Write a program for polygon clipping.
6. Write a program for displaying 3D objects as a 2D display using perspective transformation.
7. Write a program for rotation of a 3D object about an arbitrary axis.
8. Write a program for Hidden surface removal from a 3D object.

Note: This is only the suggested list of practical. Instructor may frame additional practicals relevant to the course contents.

Unix and Linux Programming Laboratory
Course Code: BT CS 627

L T P **Credit: 01**
0 0 2

Course Level Learning Outcomes

After successful completion of the course, the student will be able to :

- run various Unix commands on a standard Unix/Linux OS
- run C programs on Unix and able to handle Unix system calls
- write shell programs on Unix OS

List of Practicals:

1. Familiarize with Unix/Linux logging/logout and simple commands.
2. Familiarize with vim editor and Linux GUIs.
3. Using Bash shell develop simple shell programs.
4. Develop advanced shell programs using awk and grep.
5. Compile and debug various C programs using different options.
6. Learning of installation and upgradation of Linux operating system.
7. Install Linux on a PC having some other previously installed operating system. All OSs should be usable.
8. As supervisor create and maintain user accounts, learn package installation, taking backups, creation of scripts for file and user management, creation of startup and shutdown scripts using at, cron etc.

Note: This is only the suggested list of practical. Instructor may frame additional practicals relevant to the course contents.

Stochastic Processes and Queuing Theory
Course Code: BT CS 631

L T P **Credits: 03**
3 0 0

Course Level Learning Outcomes

After successful completion of the course, the student will be able to :

- apply the theory of discrete and continuous time Markov-processes to complex stochastic systems.
- analyze communication, networking and networked control problems with the tools of Markov-processes and queuing models
- derive and analyze main theoretic results for the modeling of queuing networks. Discuss the computational issues and solution approaches.

Unit-1

Random variables, Functions of random variables, jointly distributed random variables. Expectation, Expectation of functions of more than one random variable, Covariance and Correlation, Auto and Cross Correlation, Correlation Coefficient, Parameter Estimation – Maximum Likelihood Estimation and Maximum a Posteriori.

Unit-2

Stochastic Processes, Classification of stochastic processes, Stationary Random Processes- First order, second order and nth order, the Bernoulli process, The Poisson process, Renewal process, Advanced Renewal Theory, Renewal Function, Alternating Renewal Processes.

Unit-3

Markov Chains, Computation of n-step transition probabilities, Chapman-Kolmogorov theorem, Classification of states of a Markov Chain, Distribution of times between state changes, Irreducible finite chains with aperiodic states, Birth and Death processes, Pure Birth Process, Pure Death Process, Non-Birth-Death Processes, Analysis of program execution time.

Unit-4

Introduction to Queuing Theory, General Queuing, Components of a queuing system, Deterministic Queues, (M/M/1):(∞/FIFO)-Single server with infinite capacity, (M/M/k) : (∞/FIFO)- Multiple server with infinite capacity, M/G/1 queuing system, M/G/1 with non-FIFO disciplines, M/G/∞, Queues with time-varying arrival rate.

Suggested Readings:

1. Chitale, R. H., *Probability and Queueing Theory*, 1st Edition, Pearson, 2008.
2. Gross, D., and Shartle, J. F., *Fundamental of Queueing Theory*, 1st Edition, Tata McGraw Hill, 2018.
3. Palaniammal, S., *Probability and Queueing Theory*, 1st Edition, Prentice Hall of India, 2011.
4. Populis, A., and Pillai, S.U., *Probability, Random Variables, and Stochastic Process*, 4th Edition, TMH, 2002.
5. Sundarapandian, V., *Probability, Statistics and Queueing* Theory, 1st Edition, Prentice Hall of India, 2009.

Wireless Communication
Course Code: BT CS 632

L T P **Credits: 03**
3 0 0

Course Level Learning Outcomes

After successful completion of the course, the student will be able to:

- make critical assessment of wireless networks
- comprehend the fundamentals of Wireless Networks
- apply the knowledge gained in the development of MAC, Network Layer protocols of Wireless Network

Unit-1

Introduction to Wireless Communication System: Evolution of mobile radio communications, examples of wireless communication systems, paging systems, Cordless telephone systems, comparison of various wireless systems.

Modern Wireless Communication System: Second generation cellular networks, third generation wireless networks, wireless in local loop, wireless local area networks, Bluetooth and Personal Area Networks.

Unit-2

Introduction to Cellular Mobile Systems: Spectrum Allocation, Basic Cellular Systems, performance criteria, Operation of Cellular systems, Analog cellular systems, Digital cellular systems.

Cellular System Design Fundamentals: Frequency Reuse, channel assignment strategies, handoff strategies, Interference and system capacity, tracking and grade off service, improving coverage and capacity.

Unit-3

Multiple Access Techniques for Wireless Communication: Introduction to Multiple Access, FDMA, TDMA, spread Spectrum multiple Access, space division multiple access, packet ratio, capacity of cellular systems.

Wireless Networking: Difference between wireless and fixed telephone networks, development of wireless networks, fixed network transmission hierarchy, traffic routing in wireless network, wireless data services, common channel signaling, ISDN (Integrated Service Digital Networks), Advanced Intelligent Networks.

Unit-4

Intelligent cell concept and application: Intelligent cell concept, Applications of intelligent microcell systems, in building communication, CDMA cellular radio networks. Bluetooth, GSM

Suggested Readings:

1. Feher, K., *Wireless Digital Communications: Modulations and Spread Spectrum Applications*, 1st Edition, Prentice Hall, 1995.
2. Lee, W.C.Y., *Essentials of Wireless Communications*, 1st Edition, Tata McGraw Hill, 2000.
3. Mark, J.W. and Zhuang, W., *Wireless Communications and Networking*, 1st Edition, PHI, 2002.
4. Rappaport, T.S., *Wireless Communication: Principles & Practices*, 2nd Edition, Prentice Hall, 2010.
5. Tse, D., and Viswanath, P., *Fundamental of Wireless Communication*, 1st Edition, Cambridge University Press, 2005.

Distributed System
Course Code: BT CS 633

L T P **Credits: 03**
3 0 0

Course Level Learning Outcomes

After successful completion of the course, the student will be able to :

- analyze the principles of distributed systems and describe the problems and challenges associated with these principles.
- implement distributed computing techniques
- design a distributed system that fulfills requirements with regards to key distributed systems properties.

Unit-1

Characterization of Distributed Systems: Introduction, Hardware concepts, Software concepts, Design Issue and challenges, Examples of distributed Systems.

Communication in Distributed Systems: Layered protocols, Asynchronous transfer mode Networks, The Client – Server Model, Remote procedure call, Java RMI case study.

Unit-2

Synchronization in Distributed Systems:

Clock Synchronization, absence of global clock, Logical clocks, Lamport's & vectors logical clocks, Causal ordering of messages.

Distributed Mutual Exclusion: Classification of distributed mutual exclusion, Token based and non-token-based algorithms, performance metric for distributed mutual exclusion algorithms.

Unit-3

Distributed Deadlock Detection: system model, deadlock prevention, avoidance, detection & resolution, distributed deadlock detection algorithm.

Distributed Shared Memory: Introduction, Architecture of DSM Systems Design and implementation, granularly, structure of shared memory space Consistency models.

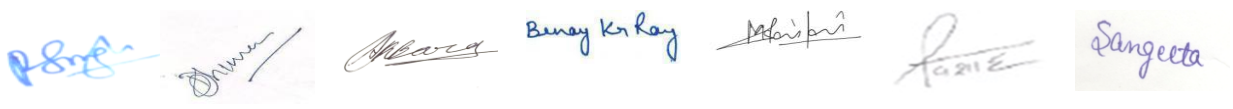
Unit-4

Distributed File Systems: Distributed File System Design, Distributed File Systems Implementation, Trends in Distributed File Systems.

Distributed Transaction: Flat and nested distributed transactions, Distributed Transaction Model, Atomic Commit protocols, Two Phase Commit Protocol.

Suggested Readings:

1. Coulouris, G., Dollimore, J., Kindberg, T., and Blair, G., *Distributed System: Concepts & Design*, 5th Edition, Pearson, 2009.
2. Ghosh, S., *Distributed Systems: An Algorithmic Approach*, CRC Press, 2014
3. Kshemakalyani, A.D., and Singhal, M., *Distributed Computing*, 2nd Edition, Cambridge University Press, 2011.
4. Sinha, P. K., *Distributed Operating Systems Concepts and Design*, 4th Edition, PHI, 2012.
5. Tanenbaum, A.S., and Steen, M. V., *Distributed Systems - Principles and Paradigms*, 2nd Edition, PHI, 2007.



