Curriculum-2020

Master of Technology

in

Computer Science and Engineering

Indian Institute of Information Technology Vadodara

Table of Contents

Master of Technology Program

The Institute offers a 2-years Master of Technology (M. Tech.) program in Computer Science and Engineering (CSE). The first year is devoted to course work while in the second year, a student has to work for his/her M.Tech. thesis. The student is expected to work either on research problems or on industry oriented problems.

Academic Year (1st July - 30th June)

Each academic year is divided into two semesters of approximately eighteen weeks duration with at least seventy working days for classes in each semester. The two semesters are:

1. Autumn Semester (July-November)
2. Winter Semester (January-April)

In addition, there are two inter-semester breaks:

1. Winter (December)
2. Summer (May-June)

The Senate approves schedules of academic activities for an academic year, inclusive of dates for registration, mid-semester and end-semester examinations, inter-semester breaks etc. The schedule is laid down in the Academic Calendar for the year.

Registration

At the beginning of each semester, until the completion of the program, a student must register for the semester and for the courses that he/she will study during the semester.

Eligibility for Course Registration

A student with no backlog courses (*i.e.* who has passed all the previous courses) will be eligible to register for all courses prescribed in the curriculum for semester. A student who has a backlog course(s) or is on academic probation may be recommended a different set of courses by the Dean of Academic Programs (Dean-AP).

Prerequisite Courses

A student registering for a course must have successfully completed the prerequisite course(s), if any, for that particular course. For hard prerequisites, a minimum grade of 'DD' is required.

Auditing of Courses

Auditing of courses allows students to gain exposure to additional subjects without increasing their overall workload. Registration of courses for Audit is permitted from third semester onwards under the following conditions:

1. A student can audit a maximum of two courses during the entire program.
2. A student has to enter the courses to be audited in the Course Registration Form while registering for the semester. The word 'Audit' would be specially mentioned in the remarks column of the student's course registration form.

3. A student can register a course for audit provided the following two conditions are satisfied: (*i*) the course instructor permits and approves the registration, and (*ii*) the lecture, lab and tutorial time-table strictly permits.
4. An audit course will not be considered as an overload.
5. If the student's performance is satisfactory, a grade of 'P' (Pass) would be awarded. If the performance is not satisfactory, 'F' (Fail) would be awarded.
6. An audit course will not be considered for the calculation of Semester Performance Index (SPI) / Cumulative Performance Index (CPI). However, the course will be reflected in the Semester Grade Report and Transcript as an Audit Course provided a grade of 'P' was obtained, otherwise the course will not appear in the Semester Grade Report and Transcript.

Course Load in Regular Semesters

A student is permitted to register for additional courses over the prescribed courses in the curriculum for a regular semester provided the total number of courses does not exceed 7 and the total credits do not exceed 26. A student is permitted to under-load his/her prescribed academic load in a regular semester by dropping one or more courses provided the number of courses is at least 4 and the registered credits are not less than 12.

Course Assessment

The assessment of students' academic performance includes mid-semester and end-semester examinations along with other continuous evaluation components. The various components of continuous assessment in a course may include home assignments, tutorial assignments, group assignments, quizzes, tests (open or closed book), viva-voce, mini projects, etc. Attendance in lectures/ tutorials/ labs may also be given due weightage in course assessment. The instructor may make attendance in lectures/ tutorials/ labs compulsory (80% or less). The instructor may, in due consultation with the Dean-AP, award 'F' grade to students who do not achieve the prescribed level of attendance in that course.

The distribution of weightage, for the assessment of academic performance of students in a course, through various modes listed above will be communicated by the course instructor at the beginning of the semester with due approval from the Director.

Letter Grade (10-point Scale)

For every course registered by a student, he/she is awarded a letter grade based on his/her combined performance in all the assessments. These letter grades are assigned points on a 10-point scale as described in the table below:

Letter Grade	Points	Remark
AA	10	Outstanding
AB	9	Excellent
BB	8	Very Good
BC	7	Good
CC	6	Average
CD	5	Below Average
DD	4	Poor
F	0	Fail
P	-	Pass
I	-	Incomplete

A student passes the course if he/she gets any grade in the range of ‘AA’ to ‘DD’, but fails if he/she gets the grade ‘F’. Certain courses are indicated as Pass/Fail courses, and in these courses a grade of ‘P’ or ‘F’ is awarded. ‘F’ grade may also be awarded in case of malpractice in examination/continuous evaluation process. Pass/Fail courses are not considered for calculation of SPI/CPI.

