Mathematics and Computing

Semester III						
S. No	**Course code**	**Course name**	**L**	**T**	**P**	**C**
1	MA401	Introduction to probability theory	3	1.5	0	9
2	CS201T	Automata Theory	3	15	0	9
3	CS407T	Mathematics for Data Science				6
4	CS203T	Design and Analysis of Algorithms	3	0	0	6
5	CS201C	Software Systems Laboratory	2	0	2	6
Total credits						**32**

Mathematics and Computing

Mathematics and Computing

1	**Title of the course** (L-T-P-C)	**Introduction to probability theory (3-1-0-8)**
2	**Pre-requisite courses(s)**	None
3	**Course content**	Combinatorial probability and urn models, Independence of events, Conditional probabilities, Random variables, Distributions, Expectation, Variance and moments, probability generating functions and moment generating functions, Standard discrete distributions (uniform, binomial, Poisson, geometric, hypergeometric), Independence of random variables, Joint and conditional discrete distributions. Univariate densities and distributions, standard univariate densities (normal, exponential, gamma, beta, chi-square, Cauchy). Expectation and moments of continuous random variables. Transformations of univariate random variables. Tchebychev's inequality. Modes of convergence. Law of large numbers. Central limit theorem.
4	**Texts/References**	1. K. L. Chung and F. AitSahlia, Elementary Probability Theory., 4th Edition, Springer Verlag, 2003 2. R. Ash : Basic Probability Theory, Dover publication, 3. W. Feller : Introduction to Probability Theory and its Applications, Volume 1, Wiley-India Edition 4. W. Feller : Introduction to Probability Theory and its Applications, Volume 2, Wiley India Edition

Mathematics and Computing

1	**Title of the course** (L-T-P-C)	**Automata Theory** **(3-1-0-8)**
2	**Pre-requisite courses(s)**	Exposure to Discrete Structures
3	**Course content**	Finite state machines (DFA/NFA/epsilon NFAs), regular expressions. Properties of regular languages. My hill-Nerode Theorem. Non-regularity. Push down automata. Properties of context-free languages. Turing machines:Turing hypothesis, Turing computability, Nondeterministic, multi tape and other versions of Turing machines. Church thesis, recursively enumerable sets and Turing computability. Universal Turing machines. Unsolvability, The halting problem, partial solvability, Turing enumerability, acceptability and decidability, unsolvable problems about Turing Machines. Post`s correspondence problem.
4	**Texts/References**	**1.** Introduction to Automata Theory, Languages and Computation, by John. E. Hopcroft, Rajeev Motwani, J. D. Ullman, 3rd edition. Pearson. 2013. **2.** Elements of the Theory of Computation, by H.R. Lewis and C. H. Papadimitrou, 2nd Edition. Prentice Hall Inc, 1998.

Mathematics and Computing

1	**Title of the course** (L-T-P-C)	**Mathematics for Data Science (3-0-0-6)**
2	**Pre-requisite courses(s)**	Exposure to basic concepts in calculus and linear algebra
3	**Course content**	Introduction to Data science and Motivation for the course. Review of calculus, naTve set theory, notion of limits, ordering, series and its convergence. Introduction to Linear Algebra in Data science, notion of vector space, dimension and rank, algorithms for solving linear equations, importance of norms and notion of convergence, matrix decompositions and its use. Importance of optimization in data science: Birds view of Linear Regression, Multivariate Regression, Logistic Regression etc. Convex Optimization: Convex sets, convex functions, duality theory, different types of optimization problems, Introduction to linear program. Algorithms: Central of gravity method, Gradient descent methods, Nestrov acceleration, mirror descent/ Nestrov dual averaging, stochastic gradient methods, Rmsprop, SIGNSGD, ADAM algorithm etc. Non-convex optimization: Demonstration of convex methods on non- convex problems; merits and disadvantages.
4	**Texts/References**	**1.** C. Bishop, “Pattern Recognition and Machine Learning,” Springer, 2006. **2.** Cambridge university press, 2018 (reprint). for Machine Learning,” Now publisher, 2017.