Unified Modeling Language and Agile Methodology
Course Code: BT CS 634

L T P **Credits: 03**
3 0 0

Course Level Learning Outcomes

After successful completion of the course, the student will be able to:

- analyze existing problems with the team, development process and wider organization
- apply a thorough understanding of Agile principles and specific practices
- judge and craft appropriate adaptations to existing practices or processes depending upon analysis of typical problems

Unit-1: UML & Object-Oriented Modeling

An Overview of UML, architecture, UML Modeling Concepts – system models & views, data types, classes, objects, event & messages, object-oriented modeling; Modeling Relations- association, aggregation composition, generalization, dependency and structural diagrams, Requirement Elicitation- introduction, concepts & activities. Object oriented analysis design using diagrams such as use-case diagrams, class diagrams, sequence diagrams, collaboration diagrams, activity diagrams, object diagrams, interaction diagrams and state chart diagrams.

Unit-2: Fundamentals of Agile

The Genesis of Agile, Introduction and background, Agile Manifesto and Principles, Overview of Scrum, Extreme Programming, Feature Driven development, Lean Software Development, Agile project management, Design and development practices in Agile projects, Test Driven Development, Continuous Integration, Refactoring, Pair Programming, Simple Design, User Stories, Agile Testing, Agile Tools.

Unit-3: Agile Scrum Framework

Introduction to Scrum, Project phases, Agile Estimation, Planning game, Product backlog, Sprint backlog, Iteration planning, User story definition, Characteristics and content of user stories, Acceptance tests and Verifying stories, Project velocity, Burn down chart, Sprint planning and retrospective, Daily scrum, Scrum roles – Product Owner, Scrum Master, Scrum Team, Scrum case study, Tools for Agile project management.

Unit-4: Agile Software Design and Development

Agile design practices, Role of design Principles including Single Responsibility Principle, Open Closed Principle, Liskov Substitution Principle, Interface Segregation Principles, Dependency Inversion Principle in Agile Design, Need and significance of Refactoring, Refactoring Techniques, Continuous Integration, Automated build tools, Version control.

Suggested Readings:

1. Booch, G., Rumbaugh, J., and Jacobson, I., *Unified Modeling Language User Guide*, 2nd Edition, Addison Wesley, 2012.
2. Bruegge, B., and Dutoit, A. H., *Object-Oriented Software Engineering Using UML Patterns and Java*, 3rd Edition, Pearson, 2013.
3. Fowler, M., *UML Distilled: A Brief Guide To The Standard Object Modeling Language*, Addison-Wesley Professional, 2004.

4. Martin, R. C., *Agile Software Development- Principles, Patterns and Practices*, Subsequent edition, Prentice Hall, 2013.
5. Schawber, K., and Beedle, M., *Agile Software Development with Scrum*, 1st Edition, Pearson, 2001.

Scheme and Syllabus for B. Tech. (Computer Science and Engineering) (4th Year)

(Semester 7th and Semester 8th)

w.e.f. Session 2021-22

Department of
Computer Science and Engineering
School of Engineering & Technology

CENTRAL UNIVERSITY OF HARYANA
MAHENDERGARH-123031
HARYANA

Central University of Haryana
School of Engineering and Technology
Department of Computer Science and Engineering

B. Tech. 4thYEAR (SEMESTER–VII)

S. No.	Course Code	Course Title	Teaching Schedule			Credits
			L	T	P	
1	BT CS 701	Machine Learning	3	0	0	3
2	BT CS 702	Principles of Cloud Computing	3	1	0	4
3	BT CS 703	Machine Learning Lab	0	0	2	1
4	BT CS 704	Minor Project	0	0	8	4
5	BT CS 705	Summer Training****	0	2	0	2
6	**DCEC – 4 / PE - 4**					
	BT CS 721	Information Security	3	0	0	3
	BT CS 722	Software Testing	3	0	0	3
	BT CS 723	Big Data Analytics	3	0	0	3
	BT CS 724	High Performance Computing	3	0	0	3
	BT CS 725	Digital Image Processing	3	0	0	3
7	**DCEC – 5 / PE – 5 Lab**					
	BT CS 731	Competitive Programming	0	0	4	2
	BT CS 732	MATLAB/Scilab/Octave	0	0	4	2
8		OE/GEC	3	1	0	4
Total			12	3	14	23

Central University of Haryana
School of Engineering and Technology
Department of Computer Science and Engineering
B. Tech. 4th YEAR (SEMESTER–VIII)

Group 1

S. No.	Course Code	Course Title	Teaching Schedule			Credits
			L	T	P	
1	BT CS 801	Major Project	0	0	16	8
2	**DCEC – 6 / PE - 6**					
	BT CS 831	Software Project Management	3	0	0	3
	BT CS 832	Natural Language Processing	3	0	0	3
	BT CS 833	Professional Ethics and IPR	3	0	0	3
3	**DCEC – 7 / PE - 7**					
	BT CS 834	Internet of things	3	0	0	3
	BT CS 835	Cyber Security	3	0	0	3
	BT CS 836	Information Retrieval	3	0	0	3
4		OE/GEC	3	1	0	4
Total			**9**	**1**	**16**	**18**

Group 2

S. No.	Course Code	Course Title	Teaching Schedule			Credits
			L	T	P	
1	BT CS 811	Industrial Internship	0	0	36	18
Total			**0**	**0**	**36**	**18**

L = Lecture, T = Tutorial, P = Practical, & C = Credits

List of Open electives$
Offered by another department {This is only an indicative list (Not Exhaustive)}

S. No.	Course Title
1	Embedded System Design
2	Digital Image Processing
3	Digital Signal Processing
4	Mechatronics
5	Bioinformatics
6	VLSI System Design
7	Optimization Techniques

8	Mobile Communication
9	Satellite Communication
10	Information Theory and Coding
11	Video Activity Detection
12	Remote Sensing
13	Soft Computing
14	Entrepreneurship
15	Electric Vehicle
16	Technical Writing
17	Remote Sensing and GIS
18	Reinforcement Learning
19	Deep Learning
20	Computational Complexity
21	Real-Time Systems
22	Computer Vision
23	Introduction to Robotics
24	Pattern Recognition and Applications
25	Health Research Fundamentals
26	Soft Skills
27	Project Management
28	Operational Research

[$] Student can do the following courses through offline (in University Department) or through online (MOOC/ NPTEL/Swyam). If the student chooses the online mode, he/ she has to submit the grade score card.

Program Name: B. Tech.-Computer Science and Engineering

Course Code: BT CS 701	**Course Name:** Machine Learning	**L**	**T**	**P**	**C**
		3	**0**	**-**	**3**
Year and Semester	**4th year** **7th Semester**	**Contact hours per week:** (3 Hrs.) **Exam:** (3hrs.)			
Pre-requisite of course	Nil	**Evaluation**			
		CIE: 30	**TEE: 70**		

Course Objectives:

1. To introduce students to the basic concepts and techniques of Machine Learning.
2. To become familiar with regression methods, classification methods, clustering methods.
3. To enable students to apply machine learning concepts in real life problems.

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

CO701.1	Gain knowledge about basic concepts of machine learning
CO701.2	Identity machine learning techniques suitable for given problem
CO701.3	Solve the problems using various machine learning techniques
CO701.4	Apply dimensionality reduction techniques
CO701.5	Design application using machine learning techniques

Module No	**COURSE SYLLABUS CONTENTS OF MODULE**	**Hrs**	**COs**
1	**Introduction** Introduction and Basic Concepts: Definition of ML, Taxonomy of ML, types of learning, hypothesis space and inductive bias, evaluation, cross-validation, Instance based learning, Feature reduction, Collaborative filtering-based recommendation, Gradient Decent learning.	**14**	**CO701.1, CO701.2**
2	**Kernel Methods and Support Vector Machines** The Two-Class Problem, Dual Representation, Soft Margin Classification; Origins of Kernel methods, Kernel Mapping, The Kernel Trick; Constructing Kernels, Support Vector Machines: Formulation and Computation; Radial Basis Function Networks; Positive Semi-Definite Kernels, Linear Kernel, Polynomial Kernel.	**13**	**CO701.1, CO701.2**
3	**Clustering** Basic Clustering Techniques, Standard k-Means (Lloyd) Algorithm, Generalized Clustering, Overpartitioning, Merging, Modifications to the k-Means Algorithm, k-Means Wrappers, Rough k-Means, Fuzzy k-Means, k-Harmonic Means Algorithm, Hybrid Clustering Algorithms; Estimation using Incomplete Data, Two classes, Multiple classes, Least squares for classification, Fisher's linear discriminant, Relation to least squares, Fisher's discriminant for multiple classes	**10**	**CO701.2, CO701.3, CO701.4**
4	**Learning** Learning a Class from Examples, Linear, Non-linear, Multi-class and multi-label classification, Generalization error bounds: VC	**8**	**CO701.2 CO701.4 CO701.5**

	Dimension, Decision Trees: ID3, Classification and Regression Trees, Regression: Linear Regression, Multiple Linear Regression, Logistic Regression. Introduction to clustering, Hierarchical: AGNES, DIANA, Partitional: K-means clustering, KMode Clustering, Self-Organizing Map, Expectation Maximization, Gaussian Mixture Models		

Text Books:

1. Alpaydin, E., 2020. *Introduction to Machine Learning*. MIT press.
2. Deisenroth, M.P., Faisal, A.A. and Ong, C.S., 2020. *Mathematics for Machine Learning*. Cambridge University Press.
3. Kubat, M., 2017. *An Introduction to Machine Learning*. Springer, Cham.
4. Marsland, S., 2015. *Machine Learning: An Algorithmic Perspective*. CRC press.
5. Mohri, M., Rostamizadeh, A. and Talwalkar, A., 2018. *Foundations of Machine Learning*. MIT press.
6. Murphy, K.P., 2012. *Machine Learning: A Probabilistic Perspective*. MIT press.