‘I’ grade will be awarded in a course if the overall performance of the student is satisfactory in the course, but the student either misses the end-semester examination due to illness, accident/death in the family or obtains such an approval from the Dean-AP under exceptional circumstances. A student who misses the end-semester examination must apply for permission with reasoning and proof. An application not so supported will not be considered. Grade ‘I’ awarded for missing the end-semester examination will be converted into a performance grade (depending on the overall performance of the student in the course) after taking an examination equivalent to the end-semester examination of that particular course. An ‘I’ grade must be converted into a performance grade by the specified date in the academic calendar for the next semester, otherwise it will be converted into an ‘F’ grade.

Performance Indices

[A] Semester Performance Index (*SPI*)

The performance of a student in a semester is indicated by the *SPI*. The *SPI* is the weighted average of the grade points obtained in all the courses registered by the student during the semester, calculated to two decimal places.

[B] Cumulative Performance Index (*CPI*)

An up-to-date assessment of the overall performance of a student from the time of entering the Institute is obtained by calculating the student's *CPI*. The *CPI* is a weighted average of the grade points obtained in all the courses registered for credit by the student after entering the Institute. The *CPI* is also calculated to two decimal places.

Calculation of *SPI* and *CPI*

Let the course credits are, U1, U2,....and the numeric values of the corresponding grade awarded in the courses are ,G1, G2….., respectively, the SPI is given by

$$SPI = \frac{U1G1+U2G2+........}{U1+U2+........}$$

In the above computation, the courses with 'P' grade are not considered. Similarly, the *CPI* indicates the cumulative academic performance in all the courses taken including those taken in the current semester as

$$CPI = \frac{1}{Total\ Credits} \sum_{i=1}^{4} \left(SPI \times Total\ credits\ of\ the\ i^{th}\ Semester\right)$$

Graduating *CPI* and *Class*

For the purposes of computing the *CPI* at the end of the M. Tech. program, the students' *CPI* will be computed on the basis of the best *CPI* obtained from the courses taken. The grade of M. Tech. Thesis/Project (if graded as prescribed in the curriculum) will be included while computing the final *CPI* of the student. The minimum *CPI* for eligible to graduate the M. Tech. program is 6.00. The Transcript of a graduating student will indicate

1. *Distinction* when *CPI* ≥ 9.00,
2. *First Class* when 6.50 ≤ *CPI* <9.00 and
3. *Pass* when 6.00 ≤ *CPI* < 6.50

Repeating a Course

[A] As a Backlog Course

A student must repeat a course taken for credit in which he/she has obtained an 'F' grade. Such a course is regarded as a backlog course. A backlog elective course can be replaced by another elective of the same category. A student can appear for examination (Mid-Semester and End-Semester) components only as a backlog course. The component of continuous evaluation will be carried forward from earlier evaluation. In case a student has obtained an ‘F’ grade in a Lab Course, he/she has to repeat the course, i.e., attend all lab sessions and take lab exams.

[B] For Grade Improvement

A student whose *CPI* is less than 5.00 can be permitted to reappear in the courses in which a ‘DD’ grade was obtained. This is for the purpose of grade improvement in a regular semester only. The grade obtained in the repeated attempt(s) will be considered for the purpose of calculating the *CPI* for the semesters thereafter. However, the grade obtained in the first and subsequent attempt(s) will be shown in the Transcript.

Award of Degree

The M. Tech. (CSE) will be conferred on a student after he/she has fulfilled the graduation requirements stipulated in the curriculum (as approved by the senate).

Award of Medals

The students with highest scholastic performance will be awarded with the Chairperson’s Gold Medal and the Institute Medals as per the Institute policy.

Minimum and Maximum Period for Completion of M. Tech. Program

The minimum period to complete the program is two academic years. In any case, a student should fulfil the requirements for her/his degree within a maximum period of three academic years, failing which she/he will be required to leave the Institute. The period of three years excludes any semester in which the student has availed “withdrawn” status.