1	**Title of the course** (L-T-P-C)	**Design and Analysis of Algorithms** **(3-0-0-6)**
2	**Pre-requisite courses(s)**	Computer Programming and Utilization, Discrete Structures, Data Structures and Algorithms , Data Structures and Algorithms Laboratory
3	**Course content**	Syllabus is divided roughly into 8 modules; each module roughly takes two weeks. **Module 1:** Introduction Examples and motivation. Asymptotic complexity: informal concepts, formal notation, examples **Module 2**: Searching in list: binary search, Sorting: insertion sort, selection sort, merge sort, quicksort, stability and other issues. **Module 3:** Divide and conquer: binary search, recurrence relations. nearest pair of points, merge sort, integer multiplication, matrix multiplication. **Module 4**: Graphs: Motivation, BFS, DFS, DFS numbering and applications, directed acyclic graphs, directed acyclic graphs, Shortest paths: unweighted and weighted, Single source shortest paths: Dijkstra, Minimum cost spanning trees: Prim's algorithm, Kruskal's Algorithm **Module 5:** Union-Find data structure, Priority queues, heaps. Heap sort. Dijstra/Prims revisited using heaps, Search Trees: Introduction Traversals, insertions, deletions Balancing **Module 6**: Greedy algorithms: Greedy: Interval scheduling, Proof strategies, Huffman coding. **Module 7:** Dynamic Programming: weighted interval scheduling, memoization, edit distance, longest ascending subsequence. matrix multiplication, shortest paths: Bellman Ford, shortest paths: Floyd Warshall **Module 8:** Intractability: NP completeness, reductions, examples, Misc topics.
4	**Texts/References**	**1.** Algorithms, by Sanjoy Dasgupta, Christos Papadimitriou and Umesh Vazirani, McGraw Hill Education, 2006. **2.** Introduction to Algorithms, 3rd edition, by Cormen, Leiserson, Rivest and Stein, PHI Learning Pvt. Ltd., 2010. **3.** Algorithm Design, 1st edition, by Kleniberg and Tardos, Pearson, 2014.

1	**Title of the course** (L-T-P-C)	**Software Systems Laboratory (1-3-0-8)**
2	**Pre-requisite courses(s)**	--
3	**Course content**	Vim/emacs HTML, CSS **1.** Report and presentation software: latex, beamer, drawingsoftware(e.g. inkscape, xfig, open-office) **2.** IDE (e.g. eclipse, netbeans), code reading,debugging Basic Java Java collections, interfaces **3.** Java threads Java GUI Introduction to documentation: e.g. doxygen/javadocs **4.** Version management: SVN/Git **5.** Unix basics: shell, file system, permissions, process hierarchy, process monitoring, ssh, rsync **6.** Unix tools: e.g. awk, sed, grep, find, head, tail, tar,cut, sort **7.** Bash scripting: I/O redirection, pipes **8.** Python programming **9.** Makefile, libraries and linking **10.** Graph plotting software (e.g., gnuplot) **11.** Profiling tools (e.g., gprof, prof) **12.** Optional topics (may be specific to individualstudents 302222 projects): intro to sockets, basic SQL for data storage,JDBC/pygresql A project would be included which touches upon manyof the above topics, helping students see the connection across seemingly disparate topics. The project is also expected to be a significant load: 20-30 hours of work.
4	**Texts/References**	**1.** Online tutorials for HTML/CSS, Inkscape, OODrawUnix Man Pages for all unix tools, AdvancedBash Scripting Guide from the Linux Documentation Project (www.tldp.org). **2.** The Python Tutorial Online Book(http://docs.python.org/3/tutorial/index.html). **3.** The Java Tutorials (http://docs.oracle.com/javase/tutorial/). **4.** Latex - A document preparation system, 2/e, byLeslie Lamport, Addison-Wesley, 1994.