Course Code: BT CS 702	Course Name: Principles of Cloud Computing	L	T	P	C
		3	1	-	4
Year and Semester	4th year 7th Semester	Contact hours per week: (4 Hrs.) Exam: (3hrs.)			
Pre-requisite of course	Nil	Evaluation			
		CIE: 30	TEE: 70		

Course Objectives:

1. To understand the basics of Cloud Computing.
2. To explore the cloud architecture and virtualization technology
3. To learn about cloud management technique and cloud security

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

CO702.1	Having a clear understanding of the subject related concepts and of contemporary issues.
CO702.2	Having an ability to design virtualized platforms and understand the services of different cloud service providers.
CO702.3	Analyze the requirement of security in cloud environment.

Module No	COURSE SYLLABUS CONTENTS OF MODULE	Hrs	COs
1	**Introduction** **Definition of Cloud Computing**: Defining a Cloud, Cloud Types – NIST model, Cloud Cube model, Deployment models, Service models, Cloud Reference model, Characteristics of Cloud Computing, Benefits and advantages of Cloud Computing; **Cloud Architecture**: A brief introduction on Infrastructure, Platforms, Virtual Appliances, Communication Protocols, Applications, Connecting to the Cloud by Clients; Services and Applications by Type: IaaS, PaaS, SaaS, IDaaS and CaaS.	**14**	**CO702.1**
2	**Virtualization**: Use of Concepts of Abstraction and Virtualization: Virtualization technologies and tools, Load Balancing and Virtualization, Hypervisors, Porting of applications in the Cloud; Concepts of Infrastructure as a Service; Use of IaaS Application frameworks; Use of Google, Amazon and Microsoft Web Services	**13**	**CO702.1, CO702.2**
3	**Cloud Management**: Features of network management systems, Monitoring of an entire cloud computing deployment stack – an overview with mention of some products, Lifecycle management of cloud services. **Cloud Security**: Cloud security concerns, Security boundary, Security service boundary, Security of data, Identity management.	**10**	**CO702.2, CO702.3,**
4	**Case Study on Open Source & Commercial Clouds**: Eucalyptus, Microsoft Azure, Amazon EC2, VM ware.	**8**	**CO702.3**

	Cloud Federation: Need of cloud federation, Advantages, Cloud federation Architecture, Pricing Model, Different Federation Formation Frameworks, Load balancing.		

Text Books:

1. Buyya, R., Vecchiola, C. and Selvi, S.T., 2013. *Mastering Cloud Computing: Foundations and Applications Programming*. Newnes.
2. Miller, M., 2008. *Cloud Computing*. Pearson.
3. Moyer, C.M., 2011. *Building Applications in the Cloud: Concepts, Patterns, and Projects*. Pearson Education India.
4. Sosinsky, B., 2010. *Cloud Computing Bible*. John Wiley & Sons.
5. Velte, A.T., Velte, T.J. and Elsenpeter, R., 2019. *Cloud Computing: A Practical Approach.* Tata McGraw-Hill.

Course Code: BT CS 703	**Course Name:** Machine Learning Laboratory	**L**	**T**	**P**	**C**
		0	**0**	**2**	**1**
Year and Semester	**4th year** **7th Semester**	**Contact hours per week:** (2 Hrs.) **Exam:** (3hrs.)			
Pre-requisite of course	Nil	**Evaluation**			
		CIE: 30	**TEE: 70**		

Course Objectives:

1. To introduce basic concepts and applications of machine learning.
2. To help students to learn the application of machine learning in the different fields of science.

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

CO703.1	Gain knowledge about basic concepts of machine learning
CO703.2	Identity machine learning techniques suitable for given problem
CO703.3	Solve the problems using various machine learning techniques
CO703.4	Apply dimensionality reduction techniques
CO703.5	Design application using machine learning techniques

Module No	**COURSE SYLLABUS List of Practicals**	**Hrs**	**COs**
1	Implement Decision Tree learning.	**2**	**CO703.1**
2	Implement Logistic Regression.	**2**	**CO703.1**
3	Implement classification using Multilayer perceptron.	**2**	**CO703.1**
4	Implement classification using SVM	**2**	**CO703.2**
5	Implement Adaboost	**2**	**CO703.2**
6	Implement Bagging using Random Forests	**2**	**CO703.2**
7	Implement K-means Clustering to Find Natural Patterns in Data	**2**	**CO703.3**
8	Implement Hierarchical clustering.	**2**	**CO703.3**
9	Implement K-mode clustering	**2**	**CO703.3**
10	Implement Principal Component Analysis for Dimensionality Reduction.	**2**	**CO703.4**
11	Implement Multiple Correspondence Analysis for Dimensionality Reduction.	**2**	**CO703.4**
12	Implement Gaussian Mixture Model Using the Expectation Maximization.	**2**	**CO703.4**
13	Evaluating ML algorithm with balanced and unbalanced datasets.	**2**	**CO703.5**
14	Comparison of Machine Learning algorithms.	**2**	**CO703.5**
15	Implement k-nearest neighbors' algorithm	**2**	**CO703.5**

Text Books:

1. Alpaydin, E., 2020. *Introduction to Machine Learning*. MIT press.
2. Deisenroth, M.P., Faisal, A.A. and Ong, C.S., 2020. *Mathematics for Machine Learning*. Cambridge University Press.
3. Kubat, M., 2017. *An Introduction to Machine Learning*. Springer, Cham.
4. Marsland, S., 2015. *Machine learning: An Algorithmic Perspective*. CRC press.

5. Mohri, M., Rostamizadeh, A. and Talwalkar, A., 2018. *Foundations of Machine Learning*. MIT press.
6. Murphy, K.P., 2012. *Machine Learning: A Probabilistic Perspective*. MIT press.

<table>
<tr><td>Course Code:
BT CS 704</td><td colspan="2">Course Name: Minor Project</td><td>L
0</td><td>T
0</td><td>P
8</td><td>C
4</td></tr>
<tr><td>Year and Semester</td><td>4th year
7th Semester</td><td colspan="4">Contact hours per week: (8 Hrs.)
Exam: (3hrs.)</td></tr>
<tr><td rowspan="2">Pre-requisite of course</td><td rowspan="2">Nil</td><td colspan="4">Evaluation</td></tr>
<tr><td colspan="2">CIE: 60</td><td colspan="2">TEE: 140</td></tr>
<tr><td colspan="7">Course Objectives:
1. To address the real-world problems and find the required solution.
2. To design the problem solution as per the requirement analysis done.
3. To study the basic concepts of programming/ hardware/ emulator for Raspberry pi/Arduino/ ARM Cortex/ Intel Galileo etc.
4. To implement the minor project intended solution for project-based learning.
5. To test the minor project successfully.
6. To improve the team building, communication and management skills of the students.</td></tr>
<tr><td colspan="7">Course Outcomes: On completion of the course, student would be able to:</td></tr>
<tr><td>CO704.1</td><td colspan="6">Demonstrate a through and systematic understanding of project contents.</td></tr>
<tr><td>CO704.2</td><td colspan="6">Understand methodologies and professional way of documentation and communication.</td></tr>
<tr><td>CO704.3</td><td colspan="6">Know the key stages in development of the project.</td></tr>
<tr><td>CO704.4</td><td colspan="6">Extend or use the idea in minor project for major project.</td></tr>
</table>

Minor Project

Students have to submit a project at the end of the semester. The student will submit a synopsis at the beginning of the semester for approval from the departmental committee in a specified format. The student will have to present the progress of the work through seminars or progress reports.

This course will be conducted largely as an individual or small group project under the direct supervision of a member of academic staff. The specific project topic undertaken will reflect the common interests and expertise of the student(s) and supervisor. Students will be required to:

1. Perform a literature search to review current knowledge and developments in the chosen technical area;
2. Undertake detailed technical work in the chosen area using one or more of:
    - theoretical studies
    - computer simulations
    - hardware construction;
3. Produce progress reports within the time frame specified for the project;
4. Deliver a seminar on the general area of work being undertaken and specific contributions to that field;
5. Prepare a formal report describing the work undertaken and results obtained so far; and
6. Present the work during external viva including demonstrations of operational hardware and software.

Course Code: BT CS 705	**Course Name:** Summer Training	**L**	**T**	**P**	**C**
		0	**2**	**0**	**2**
Year and Semester	**4th year** **7th Semester**	**Contact hours per week:** (2 Hrs.) **Exam:** (3hrs.)			
Pre-requisite of course	Nil	**Evaluation**			
		CIE: 15	**TEE: 35**		

Course Objectives:

1. To allow the students to relate the classroom learning outcomes to the actual engineering field experiences in either governmental or private sectors.
2. To build up the student's disciplinary, ability and personality to communicate effectively through teams with the most updated various industries and technologies.
3. To introduce the students for first time to their expecting careers.
4. To build up the relation between the college and the various industrial fields as well as knowing the needs and expectations of these fields for the graduated students.
5. To enable the private and public organizations to identify the skills of the trainee, and polarizing him/her for jobs.