M. Tech. Program: Course Categories and Distribution of Credits

Definition of Credits

Teaching of the courses shall be reckoned in credits; credits are assigned to the courses based on the following general pattern:

1 hour of Lecture (L) per week	1 Credit
1 hour of Tutorial (T) per week	1 Credit
2 hours of Laboratory (P) per week	1 Credit
3 hours of Laboratory (P) per week	2 Credits
6 hours of Thesis (P) per week	4 Credits

Course Categories and Range of Credits for M. Tech. Program

M. Tech. program offered by the institute is designed with the following credit guidelines presented in table below

Course Categories	Min. Credits	Max. Credits
Program Core	12	16
Program Elective	14	20
Thesis	22	24
Total Credits	**54-60**	

Courses in terms of Lecture-Tutorial-Practicals: Credits (L-T-P: C)

Program Core	3-0-0: 3, 0-1-2: 2, 2-0-0: 2, 1-0-2: 2, 3-0-2: 4, 3-1-0: 4
Program Elective	3-0-0: 3, 1-0-0: 1, 1-0-2: 2, 0-1-2: 2, 1-0-2: 2
Comprehensive Viva	0-0-0: 2
Thesis	0-0-3: 2, 0-0-6: 4

Note: *Courses with (1-x-x) or (0-x-x) will not be evaluated through mid- and end-semester examinations.*

Semester wise credit distribution:

Computer Science and Engineering

Semester	Course Category	Credits			
		L	**T**	**P**	**C**
I	Mathematics	2	0	0	2
	Program Core	x	x	x	9
	Program Elective	x	x	x	4
	Lab Course	0	0	2	1
	Total	**x**	**x**	**x**	**16**
II	Program Core	3	0	0	3
	Program Elective	x	x	x	11
	Humanities and Social Sciences	x	x	x	2
	Total	**x**	**x**	**x**	**16**
III	Comprehensive Viva	0	0	0	2
	Humanities and Social Sciences	x	x	x	2
	Thesis	0	0	15	10
	Total	**0**	**0**	**15**	**14**
IV	Thesis	0	0	18	12
	Total	**0**	**0**	**18**	**12**
Grand Total		**x**	**x**	**x**	**58***

*****_The credits given are for guidelines_**.** _The total credits may vary in the range 58-60._

Graduation Requirements

A student to be eligible to receive M. Tech. degree should meet the following criteria:

1. completed all registered courses with minimum DD/Pass Grade.
2. should have secured CPI $\geq$ 6.00 (on 10.00 point scale).
3. should have acquired minimum 34 credits in 'Graded Courses' out of which there must be at least
   - 14 credits from Program Core,
   - 14 credits from Program Electives and
   - 4 credits from Humanities and Social Sciences Electives.
4. should have secured minimum 22 credits from 'Thesis' and 2 credits of Comprehensive viva.

Computer Science and Engineering

Courses: Semester-I

Course Code	Course Name	Credits			
		L	T	P	C
MA601	Essential Mathematics	2	0	0	2
CS603	Advanced Algorithms	3	0	0	3
CS609	Probability, Statistics, Random Process	3	0	0	3
CS611	Computer Systems	3	0	0	3
CS661	Algorithms and DBMS Lab	0	0	2	1
CSXXX	Program Elective*	x	x	x	4
Total		**x**	**x**	**x**	**16**

List of Program Electives along with Course Code and Credits is provided in Appendix-I

Courses and Contents

MA601	Essential Mathematics	2-0-0: 2

Objective: Students entering a PG program usually find that their mathematical foundation is inadequate to pursue research for their thesis. It is also a fact that, for them to achieve the required level of mathematical maturity entirely through self-study is difficult. This course is designed with an objective to provide the essential knowledge required to remove this inadequacy. The content of the course is designed keeping in mind the mixed audience coming from computer science and allied engineering disciplines. At the conclusion of this course, students should have a sound understanding of what mathematics is about, and should have acquired a level of mathematical literacy that would enable them to see its relevance in their own domain of knowledge.

Learning Outcomes: Upon successful completion of this course, student will be able to formulate the engineering problems in the language of mathematics

Prerequisite: Nil

Contents:

Module 1: Sets, Relations and Functions: Order, Equivalence and Correspondence; Groups, Rings and Fields: Permutations, Symmetries, Polynomials.

Module 2: Vector Space: Basis, Linear transformat-ions, Norm and Inner-Product, Orthogonality, Metric: continuity, convergence and completeness, Finite Dimensional Vector Space: System of linear equations, Eigen-values, Eigen vectors, Matrix inverse, Least squares and Pseudo inverse, Change of basis and similarity transform.