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

CO705.1	An ability to apply knowledge of mathematics, science, and engineering
CO705.2	An ability to function as a team
CO705.3	An ability to identify, formulate, and solve engineering problems
CO705.4	An understanding of professional and ethical responsibility
CO705.5	An ability to communicate effectively
CO705.6	The broad education necessary to understand the impact of engineering solutions in a global, economic, environmental, and societal context
CO705.7	A knowledge of contemporary issues
CO705.8	An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.

Summer Training is a joint effort between the Engineering department, the public, and the private sectors in the area of specialization to allow students to practice the skills and knowledge. ST students are required to spend one and half months (6 working weeks) of practical training in a relevant field in industry at the end of the sixth semester.

Summer Training Report: Practical Training conducted after sixth semester will be evaluated in the seventh semester based on Viva-Voce.

Course Code: BT CS 721	Course Name: Information Security	L	T	P	C
		3	0	-	3
Year and Semester	**4th year** **7th Semester**	**Contact hours per week:** (3 Hrs.) **Exam:** (3hrs.)			
Pre-requisite of course	Nil	**Evaluation**			
		CIE: 30	**TEE: 70**		

Course Objectives:

1. To understand the concepts of classical encryption techniques and concepts of finite fields and number theory.
2. To explore the working principles and utilities of various cryptographic algorithms including secret key cryptography, hashes and message digests, and public key algorithms.
3. To explore the design issues and working principles of various authentication protocols, PKI standards.
4. To apply the concepts of cryptographic utilities and authentication mechanisms to design secure applications

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

CO721.1	learn the mathematical fundamentals from abstract algebra and number theory for the design and analysis of various cryptographic primitives.
CO721.2	understand the role of cryptographic techniques in modern communication systems.
CO721.3	design cryptographic systems to meet the specifications in terms of security, circuit complexity and power consumption by effectively making use of various primitives
CO721.4	Investigate latest developments in cryptography and cryptanalysis through most recent publications in small groups and prepare presentations on these topics

Module No	COURSE SYLLABUS CONTENTS OF MODULE	Hrs	COs
1	**Properties of arithmetic operations**- Inverse Primes, GCD, Euclidean Algorithm, Modular Arithmetic, Properties of Modular Arithmetic, Computing Inverse, Fermat's Theorem, Random Number Generation.	**14**	**CO721.1**
2	**Classical Cryptography** – Substitution and Transposition Cipher, Modern Cryptographic Techniques, Private Key Cryptosystems, Block cipher, Standards, Data Encryption Standard, AES, Linear and differential cryptanalysis Stream cipher, Key stream generators, RC4 cryptosystem	**13**	**CO721.2**
3	**Public key cryptosystems** – One-way functions, Factorization problem, RSA crypto system, Discrete logarithm problem, Elgamal crypto system, Key management, Diffie Hellmann key exchange, Digital Signatures, Elgamal DSA.	**10**	**CO721.3**
4	**Message authentication requirements** – Hash function, features of MD5 and SHA algorithms, Security of Hash function, Message Authentication Codes, Applications of authentication, electronic mail security, SSL/ TLS, Digital Certificate and X509.	**8**	**CO721.4**

Text Books:

1. Cole, E., 2011. *Network Security Bible*. John Wiley & Sons.
2. Dummit, D.S. and Foote, R.M., 2004. *Abstract Algebra* (3rd Edition). Hoboken: Wiley.
3. Kranakis, E., 2013. *Primality and cryptography*. Springer-Verlag.
4. Stallings, W., 2006. *Cryptography and Network Security*, (4th Edition). Pearson Education India.
5. Stinson, D.R. and Paterson, M., 2018. *Cryptography: Theory and Practice*. CRC press.

Course Code: BT CS 722	**Course Name:** Software Testing	**L**	**T**	**P**	**C**
		3	**0**	**-**	**3**
Year and Semester	**4th year** **7th Semester**	**Contact hours per week:** (3 Hrs.) **Exam:** (3hrs.)			
Pre-requisite of course	Nil	**Evaluation**			
		CIE: 30	**TEE: 70**		

Course Objectives:

1. To understand the basic software debugging methods.
2. To apply white box testing methods and techniques.
3. To apply black Box testing methods and techniques.
4. To analyze the test plans.
5. To apply the different testing tools (familiar with open-source tools)
6. To understand the quality Assurance models.

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

CO722.1	Identify various types of software risks and its impact on different software application
CO722.2	make selection of test cases and their optimization
CO722.3	get aware of various activities in testing and to generate test data using tools
CO722.4	apply the object-oriented concepts on test data

Module No	**COURSE SYLLABUS CONTENTS OF MODULE**	**Hrs**	**COs**
1	Software Testing, Importance of testing, Roles and Responsibilities, Testing Principles, Attributes of Good Test, V-Model SRS Verification, Source Code Reviews, Basic Introduction to - Functional Testing, Structural Testing Cyclomatic Complexity, Data Flow Testing, Mutation Testing.	**14**	**CO722.1**
2	Regression Testing: What is Regression Testing? Regression Test cases selection, Reducing the number of test cases, Code coverage prioritization technique. Reducing the number of test cases: Prioritization guidelines, Priority category, Risk Analysis.	**13**	**CO722.2**
3	Software Testing Activities: Levels of Testing, Debugging, Testing techniques and their applicability, Exploratory Testing Automated Test Data Generation: Test Data, Approaches to test data generation, Test Data Generation Tools, Software Testing Tools, and Software test Plan.	**10**	**CO722.3**
4	Object Oriented Testing: Definition, Issues, Class Testing, Object Oriented Integration and System Testing. Testing Web Applications: Web Testing, User Interface Testing, Usability Testing, Security Testing, Performance Testing, Database testing, Post Deployment Testing.	**8**	**CO722.4**

Text Books:

1. Ammann, P. and Offutt, J., 2016. *Introduction to Software Testing*. Cambridge University Press.
2. Beizer, B., 2003. *Software Testing Techniques*. Dreamtech Press.

3. Chauhan, N., 2010. *Software Testing: Principles and Practices*. Oxford university press.
4. Hetzel, B., 1988. *The Complete Guide to Software Testing*. QED Information Sciences, Inc.
5. Jorgensen, P.C., 2013. *Software Testing: A Craftsman's Approach*. Auerbach Publications.
6. Myers, G.J., Sandler, C. and Badgett, T., 2011. *The Art of Software Testing*. John Wiley & Sons.

Course Code: BT CS 723	Course Name: Big Data Analytics	L	T	P	C
		3	0	-	3
Year and Semester	**4th year 7th Semester**	**Contact hours per week:** (3 Hrs.) **Exam:** (3hrs.)			
Pre-requisite of course	Nil	**Evaluation**			
		CIE: 30	**TEE: 70**		

Course Objectives:

1. To provide an overview of an exciting growing field of big data analytics.
2. To introduce the tools required to manage and analyze big data like Hadoop.
3. To teach the fundamental techniques and principles in achieving big data analytics with scalability and streaming capability.

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

CO723.1	Having a clear understanding of the subject related concepts and of contemporary issues.
CO723.2	Having an ability to design and conduct experiments, as well as to analyze and interpret data
CO723.3	Having an ability to use techniques, skills and modern engineering tools necessary for engineering practice

Module No	COURSE SYLLABUS CONTENTS OF MODULE	Hrs	COs
1	**Introduction to big data and analytics:** Big Data Overview Characteristics of Big Data Business Intelligence vs Data Analytics. Data Analytics Life Cycle Data Analytics in Industries Exploring Big Data Challenges in handling Big Data.	**14**	**CO723.1**
2	**Big Data Tools:** Need of Big data tools - understanding distributed systems - Overview of Hadoop comparing SQL databases and Hadoop Hadoop Eco System - Distributed File System: HDFS, Design of HDFS writing files to HDFS Reading files from HDFS. Hadoop Daemons - Hadoop Cluster Architecture YARN Advantages of YARN.	**13**	**CO723.1, CO723.2**
3	**Introduction to MapReduce:** Developing MapReduce Program Anatomy of MapReduce Code -Simple Map Reduce Program - counting things Map Phase shuffle and sort - Reduce Phase Master slave architecture Job Processing in hadoop Map Reduce Pipelining. Use of Combiner - Block vs Split Size - working with Input and output format Key, Text, Sequence	**10**	**CO723.2**
4	**Frameworks:** Applications on Big Data Using Pig and Hive – Data processing operators in Pig – Hive services – HiveQL – Querying Data in Hive - fundamentals of HBase and ZooKeeper - IBM InfoSphere BigInsights and Streams.	**8**	**CO723.3**

Text Books:

1. Baesens, B., 2014. *Analytics in a Big Data World: The Essential Guide to Data Science and Its Applications*. John Wiley & Sons.
2. Bahga, A. and Madisetti, V., 2016. *Big Data Science & Analytics: A Hands-On Approach*. VPT Publisher.
3. Leskovec, J., Rajaraman, A. and Ullman, J.D., 2020. *Mining of Massive Data Sets*. Cambridge university press.
4. White, T., 2012. *Hadoop: The Definitive Guide*. (3rd Edition) " O'Reilly Media, Inc.".
5. White, T., 2015. *Hadoop: The Definitive Guide*. (4th Edition) " O'Reilly Media, Inc.".
6. Zikopoulos, P. and Eaton, C., 2011. *Understanding Big Data: Analytics for enterprise Class Hadoop and Streaming Data*. McGraw-Hill Osborne Media.

Course Code: BT CS 724	Course Name: High Performance Computing	L	T	P	C
		3	0	-	3
Year and Semester	4th year 7th Semester	Contact hours per week: (3 Hrs.) Exam: (3hrs.)			
Pre-requisite of course	Nil	Evaluation			
		CIE: 30	TEE: 70		

Course Objectives:

1. To provide systematic and treatment of the hardware and the software high performance techniques involved in current day computing.
2. To introduce the fundamentals of high-performance computing with the graphics processing units and many integrated cores using their architectures and corresponding programming environments
3. To introduce the learner to fundamental and advanced parallel algorithms through the GPU programming environments

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

CO724.1	To knowledge the overview and analyze the performance metrics of high-performance computing
CO724.2	To comprehend the various High Performance Computing Paradigms and Job Management Systems
CO724.3	To design and develop various applications with OpenMP, MPI and CUDA
CO724.4	To analyze the benchmarks of high-performance computing
CO724.5	To demonstrate the various emerging trends of high-performance computing
CO724.6	To apply high performance computing concepts in problem solving

Module No	COURSE SYLLABUS CONTENTS OF MODULE	Hrs	COs
1	**Introduction to High Performance Computing (HPC)**: Overview of Parallel Computers and high-performance computing (HPC), History of HPC, Numerical and HPC libraries, Performance metrics. **HPC Paradigms**: Supercomputing, Cluster Computing, Grid Computing, Cloud Computing, Many cores Computing, Peta scale Systems	**14**	**CO724.1, CO724.2**
2	**Parallel Programming**: Introduction to OpenMP, Parallel constructs, Runtime Library routines, Work-sharing constructs, Scheduling clauses, Data environment clauses, atomic, master No wait Clause, Barrier Construct, overview of MPI, MPI Constructs, OpenMP vs MPI.	**13**	**CO724.2, CO724.3**
3	**Batch scheduling**: Condor, Slurm, SGE, PBS, Light weight Task Scheduling: Falkon, Sparrow Introduction to GPU Computing, CUDA Programming Model, CUDA API, Simple Matrix, Multiplication in CUDA, CUDA Memory Model, Shared Memory Matrix Multiplication, Additional CUDA API Features	**10**	**CO724.4, CO724.5**

4	**Achieving Performance**: Measuring performance, identifying performance bottlenecks, Partitioning applications for heterogeneous resources, Using existing libraries and frameworks. **HPC Benchmarks**: HTC, MTC (Many Tasks Computing), Top Super computers in the world, Top Super Computer architectural details, Exploring HPC Benchmarks: HPL, Stream.	**8**	**CO724.5 CO724.6**

Text Books:

1. Czech, Z.J., 2017. *Introduction to Parallel Computing*. Cambridge University Press.
2. Eijkhout, V., van de Geijn, R. and Chow, E., 2016. *Introduction to High Performance Scientific Computing*. Zenodo.
3. Farber, R., 2011. *CUDA Application Design and Development*. Elsevier.

Course Code: BT CS 725	Course Name: Digital Image Processing	L	T	P	C
		3	0	-	3
Year and Semester	4th year 7th Semester	Contact hours per week: (4 Hrs.) Exam: (3hrs.)			
Pre-requisite of course	Nil	Evaluation			
		CIE: 30	TEE: 70		

Course Objectives:

1. To introduce the basic principles of digital images,
2. To understand the image data structures,
3. To study the image processing algorithms.

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

CO725.1	Understand the fundamental concepts of a digital image processing system
CO725.2	Analyze images in the frequency domain using various transform
CO725.3	Evaluate the techniques for image enhancement and image restoration
CO725.4	Interpret image compression standards
CO725.5	Categorize various compression techniques

Module No	COURSE SYLLABUS CONTENTS OF MODULE	Hrs	COs
1	**Introduction** Image Processing Fourier Transform and Z-Transform, Causality and stability, Toeplit and Circulate Matrics, orthogonal and unitary Matrices and Kroenker product, Markov Processes, KI Transform, Mean square Estimates and Orthogonal Principles Image sampling & Quantization: Band Limited Image, Sampling Versus Replication, Reconstruction of Image from samples Sampling Theorem, Sampling Theorem for Random Fields, Optimal Sampling, Non-rectangular Grid Sampling, Sampling Aperture, Display Aperture/ Interpolation Functions, Lagrange Interpolation, Moire Effect Image Quantization: Uniform Optimal Quantizer, Properties of Mean Square Quantizer, Compandor Design, Visual Quantization	**14**	**CO725.1, CO725.2**
2	**Image Transforms** Two Dimensional Orthogonal and Unitary Transforms and their properties, One Dimensional and Two Dimensional DFT, Cosine, Sine, Hadamard, SLANT, HAAR and KI Transforms and their properties, Approximation to KI Transforms. Image representation by stochastic models: One Dimensional Causal Models, AR, ARMA models, Spectral factorization, Non Causal Representation, Image Decomposition.	**13**	**CO725.2**
3	**Image Enhancement and Restoration** Point Operation, Histogram Modeling, Spatial Operations, Transform Operations, Multispectral Image Enhancement.	**10**	**CO725.3**

	Image Filtering: Image Observation Models, Inverse and Wiener filter, FIR Wiener Filters, Filtering using Image Transform, Causal Models and recursive filtering, Maximum entropy restoration, Extrapolation of band limited signal.		
4	**Image Analysis and Image Compression** Spatial feature extraction, Edge detection, Boundary extraction, Boundary representations, Region representations, Moment representations, Structures, Texture, Image Segmentation, Image Classification Image Reconstruction from Projections: Data Compression: Pixel Coding, Predictive Techniques, Transform Coding Theory, Transform coding of Image, Coding of two-tone image.	**8**	**CO725.4, CO725.5**

Text Books:

1. Dougherty, E. R., 2020. *Digital image processing methods*. CRC Press.
2. Gonzalez, R.C., and Woods, R. E., 2009. *Digital Image Processing*. Pearson 3rd Edition, Cham.
3. Jain, A.K., 1989. *Fundamentals of Digital Image Processing*. Prentice-Hall, Inc.
4. Pratt, W.K., 2007. *Digital Image Processing.* John Wiley & Sons Inc., New York.
5. Shih, F. Y., 2017. *Image processing and mathematical morphology: fundamentals and applications*. CRC press.

Course Code: BT CS 731	**Course Name:** Competitive Programming	**L**	**T**	**P**	**C**
		0	**0**	**4**	**2**
Year and Semester	**4th year** **7th Semester**	**Contact hours per week:** (4 Hrs.) **Exam:** (3hrs.)			
Pre-requisite of course	Nil	**Evaluation**			
		CIE: 30	**TEE: 70**		

Course Objectives:

1. Introduce the competitive programming.
2. To understand the concepts of Data Structures and Algorithms
3. To apply in the programming challenges in competitive platforms like codechef.com, uva.onlinejudge.com
4. To make students aware about the approaches applied at the world competitions.

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

CO731.1	describe how algorithmic problems are solved
CO731.2	recognize the time and memory complexity of an algorithm or a structure
CO731.3	analyse the given problem and recognize subproblems
CO731.4	assemble the solutions of subproblems to solve the whole problem
CO731.5	assess advantages and shortcomings of different algorithms

Module No	**COURSE SYLLABUS CONTENTS OF MODULE**	**Hrs**	**COs**
1	Introduction: What is Competitive Programming, Programming Techniques, Recursive Algorithms, Bit Manipulation, Efficiency (Time complexity) with examples. Sorting and Searching: Bubble Sort, Merge sort, Sorting Lower Bound, Counting Sort, Sweep Line algorithms, Scheduling Events, Tasks and Deadlines, Binary Search.	**10**	**CO731.1**
2	Data Structures: Dynamic Arrays, Set Structures, Set versus Sorting, Map Versus Array, Priority Queue versus Multiset. Dynamic Programming: Basic concepts, When Greedy Fails, Finding an Optimal Solution, Counting Solutions, Longest Increasing Subsequence, Paths in a Grid, Knapsack Problems, From permutations to subsets, Counting Tilings.	**12**	**CO731.2**
3	Graph Algorithms: Basic of Graphs, Graph Traversal, Shortest Paths, Directed Acyclic Graphs, Successor Graphs, Minimum Spanning Trees. Algorithm Design Topics: Bit-parallel Algorithms, Amortized Analysis, Finding Minimum Values Range Queries: Queries on Static Arrays, Tree Structures	**12**	**CO731.3, CO731.4**
4	Tree Algorithms: Basic Techniques, Tree Queries (finding ancestors, subtrees and paths, lowest common Ancestors, Merging Data Structures). Mathematics: Number Theory, Combinatorics, Matrices, Probability, Game Theory String Algorithms: Basic topics, String Hashing, Z-algorithm, Suffix Arrays	**14**	**CO731.5**

Text Books:

1. Halim, S., Halim, F., Skiena, S. S., and Revilla, M. A., 2013. *Competitive Programming 3*. Lulu Independent Publish.
2. Laaksonen, A., 2020. *Guide to Competitive Programming*. Springer International Publishing.