Module 3: Introduction to Graphs and Connections with Linear Algebra: Random Graphs, Adjacency and Incidence matrices, Graph spectrum, Graph Partitioning and Clustering, Max-Min flow and Graph cuts, Shortest path algorithms.

Text Books:

1. *Introduction to Linear Algebra*, Gilbert Strang, 5th Ed, SIAM, 2016.
2. Kolman, Busby, Ross, *Discrete Mathematical Structures*, Sixth Edition, Pearson, 2008

Reference Books:

1. Lehman, Leighton and Meyer, *Mathematics for Computer Science*, 2017. https://courses.csail.mit.edu/6.042/spring17/mcs.pdf
2. *Linear Algebra*, Kunze Ray, Hoffman Kenneth 2nd Ed, Phi Learning, 2014.
3. *Fundamentals of Matrix Computations,* David S. Watkins, 3rd ed, Wiley.
4. Kepner and Jananthan, *Mathematics of Big Data*, 2018

CS603	**Advanced Algorithms**	**3-0-0: 3**

Objective: Advanced Algorithms is an advanced course on designing and analysis of algorithms and is an important part of computer science. This course introduces students to advanced techniques for the design and analysis of algorithms, and explores a variety of applications.

Learning Outcomes: By the end of this course, students should develop the following skills:

1. Understand and apply the algorithms discussed, prove their correctness, and analyze their time complexity in a mathematically rigorous manner.
2. Understand the fundamentals behind the techniques, so that you are able to develop algorithms for new problems where these techniques can be applied.
3. Given a practical application, identify the computational issues and apply suitable algorithms to solve it effectively.

Prerequisite: Fundamental course in algorithms or complexity.

Contents:

Review of Algorithm analysis, order arithmetic: Growth functions, Recurrences and solution of recurrence equations, time and space complexities, average and worst case analysis, lower bounds. Algorithm design techniques: divide and conquer, Greedy Algorithms, Dynamic Algorithm.

Review of Advanced Data Structures: AVL Trees, Red-Black Trees, Augmenting Data Structures, Optimal Binary Search Trees, Skip Lists, Self-Adjusting Binary Search Trees (Splay trees/amortized analysis), Binomial Heaps, Fibonacci Heaps, Disjoint Sets (union-find trees), Hashing, Perfect Hashing.

Introduction to Amortized analysis, Aggregate, Accounting, and Potential methods, String Matching Algorithms: Naive Algorithm; Robin-Karp Algorithm, String matching with Finite Automata, Knuth-Morris-Pratt and Boyer-Moore Algorithms.

Graph Algorithms: Review of Graph Traversal algorithms (BFS and DFS), Bellman-Ford Algorithm Shortest paths in a DAG, Johnson's Algorithm for sparse graphs, Flow networks and the Ford-Fulkerson Algorithm, Maximum bipartite matching.

Hard problems- Problem classes P, NP, NP-hard and NP-complete, Reducibility, Introduction to Approximation Algorithms and Randomized Algorithms

Text Book:

1. T.H.Cormen, C.E.Leiserson, R.L.Rivest, "*Introduction to Algorithms*", Prentice Hall India, 2002.

Reference Books:

1. D.E.Knuth, "*The Art of Computer Programming*", Vols. 1 and 3, Addison Wesley, 1998.
2. A.V.Aho, J.E.Hopcroft, J.D.Ullman, "*Design and Analysis of Algorithms*", Addison Wesley, 1976.
3. E.Horowitz, S.Sahni, "*Fundamentals of Computer Algorithms*", Galgotia Publishers, 1984.
4. K.Melhorn, "*Data Structures and Algorithms*", Vols.1 and 2, Springer Verlag, 1984.
5. P.W.Purdom, Jr. and C.A.Brown, "*The Analysis of Algorithms*", Holt Rhinehart and Winston, 1985.

CS609	Probability, Statistics and Random Process	3-0-0: 3

Objective: The main objective of this course is to provide students with the foundations of probabilistic and statistical analysis mostly used in varied applications in engineering and science mainly in signal/images processing, machine learning, artificial intelligence, and data networks.

Learning Outcomes: After studying this course, students will be able to understand and appreciate the key fundamentals of probability and statistical analysis which would be helpful to them in building strong foundations for advanced courses.

Prerequisite: Working knowledge of Linear Algebra and Calculus.