Course Code: BT CS 732	Course Name: MATLAB	L	T	P	C
		0	0	4	2
Year and Semester	**4th year** **7th Semester**	**Contact hours per week:** (4 Hrs.) **Exam:** (3hrs.)			
Pre-requisite of course	Nil	**Evaluation**			
		CIE: 30	**TEE: 70**		

Course Objectives:

1. To learn fundamental computer programming concepts such as variables, control structures, functions and many others.
2. To learn about various data types and how to handle them in MATLAB.
3. To learn the powerful support MATLAB provides for working with matrices.
4. To learn about file input/output.

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

CO732.1	Understand the main features of the MATLAB development environment
CO732.2	Use the MATLAB GUI effectively
CO732.3	Write simple programs in MATLAB to solve scientific and mathematical problems
CO732.4	Create and control simple plot and user-interface graphics objects in MATLAB

Module No	COURSE SYLLABUS CONTENTS OF MODULE	Hrs	COs
1	Introduction to MATLAB: Getting Started, Scripts, Making Variables, Manipulating Variables, Basic Plotting. Visualization and Programming: Functions, Flow Control, Line Plots, Image/Surface Plots, Efficient Codes, Debugging	**14**	**CO732.1**
2	Solving Equations, Curve Fitting, and Numerical Techniques: Linear Algebra, Polynomials, Optimization, Differentiation/Integration, Differential Equations	**13**	**CO732.2**
3	Advanced Methods: Probability and Statistics, Data Structures, Images, File I/O	**10**	**CO732.3,**
4	Various Functions and Toolboxes: Documentation, Miscellaneous Useful Functions, Graphical User Interfaces, Simulink, Symbolic Toolbox, Image Processing, Hardware Interface	**8**	**CO732.4**

Text Books:

1. Dyer, K. B. (2018). Introduction to MATLAB®. In *Introduction to Mechanism Design* (pp. 79-100). CRC Press.
2. Etter, D.M., Kuncicky, D.C. and Hull, D.W., 2002. *Introduction to MATLAB.* Prentice Hall.
3. Lyshevski, S.E., 2005. *Engineering and Scientific Computations using MATLAB*. John Wiley & Sons.
4. Moore, H. and Sanadhya, S., 2007. *MATLAB for Engineers*. Upper Saddle River, New Jersey: Prentice Hall.
5. Sigmon, K. and Davis, T.A., 2004. *MATLAB Primer*. Chapman and Hall/CRC.

<table>
<tr><td>Course Code:
BT CS 801</td><td colspan="2">Course Name: Major Project</td><td>L
0</td><td>T
0</td><td>P
16</td><td>C
8</td></tr>
<tr><td>Year and Semester</td><td>4th year
8th Semester</td><td colspan="4">Contact hours per week: (16 Hrs.)
Exam: (3hrs.)</td></tr>
<tr><td rowspan="2">Pre-requisite of course</td><td rowspan="2">Minor Project</td><td colspan="4">Evaluation</td></tr>
<tr><td colspan="2">CIE: 60</td><td colspan="2">TEE: 140</td></tr>
<tr><td colspan="7">Course Objectives:
1. To offer students a glimpse into real world problems and challenges that need IT based solutions.
2. To enable students to create very precise specifications of the IT solution to be designed.
3. To improve the team building, communication and management skills of the students.</td></tr>
<tr><td colspan="7">Course Outcomes: On completion of the course, student would be able to:</td></tr>
<tr><td>CO801.1</td><td colspan="6">Understand project characteristics and various stages of a project.</td></tr>
<tr><td>CO801.2</td><td colspan="6">To enable students to create very precise specifications of the IT solution to be designed.</td></tr>
<tr><td>CO801.3</td><td colspan="6">Analyze the learning and understand techniques for Project planning, scheduling and Execution Control</td></tr>
<tr><td>CO801.4</td><td colspan="6">Build and test the major project successfully</td></tr>
<tr><td>CO801.5</td><td colspan="6">develop effective communication skill by delivering a seminar based on major project.</td></tr>
</table>

Major Project

Students have to submit a project at the end of the semester. The student will submit a synopsis at the beginning of the semester for approval from the departmental committee in a specified format. The student will have to present the progress of the work through seminars or progress reports.

This course will be conducted largely as an individual or small group project under the direct supervision of a member of academic staff. The specific project topic undertaken will reflect the common interests and expertise of the student(s) and supervisor. Students will be required to:

1. Perform a literature search to review current knowledge and developments in the chosen technical area;
2. Undertake detailed technical work in the chosen area using one or more of:
    - Theoretical studies
    - Computer simulations
    - Hardware construction;
3. Produce progress reports within the time frame specified for the project;
4. Deliver a seminar on the general area of work being undertaken and specific contributions to that field;
5. Prepare a formal report describing the work undertaken and results obtained so far; and

Present the work during external viva including demonstrations of operational hardware and software.

Course Code: BT CS 831	Course Name: Software Project Management	L	T	P	C
		3	0	-	3
Year and Semester	4th year 8th Semester	Contact hours per week: (3 Hrs.) Exam: (3hrs.)			
Pre-requisite of course	Software Engineering	Evaluation			
		CIE: 30	TEE: 70		

Course Objectives:

1. To introduce the important concepts of project management related to managing software development.
2. To make students familiar with the different activities involved in Software Project Management.
3. To successfully plan and implement a software project management activity, and to complete a specific project in time with the available budget.

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

CO831.1	Identify the need for Software Project Management
CO831.2	Identify different techniques for software cost estimation
CO831.3	Understand activity planning and risk management
CO831.4	Manage and control projects

Module No	COURSE SYLLABUS CONTENTS OF MODULE	Hrs	COs
1	**Project Evaluation and Project Planning**: Importance of Software Project Management, Activities Methodologies, Categorization of Software Projects, Setting objectives, Management Principles, Management Control, Project portfolio Management, Cost-benefit evaluation technology, Risk evaluation, Strategic program Management, Stepwise Project Planning.	**14**	**CO831.1**
2	**Project Life Cycle and Effort Estimation:** Software process and Process Models, Choice of Process models, Mental delivery, Rapid Application development, Agile methods, Extreme Programming, SCRUM, managing interactive processes, Basics of Software estimation, Effort and Cost estimation techniques, COSMIC Full function points, COCOMO II- A Parametric Productivity Model, Staffing Pattern.	**13**	**CO831.2**
3	**Activity Planning and Risk Management:** Objectives of Activity planning, Project schedules, Activities, Sequencing and scheduling, Network Planning models, Forward Pass & Backward Pass techniques, Critical path (CRM) method, Risk identification, Assessment, Monitoring, PERT technique, Monte Carlo simulation, Resource Allocation, Creation of critical patterns, Cost schedules.	**10**	**CO831.3**
4	**Project Management and Control:** Framework for Management and control, Collection of data Project termination, visualizing progress, Cost monitoring, Earned Value Analysis, project	**8**	**CO831.4**

	tracking, change control, Software Configuration Management, Managing contracts, Contract Management.		

Text Books:

1. Futrell, R.T., Shafer, D.F. And Shafer, L., 2002. *Quality Software Project Management*. Prentice Hall Professional.
2. Meredith, J.R., Shafer, S.M. And Mantel Jr, S.J., 2017. *Project Management: A Strategic Managerial Approach*. John Wiley & Sons.
3. Royce, W., 1998. *Software Project Management*. Pearson Education India.
4. Stellman, A. And Greene, J., 2005. *Applied Software Project Management*. " O'Reilly Media, Inc.".
5. Wysocki, R.K., 2010. *Effective Software Project Management*. John Wiley & Sons.

Course Code: BT CS 832	Course Name: Natural Language Processing	L	T	P	C
		3	0	-	3

Year and Semester	4th year 8th Semester	Contact hours per week: (3 Hrs.) Exam: (3hrs.)	
Pre-requisite of course	Nil	Evaluation	
		CIE: 30	TEE: 70

Course Objectives:

1. To introduce the fundamental concepts and techniques of natural language processing.
2. To give students a clear understanding of linguistics methods, various tools and aspects of NLP.
3. To gain an in-depth understanding of the computational properties of natural languages and the commonly used algorithms for processing linguistic information.

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

CO832.1	Describe the challenges involved in developing NLP solutions
CO832.2	Describe various recent statistical methods in natural language processing
CO832.3	Develop the linguistics and their application to part-of-speech tagging
CO832.4	Develop background to various tools and aspects of NLP like syntax and semantic analysis, parsing, machine translation, information retrieval and statistical discourse processing

Module No	COURSE SYLLABUS CONTENTS OF MODULE	Hrs	COs
1	**Introduction:** NLP tasks in syntax, semantics, and pragmatics. Applications such as information extraction, question answering, and machine translation. The problem of ambiguity. The role of machine learning. Brief history of the field.	**14**	**CO832.1**
2	**N-gram Language Models:** The role of language models. Simple N-gram models. Estimating parameters and smoothing. Evaluating language models. **Part of Speech Tagging and Sequence Labeling:** Lexical syntax. Hidden Markov Models (Forward and Viterbi algorithms and EM training)	**13**	**CO832.2**
3	**Syntactic parsing:** Grammar formalisms and treebanks. Efficient parsing for context-free grammars (CFGs). Statistical parsing and probabilistic CFGs (PCFGs). Lexicalized PCFGs. Neural shift-reduce dependency parsing **Semantic Analysis:** Lexical semantics and word-sense disambiguation. Compositional semantics. Semantic Role Labeling and Semantic Parsing.	**10**	**CO832.3**
4	**Mechanics of Speech:** Speech Production Mechanism, Nature of Speech Signal, Discrete Time Modeling of Speech Production, Representation of Speech Signals, Classification of Speech Sounds, Phones, Phonemes, Phonetics, IPA and Phonetic Alphabets, Articulatory Features, Auditory Perceptions, Anatomical Pathways from Ear to the Perception of Sound Peripheral Auditory System.	**8**	**CO832.4**

Text Books:

1. Hapke, H., Howard, C. and Lane, H., 2019. *Natural Language Processing in Action*. Simon and Schuster.
2. Jurafsky, D., 2000. *Speech & Language processing*. Pearson Education India.
3. Manning, C. and Schutze, H., 1999. *Foundations of Statistical Natural Language Processing*. MIT press.
4. Rabiner, L.R. and Juang, B.H., 1999. *Fundamentals of Speech Recognition*. Tsinghua University Press.
5. Thanaki, J., 2017. *Python Natural Language Processing*. Packt Publishing Ltd.

Course Code: BT CS 833	**Course Name:** Professional Ethics and IPR	**L**	**T**	**P**	**C**
		3	**0**	**-**	**3**
Year and Semester	**4th year** **8th Semester**	**Contact hours per week:** (3 Hrs.) **Exam:** (3hrs.)			
Pre-requisite of course	Nil	**Evaluation**			
		CIE: 30	**TEE: 70**		

Course Objectives:

1. To provide basic knowledge about engineering Ethics, Variety of moral issues and Moral dilemmas, Professional Ideals and Virtues
2. To provide basic familiarity about Engineers as responsible Experimenters, Research Ethics, Codes of Ethics, Industrial Standards, Exposure to Safety and Risk, Risk Benefit Analysis
3. To have an idea about the Collegiality and Loyalty, Collective Bargaining, Confidentiality, Occupational Crime, Professional, Employee, Intellectual Property Rights.
4. To have an adequate knowledge about MNC's, Business, Environmental, Computer Ethics, Honesty, Moral Leadership, sample Code of Conduct.

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

CO833.1	Identify and analyze an ethical issue in the subject matter under investigation or in a relevant field
CO833.2	Identify the multiple ethical interests at stake in a real-world situation or practice
CO833.3	Articulate what makes a particular course of action ethically defensible
CO833.4	Assess their own ethical values and the social context of problems
CO833.5	Identify ethical concerns in research and intellectual contexts, including academic integrity, use and citation of sources, the objective presentation of data, and the treatment of human

Module No	**COURSE SYLLABUS CONTENTS OF MODULE**	**Hrs**	**COs**
1	**Values:** Concept, Types, Rokeach Value Survey **Different Kinds of Values:** Individual, Societal, Material, Psychological, Cultural, Moral and Ethical, Spiritual; The Burgeoning Crises at Each of these levels. **Modern Approach to the Study of the Values:** Analyzing Individual Human Values such as Creativity, Freedom, Wisdom and Love; Value Spectrum for a Good Life; The Indian Concept of Values, Comparison of Eastern and Western concept of values.	**10**	**CO833.1, CO833.2**
2	**Ethics:** Values, Morals and Ethics; Need for Ethics in Professional Life; Kohlberg's Theory of Moral Development and its Applicability to Engineers. **Professional Ethics:** Values in Work Life; Professional Ethics and Ethos; Codes of Conduct, Whistle-Blowing, Corporate Social Responsibility, Case Studies on Ethics in Business.	**12**	**CO833.3, CO833.4**
3	**Introduction to IPR:** Nature and Enforcement, International Character of IPRs, Role of IPRs in Economic Development.	**12**	**CO833.4**

	Patents: Introduction to Patents, Object of Patent Law, Inventions not Patentable, Obtaining Patents, Rights and Obligations of a Patentee. **Copyrights:** Introduction to Copyrights, Subject-Matters of Copyright, Rights Conferred by Copyright, Infringement, Assignment and Licensing of Copyrights, Copyright Societies, International Copyright, Performer's Rights.		
4	**Trademarks:** Functions, Significance and types of Trademarks, Distinctiveness and Deceptive Similarity, Registration Procedure, Trademark Registry, Grounds for Refusal of Registration of Trademarks, Concurrent Use, Character Merchandising. **Trade Secrets:** Meaning, Types of Trade Secrets, Statutory Position of Trade Secrets in India, Proofs Required in Trade Secret Litigation Case.	**14**	**CO833.5**

Text Books:

1. Durkheim, E., 2013. *Professional Ethics and Civic Morals*. Routledge.
2. Fleddermann, C.B., 1999. *Engineering Ethics*. Upper Saddle River, NJ: Prentice Hall.
3. Friedman, B. and Kahn Jr, P.H., 2007. Human Values, Ethics, and Design. In *The human-computer interaction handbook* (pp. 1267-1292). CRC press.
4. Govindarajan, M., Natarajan, S. and Senthilkumar, V.S., 2009. *Engineering Ethics Includes Human Values*. Prentice-Hall of India Private Limited.
5. Harris Jr, C.E., Pritchard, M.S., Rabins, M.J., James, R. and Englehardt, E., 2013. *Engineering Ethics: Concepts and Cases*. Cengage Learning.
6. Naagarazan, R.S., 2007. *A Textbook on Professional Ethics and Human Values*. New Age International.
7. Piccolo, F. L. (2016). *Ethics and planning research*. Routledge.
8. Whitbeck, C., 2011. *Ethics in Engineering Practice and Research*. Cambridge University Press.

Course Code: BT CS 834	**Course Name:** Internet of Things	**L**	**T**	**P**	**C**
		3	**0**	**-**	**3**
Year and Semester	**4th year** **8th Semester**	**Contact hours per week:** (3 Hrs.) **Exam:** (3hrs.)			
Pre-requisite of course	Nil	**Evaluation**			
		CIE: 30	**TEE: 70**		

Course Objectives:

1. To understand the definition and significance of the Internet of Things
2. To gives a foundation in the Internet of Things, including the components, tools, and analysis
3. To understand the factors that contributed to the emergence of IoT
4. To examine the potential business opportunities that IoT can uncover

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

CO834.1	Able to understand the application areas of IOT
CO834.2	Able to realize the revolution of Internet in Mobile Devices, Cloud & Sensor Networks
CO834.3	Able to understand building blocks of Internet of Things and characteristics

Module No	**COURSE SYLLABUS CONTENTS OF MODULE**	**Hrs**	**COs**
1	**Introduction To Internet of Things:** Definition & Characteristics of IoT - Challenges and Issues - Physical Design of IoT, Logical Design of IoT - IoT Functional Blocks, Security. **Components In Internet of Things:** Control Units Communication modules Bluetooth Zigbee Wifi GPS- IOT Protocols, MQTT, Wired Communication, Power Sources.	**14**	**CO834.1**
2	**Technologies Behind IoT:** Four pillars of IOT paradigm, - RFID, Wireless Sensor Networks, SCADA (Supervisory Control and Data Acquisition), M2M - IOT Enabling Technologies - BigData Analytics, Cloud Computing, Embedded Systems.	**13**	**CO834.1, CO834.2**
3	**Programming The Microcontroller For IoT:** Working principles of sensors IOT deployment for Raspberry Pi /Arduino /Equivalent platform Reading from Sensors, Communication: Connecting microcontroller with mobile devices, communication through Bluetooth, wifi and USB - Contiki OS. **Resource Management in IoT:** Clustering, Clustering for Scalability, Clustering Protocols for IOT.	**10**	**CO834.2, CO834.3**
4	**IoT Applications and case studies:** Business models for the internet of things, Smart city, smart mobility and transport, smart buildings and infrastructure, smart health, environment monitoring and surveillance.	**8**	**CO834.3**

Text Books:

1. Bahga, A. and Madisetti, V., 2014. *Internet of Things: A hands-on approach*. Vpt.
2. Doukas, C., 2012. *Building Internet of Things with the ARDUINO*. CreateSpace Independent Publishing Platform.
3. Greengard, S. (2021). *The internet of things*. MIT press.
4. Hassan, Q. F., & Madani, S. A. (Eds.). (2017). *Internet of things: Challenges, advances, and applications*. CRC Press.
5. Uckelmann, D., Harrison, M. and Michahelles, F. eds., 2011. *Architecting the Internet of Things*. Springer Science & Business Media.
6. Vermesan, O. and Friess, P., 2014. *Internet of Things-from Research and Innovation to Market Deployment*. Aalborg: River publishers.