Contents:

INTRODUCTION: Classical, relative frequency and axiomatic definitions of probability, σ-field, measurable space, probability space, addition rule and conditional probability, multiplication rule, total probability, Bayes' theorem and independence.

RANDOM VARIABLES: Discrete, continuous and mixed random variables, probability mass, probability density and cumulative distribution functions, mathematical expectation, moments, probability and moment generating function, median and quantiles, Markov inequality, Chebyshev's inequality.

SPECIAL DISTRIBUTIONS: Discrete uniform, binomial, geometric, negative binomial, hypergeometric, Poisson, continuous uniform, exponential, gamma, beta, normal, lognormal, inverse Gaussian, Cauchy, double exponential distributions, reliability and hazard rate, reliability of series and parallel systems.

JOINT DISTRIBUTIONS: Joint, marginal and conditional distributions, product moments, correlation and regression, independence of random variables, bivariate normal distribution, representation of multivariate normal distribution.

TRANSFORMATIONS: Functions of random vectors, distributions of order statistics, distributions of sums of random variables.

SAMPLING DISTRIBUTIONS: Mean, median, variance, standard deviation, The law of large numbers, The Central Limit Theorem, distributions of the sample mean and the sample variance for a normal population, Chi-Square, t and F distributions.

ESTIMATION: Unbiasedness, consistency, minimum mean-squared estimation, the method of moments, the method of maximum likelihood estimation, maximum a posterior probability estimation.

RANDOM PROCESSES: Definition and examples, Stationary processes, wide-sense stationary, strict sense stationary, Ergodicity and ergodic processes, Autocorrelation and autocovariance functions, Power spectral density, various noise processes like additive white Gaussian noise, fractional Gaussian noise, etc., Transmission of random process through linear time-invariant systems, Analysis of communication systems in the presence of noise.

Text Book:

1. A. Papoulis and S. Unnikrishnan Pillai, "*Probability, Random Variables and Stochastic Processes*," Fourth Edition, McGraw Hill.

Reference Books:

1. Steven Kay, "*Intuitive probability and random processes using MATLAB*", Springer.
2. Alberto Leon-Garcia, "*Probability, statistics, and random processes for electrical engineers*," Prentice Hall - Pearson.
3. Sheldon Ross, "*A first course in probability*", Eighth edition, Prentice Hall - Pearson.
4. Sheldon Ross, "*Introduction to Probability and Statistics for Engineers and Scientists*", Academic Press.

CS611	Computer Systems	3-0-0: 3

Objective: The objective of this course is to expose students to the “full span” of the computer network and DBMS. It will give the students a performance perspective towards analysis of computer and communications networks. This course will also provide the major techniques in DBMS implementations.

Learning Outcomes: By the end of this course, students should be able to

1. Demonstrate the operation of various routing protocols and their performance analysis.
2. Illustrate design and implementation of network application, transport and network layer protocols within a simulated/real networking environment.
3. Demonstrate the principles of transaction management

Prerequisite: Fundamental course in Computer Networks and DBMS

Contents:

Introduction to Routing versus forwarding, Static and dynamic routing, Unicast and Multicast Routing. Distance-Vector, Link-State, Shortest path computation, Dijkstra’s algorithm, Network Layer Protocols (IP, ICMP), IP addressing, IPV6, Address binding with ARP, Scalability issues (hierarchical addressing), IP Multicasting

Review to UDP, TCP and SCTP protocols, Multiplexing with TCP and UDP, Principles of congestion control, Approaches to Congestion control, Router-Assisted Congestion Control: Active Queue Management, Quality of service, Flow characteristics, Techniques to improve QoS.

Naming and address schemes (DNS:Resource Discovery, Lookups, IP addresses, Uniform Resource Identifiers, etc.), Distributed applications (client/server, peer-to-peer, cloud, etc.), Distributed Hash Table (DHT) Abstraction and Algorithms, Routing in Overlay Networks, HTTP as an application layer protocol, Electronic mail, File transfer, Remote login.

Overview of Relational Query Languages, The SQL Query Language, Destroying and Altering Relations, Adding and Deleting Tuples, Integrity Constraints, Primary and Candidate Keys in SQL, Foreign Keys, Referential Integrity in SQL, Categories of SQL Commands, Data Definition, Data Manipulation Statements: SELECT - The Basic Form Subqueries, Functions, GROUP BY Feature, Updating the Database, Data Definition Facilities, Views, Embedded SQL *, Declaring Variables and Exceptions, Embedding SQL Statements, Transaction Processing, Consistency and Isolation, Atomicity and Durability, Dynamic SQL.