Course Code: BT CS 835	**Course Name:** Cyber Security	**L**	**T**	**P**	**C**
		3	**0**	**-**	**3**
Year and Semester	**4th year** **8th Semester**	**Contact hours per week:** (3 Hrs.) **Exam:** (3hrs.)			
Pre-requisite of course	Nil	**Evaluation**			
		CIE: 30	**TEE: 70**		

Course Objectives:

1. Exhibit knowledge to secure corrupted systems, protect personal data, and secure computer networks in an organization.
2. Practice with an expertise in academics to design and implement security solutions.
3. Understand key terms and concepts in Cryptography, Governance and Compliance.
4. Develop cyber security strategies and policies
5. Understand principles of web security and to guarantee a secure network by monitoring and analyzing the nature of attacks through cyber/computer forensics software/tools.

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

CO835.1	To make a solid foundation in digital security to protect device from threats.
CO835.2	To learn access control mechanism and understand how to protect servers.
CO835.3	To understand the importance of a network basics and brief introduction on security of network protocols.
CO835.4	To understand cyber-attacks and learn data privacy issues and preventive measures.

Module No	**COURSE SYLLABUS CONTENTS OF MODULE**	**Hrs**	**COs**
1	Basics of digital security, protecting personal computers and devices, protecting devices from Virus and Malware, Identity, Authentication and Authorization, need for strong credentials, keeping credentials secure, Protecting servers using physical and logical security.	**14**	**CO835.1**
2	Networking basics (home network and large-scale business networks), Networking protocols, Security of protocols, SSL/ TLS, Network security basics, basics of error control mechanism, Secure Electronic Transactions, Web Security Considerations.	**13**	**CO835.2**
3	Database and program security, OS security, Security policies, Models of Security, IP security Architecture, Security of Databases, Reliability and Integrity, Protection of information, Program Security, Malicious Code, Virus Signatures, Trapdoors.	**10**	**CO835.3**
4	Introduction to cyber-attacks, application security (design, development and testing), operations security, monitoring, identifying threats and remediating them, Principles of data security - Confidentiality, Integrity and Availability, Data Privacy, Data breaches, preventing attacks and breaches with security controls, Compliance standards, Computer Ethics.	**8**	**CO835.4**

Text Books:

1. Brooks, C.J., Grow, C., Craig, P. and Short, D., 2018. *Cybersecurity Essentials*. John Wiley & Sons.
2. Graham, J., Olson, R. and Howard, R. eds., 2016. *Cyber Security Essentials*. CRC Press.
3. Gupta, B.B., Agrawal, D.P., and Wang, H., 2018. *Computer and Cyber Security: Principles, Algorithm, Applications, and Perspectives*. CRC Press.
4. Krutz, R.L. and Vines, R.D., 2010. *Cloud Security: A Comprehensive Guide to Secure Cloud Computing*. Wiley Publishing.
5. Pfleeger, C.P., 1988. *Security in Computing*. Prentice-Hall, Inc.
6. Sammons, J. and Cross, M., 2016. *The Basics of Cyber Safety: Computer and Mobile Device Safety Made Easy*. Elsevier.
7. Wu, C.H.J. and Irwin, J.D., 2016. *Introduction to Computer Networks and Cybersecurity*. CRC Press.

Course Code: BT CS 836	Course Name: Information Retrieval	L	T	P	C
		3	0	-	3
Year and Semester	**4th year** **8th Semester**	**Contact hours per week:** (3 Hrs.) **Exam:** (3hrs.)			
Pre-requisite of course	Nil	**Evaluation**			
		CIE: 30	**TEE: 70**		

Course Objectives:

1. To become familiar with difference between Information retrieval and data Base Management Systems
2. To learn different indexing techniques to apply Data Base systems
3. To understand various searching techniques to retrieve data from databases and ware houses.

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

CO836.1	understand basic Information Retrieval Systems
CO836.2	identify the different types of indices: inverted index, positional index, bi-word index etc
CO836.3	enumerate various types of indices. And also understand the concept of efficient storage of indices.
CO836.4	understand the basic concept of Search Engines their architecture and various functional components
CO836.5	understand the basic concept of Web crawlers and their architecture

Module No	COURSE SYLLABUS CONTENTS OF MODULE	Hrs	COs
1	**Introduction to Information Retrieval**: Information retrieval problem, an inverted index, Processing Boolean queries, The extended Boolean model versus ranked retrieval, an inverted index, Bi-word indexes, Positional indexes, Combination schemes. **Index construction**: Hardware basics, Blocked sort-based indexing, Single-pass in-memory indexing, Distributed indexing, Dynamic indexing, other types of indexes	**14**	**CO836.1**
2	**Index compression**: Statistical properties of terms in information retrieval, Heaps' law: Estimating the number of terms, Zipf's law: Modeling the distribution of terms, Dictionary compression, Dictionary as a string, Blocked storage, Posting's file compression. **Scoring, term weighting and the vector space model**: Parametric and zone indexes, Weighted zone scoring, Learning weights, The optimal weight, Term frequency and weighting, Inverse document frequency, Tf-idf weighting, The vector space model for scoring, Variant tf-idf functions	**13**	**CO836.2**
3	**Computing scores in a complete search system:** Efficient scoring and ranking, Inexact top K document retrieval, Index elimination, Champion lists, Static quality scores and ordering, Impact ordering, Cluster pruning, Components of an information retrieval system, Tiered indexes.	**10**	**CO836.3, CO836.4**

	Web search basics: Background and history, Web characteristics, The web graph, Spam, Advertising as the economic model, The search user experience, User query needs Crawling, Crawler architecture, DNS resolution, The URL frontier, Link analysis, The Web as a graph, Anchor text and the web graph, PageRank, Markov chains, The PageRank computation, Topic-specific PageRank		
4	**Language models for information retrieval**: Language models, Finite automata and language models, Types of language models, Multinomial distributions over words, the query likelihood model, using query likelihood language models in IR, Estimating the query generation probability, Language modelling versus other approaches in IR	**8**	**CO836.4 CO836.5**

Text Books:

1. Baeza-Yates, R. and Ribeiro-Neto, B., 1999. *Modern Information Retrieval*. New York: ACM press.
2. Croft, W.B., Metzler, D. and Strohman, T., 2010. *Search Engines: Information Retrieval in Practice* (Vol. 520, pp. 131-141). Reading: Addison-Wesley.
3. Kubat, M., 2017. *An Introduction to Machine Learning*. Springer, Cham.
4. Schütze, H., Manning, C.D. and Raghavan, P., 2008. *Introduction to Information Retrieval* (Vol. 39, pp. 234-265). Cambridge: Cambridge University Press.

<table>
<tr><td>Course Code:
BT CS 811</td><td colspan="2">Course Name: Industrial Internship</td><td>L
0</td><td>T
0</td><td>P
36</td><td>C
18</td></tr>
<tr><td>Year and Semester</td><td>4th year
8th Semester</td><td colspan="4">Contact hours per week: (36 Hrs.)
Exam: (3hrs.)</td></tr>
<tr><td rowspan="2">Pre-requisite of course</td><td rowspan="2">Summer Training</td><td colspan="4">Evaluation</td></tr>
<tr><td colspan="2">CIE: 150</td><td colspan="2">TEE: 350</td></tr>
<tr><td colspan="7">Course Objectives:
1. A practice-oriented and ‘hands-on’ working experience in the real world or industry, and to enhance the student’s learning experience.
2. An opportunity to develop a right work attitude, self-confidence, interpersonal skills and ability to work as a team in a real organizational setting.
3. An opportunity to further develop and enhance operational, customer service and other life-long knowledge and skills in a real-world work environment.
4. Pre-employment training opportunities and an opportunity for the company or organization to assess the performance of the student and to offer the student an employment opportunity after his/her graduation, if it deems fit.</td></tr>
<tr><td colspan="7">Course Outcomes: On completion of the course, student would be able to:</td></tr>
<tr><td>CO811.1</td><td colspan="6">Demonstrate the application of knowledge and skill sets acquired from the course and workplace in the assigned job function/s;</td></tr>
<tr><td>CO811.2</td><td colspan="6">Solve real life challenges in the workplace by analyzing work environment and conditions, and selecting appropriate skill sets acquired from the course;</td></tr>
<tr><td>CO811.3</td><td colspan="6">Articulate career options by considering opportunities in company, sector, industry, professional and educational advancement;</td></tr>
<tr><td>CO811.4</td><td colspan="6">Communicate and collaborate effectively and appropriately with different professionals in the work environment through written and oral means;</td></tr>
<tr><td>CO811.5</td><td colspan="6">Exhibit professional ethics by displaying positive disposition during internship.</td></tr>
</table>