Introduction of transaction processing system, Storage Structure, Transaction atomicity and durability, Transaction Isolation, serializability and recoverability, Serializability by Locks, Concurrency Control by Timestamps, Concurrency Control by Validation, Database recovery management.

Text Books:

1. "*Database System Concepts*", A. Silberschatz, H. F. Korth, S. Sudharshan, Tata McGraw Hill, New Delhi, 2019.
2. "*Computer Networking: A Top Down Approach*", Kurose and Ross, Addison-Wesley, (2012).

Reference Books:

1. "*Fundamentals of Database Systems*", R. Elmasri, S. B. Navathe, Prentice Hall, New Delhi, 2016.
2. "*Computer Networks*", Tanenbaum , A.S., Prentice Hall (2010).
3. "*Computer Networking with Internet Protocols and Tech*", Stallings, W., Prentice Hall of India (2010).
4. "Data communication and Networking", Forouzan, B.A., McGraw Hill (2006).

CS661	**Algorithms and DBMS Lab**	**0-0-2: 1**

Objective: would be designed to provide hands-on experience in programming data structures and algorithms, followed by advanced database management systems to learn a few systems programming tools, and scripting.

Learning Outcomes: Upon successful completion of this course, student will be able to:

1. Understand the fundamentals behind the techniques, so that students are able to develop algorithms for new problems where these techniques can be applied.
2. Given a practical application, identify the computational issues and apply suitable algorithms to solve it effectively.
3. Demonstrate the use of various DBMS tools available

Prerequisite: Fundamental courses in Data structures and algorithms and Database management systems.

List of Experiments:

A. Algorithm

1. Time Complexity Analysis of various sorting algorithm: Searching Algorithms, Sorting Algorithms - Bubble, Selection, Insertion, Heap, Merge & Quick, Linear sort
2. Tree/Graph based algorithm including Graph Representation, Graph Traversal, Graph Searching
3. Implementation of Balanced Trees – AVL, Red Black Trees, B-Trees
4. Implementation of Shortest Path Algorithms
5. Implementation and Analysis of Greedy Algorithms
6. Implementation and Analysis of Dynamic Algorithms

7. Implementation and Analysis of Randomization Algorithms

B. DBMS

1. Creation of ER diagram for the databases (for ex. A company, an institute)
2. Create a small database (for ex. A company, an institute) in SQL/MySQL and answer various queries
3. Normalization of the database to answer various queries from SQL
4. Designing the rules for database using Triggers
5. Familiarity with MangoDB and postgresql Tool

Text / Reference Books:

1. T.H.Cormen, C.E.Leiserson, R.L.Rivest, "*Introduction to Algorit*hms", Prentice Hall India, 2002.
2. D.E.Knuth, "*The Art of Computer Programming*", Vols. 1 and 3, Addison Wesley, 1998.
3. E.Horowitz, S.Sahni, "*Fundamentals of Computer Algorithms*", Galgotia Publishers, 1984.
4. "*Database System Concepts*", A. Silberschatz, H. F. Korth, S. Sudharshan, Tata McGraw Hill, New Delhi, 2019.
5. "*Fundamentals of Database Systems*", R. Elmasri, S. B. Navathe, Prentice Hall, New Delhi, 2016.
6. "*Introduction to Linear Algebra*", Gilbert Strang, 5th Ed, SIAM, 2016.

Courses: Semester-II

Thesis Mode

Course Code	Course Name	Credits			
		L	T	P	C
CS612	Machine Learning	3	0	0	3
CSXXX	Program Electives*	x	x	x	11
HSXXX	Seminar / Technical Communication / Modular Course#	x	x	x	2
Total		**x**	**x**	**x**	**16**

*****_List of Program Electives along with Course Code and Credits (L-T-P: C) is provided in Appendix-I. Minimum three Program Electives of total 11 credits will be offered by the department._

#*The Course Code and Credits will be provided by the department.*

Courses and Contents

CS612	**Machine Learning**	**3-0-0: 3**

Objective: Machine Learning is a growing field in the area of pattern recognition, natural language processing, speech processing, image processing and vision. This course provides a broad introduction to machine learning architectures. The objectives include: 1. Formulate machine learning problems corresponding to different applications and solve using learning machines based approaches. 2. Read basic as well as current research papers and understand the issues raised by current research.

Learning Outcomes: After studying this course, students will be able to understand a variety of machine learning architectures including deep learning, and use them to solve a few problems. This course will expose students to cutting-edge research starting from a refresher in basics of machine learning, to recent developments in deep learning and random forest.

Prerequisite:
1. Linear algebra
2. Probability and Statistics
3. Computational tools: C/C++/Matlab/Python (any one of these)

Contents:
Problem setting and fundamentals of machine learning, Supervised, unsupervised, semi-supervised learning, Bayes' classifier, K-means, k-NN, principal component analysis (PCA), and least-squares estimation (LSE), minimum mean-squared estimation (MMSE).

Linear and Logistic regressions, concept of over-fitting, regularization.

Concept of convex/non convex functions, optimization methods (gradient descent/conjugate gradient descent/stochastic gradient descent).

Learning machines: Statistical theory of learning, Vapnik-Chervonkis (VC) dimension, Support vector machine (SVM): linear and nonlinear.

Introduction to neural networks and training a network: Back propagation algorithm, Convergence issues, Matrix calculus for training model architectures.

Markov networks (MRF), restricted Boltzmann Machine (RBM) and deep Boltzmann machine (DBM).

Auto encoders: Deep sparse autoencoder (SAE), Deep denoising auto encoder (DAE), Deep contractive autoencoders (CAE).

Convolution neural network (CNN): Deep CNN, All-CNN networks

Deep learning for computer vision problems, Bayesian deep learning

Decision trees and random forests

Fuzzy logic and rough set theory

Text Book:

1. Pattern classification by Duda and Hart, John Wiley.

Reference Books:

1. Deep learning book by Ian Goodfellow, MIT Press.
2. Pattern recognition and machine learning by Bishop, Springer.
3. Basic papers on machine learning by Vapnik, Hinton and other major contributors.
4. Research papers mainly from NIPS, ICML, ICLR, ICMV, ICCV, IGARSS, PReMI, IEEE signal processing magazine, IEEE computational intelligence magazine

Courses: Semester-III

Course Code	Course Name	Credits			
		L	**T**	**P**	**C**
CS791	Comprehensive Viva*	0	0	0	2
CS793	Thesis	0	0	3	2
CS795	Thesis	0	0	6	4
CS797	Thesis	0	0	6	4
HSXXX	Research Methodology / Modular Course / Seminar	x	x	x	2
Total		**x**	**x**	**x**	**14**

Required to complete during the Summer break (May-June). The evaluation will be done in the 3rd Semester.

Courses: Semester-IV

Course Code	Course Name	Credits			
		L	**T**	**P**	**C**
CS792	Thesis	0	0	6	4
CS794	Thesis	0	0	6	4
CS796	Thesis	0	0	6	4
Total		**0**	**0**	**18**	**12**

Appendix-I

List of Program Electives for CSE

Course Code	**Course Name**	**Credits (L-T-P: C)**
CS630	Randomized Algorithms	3-x-x: 3/4
CS631	Game Theory	3-x-x: 3
CS632	Data Mining	3-x-x:4
CS633	Artificial Intelligence	3-x-x: 3/4
CS634	Data Analytics & Visualization	3-x-x: 4
CS635	Deep Learning	3-x-x: 3/4
CS636	Computer Vision	3-x-x: 4
CS637	Image Processing and Analysis	3-x-x: 3/4
CS638	Social Network Analysis	3-x-x: 3
CS639	Parallel Computing	3-x-x: 4
CS640	Advanced Computer Architecture	3-x-x: 3
CS641	Embedded Systems and Internet of Things	3-x-x: 3/4
CS642	Speech Technology	3-x-x: 3
CS643	Advanced Wireless Communications and Networks	3-x-x: 3
CS644	Distributed Systems	3-x-x: 4
CS645	Cloud Computing	3-x-x: 4
CS646	Cryptography and Network Security	3-x-x: 4
CS647	Blockchain Technology	3-x-x: 3/4
CS648	Web Development Techniques	X-1-2: 2
CS649	Cyber Security	3-x-x: 3
CS650	Distributed Databases	3-x-x: 3
CS651	Security Protocols	3-x-x: 3