VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering

CURRICULUM AND SYLLABI

(2018-2019)

M.Tech (CSE) - Specialisation in Cloud Computing

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering

PEOs - M.Tech (CSE) - Specialisation in Cloud Computing

1. Graduates will be prepared to review and understand foundational Concepts in Computer Science, Cloud Computing and Engineering
2. Graduates will be empowered to critically analyze current trends and learn future issues from a system perspective at multiple levels of detail and abstraction
3. Graduates will be enabled to apply the interaction between theory and practice for problem solving based on case studies
4. Graduates will be enabled to pursue lifelong multidisciplinary learning as professional engineers and scientists to effectively communicate technical information, function effectively on teams, and apply computer engineering solutions within a global, societal, and environmental context exploiting cloud infrastructure
5. Graduates will be prepared to critically analyze existing literature, identify the gaps in the existing literature, map the existing problems in Cloud Computing and propose innovative and research oriented solutions.
6. Graduates will be enabled to design, develop and deploy services related to Cloud Computing Paradigm

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering

Student Learning Outcomes (SLO)

M.Tech(CSE) - Specialisation in Cloud Computing

1. Having an ability to apply mathematics and science in engineering applications.
2. Having a clear understanding of the subject related concepts and of contemporary issues
3. Having an ability to be socially intelligent with good SIQ (Social Intelligence Quotient) and EQ (Emotional Quotient)
4. Having Sense-Making Skills of creating unique insights in what is being seen or observed (Higher level thinking skills which cannot be codified)
5. Having design thinking capability
6. Having an ability to design a component or a product applying all the relevant standards and with realistic constraints
7. Having computational thinking (Ability to translate vast data in to abstract concepts and to understand database reasoning)
8. Having Virtual Collaborating ability
9. Having problem solving ability- solving social issues and engineering problems.
10. Having a clear understanding of professional and ethical responsibility
11. Having interest in lifelong learning
12. Having adaptive thinking and adaptability
13. Having cross cultural competency exhibited by working in teams
14. Having an ability to design and conduct experiments, as well as to analyze and interpret data
15. Having an ability to use the social media effectively for productive use
16. Having a good working knowledge of communicating in English
17. Having an ability to use techniques, skills and modern engineering tools necessary for engineering practice
18. Having critical thinking and innovative skills
19. Having a good cognitive load management [discriminate and filter the available data] skills
20. Having a good digital footprint

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering

PSOs - M.Tech(CSE) - Specialisation in Cloud Computing

1. Develop and apply innovative, state-of-the-art practices and technologies and Provide sustainable solutions to the Computer Science and Engineering Problems.
2. To program various issues related to Industry standards that built the environment and also protecting, restoring the natural environment.
3. Apply modern Programming techniques, advanced languages, Lab equipments and management tools so as to complete the Computer Science and Engineering project within specified time and funds.

Curriculum

A. University Core

Course Code	Course Name	L T P J C
MAT5002	Mathematics for Computer Engineering	3 0 0 0 3
EFL5097/ENG5001	Scientific English / Foreign Language	1 0 2 0 2
STS5001	Soft Skills - I	0 0 0 0 1
STS5001	Soft Skills - II	0 0 0 0 1
SET5001	SET Project -I	0 0 0 0 2
SET5002	SET Project - II	0 0 0 0 2
CSE6099	Master's Thesis	0 0 0 0 16
Total Credits - 27		

L T P J C: **L**ecture, **T**utorial, **P**ractical, Pro**J**ect, Total **C**redits

B. University Elective

Course Code	Course Name	L T P J C
	University Elective I	
	University Elective II	
Minimum of 6 Credits to be earned		

C. Programme Core

Course Code	Course Name	L T P J C
CSE5001	Algorithms: Design and Implementations	2 0 2 0 3
CSE5002	Operating Systems and Virtualization	2 0 2 0 3
CSE5004	Computer Networks	2 0 2 0 3
CSE6004	Cloud Computing Eco-systems	2 0 2 4 4
CSE6010	Cloud Application Development and Management	2 0 2 4 4
CSE6003	Web services	2 0 2 0 3
Total Credits - 20		

D. **Programme Electives**

Course Code	Course Name	L T P J C
CSE5006	Multicore Architectures	2 0 2 0 3
CSE6001	Bigdata Frameworks	2 0 2 4 4
CSE6005	Machine Learning	2 0 2 4 4
CSE6006	NoSQL Databases	2 0 2 4 4
CSE6026	IoT on Cloud	2 0 2 4 4
CSE6027	Mobile Cloud Computing	2 0 2 4 4
CSE6028	Cloud Security and Audit	2 0 0 4 3
CSE6029	Cloud Storage Technologies	2 0 2 4 4
CSE6030	Design Thinking	2 0 0 4 3
CSE6031	Cyberspace and Information Technology Laws	2 0 0 0 2
CSE6032	Cloud Computing Paradigm on software Engineering	2 0 0 4 3

Minimum of 17 Credits to be earned

E. **Credits Summary**

Minimum Qualifying Credits	70
University Core	27
University Electives	06
Programme Core	20
Programme Electives	17

	Mathematics for Computer Engineering	L	T	P	J	C
		3	0	0	0	3
Module	**Topics**	**No of Hours**		**SLO**		
1	Proof Techniques: Implications, equivalences, converse, inverse, contrapositive, negation, contradiction, structure, direct proofs, disproofs, natural number induction, structural induction, weak/string induction, recursion, well orderings	6		1,2,7,9		
2	Linear algebra: Eigenvalues and eigenvectors-Gerschgorin Circles– Rutishauser method, Rotation and Reflection matrices- Face Recognition application.	6		1,2,7,9		
3	Number Theory: Divisibility -division algorithm -Euclidean algorithm- Definitions and basic properties of congruences - Solving linear congruences and quadratic congruences, Applications of congruences: The Chinese remainder theorem, Euler's theorem and Fermat's little theorem- Primarily checking,	6		1,2,7,9		
4	Probability :Introduction to random variable -Binomial and Poisson distributions – Normal distribution, Weibull, exponential and Gamma distributions Performance modeling application	6		1,2,7,9		
5	Statistical Measures: Correlation and regression- Covariance– partial and multiple correlation- multiple regression – Time Series data Analysis application.	6		1,2,7,9		
6	Sampling Theory: – small sample tests- student's t –test ,F-test, chi-square test, goodness of fit , independence of attributes, Basic principles of experimentation, Analysis of variance –application using Monte-Carlo methods and decision trees	8		1,2,7,9		
7	Queuing Theory :Introduction-Markov Process-Poisson Process- Pure Berth Process-Death Process-Birth-death processes- Queue notation-Little's theorem-Queuing models M/M/1; M/M/c; M/M/∞	5		1,2,7,9		
8	Expert Lecture: Modular arithmetic-Applications to cryptosystem	2		1,2,7,9		
	Total Hours	45				
	Reference Book 1. Neal Koblitz, A course in number theory and cryptography, Springer reprint (2002). 2. J. P. Tremblay and R Manohar Discrete Mathematical Structures with applications to Computer Science, Tata McGraw Hill (2001). 3. Ronald E. Walpole , Raymond H. Myers Sharon L. Myers Keying E. Ye,Probability and Statistics for Engineers and Scientists (9th Edition), 4. H. A .Taha Operations Research, 9th Edition, PHI (2010). 5. Narasingh Deo, Graph Theory, PHI, 23rd Indian reprint (2002).					

Mode of Teaching	Class room teaching Introducing Units through applications Min of 1 lecture by experts		
Mode of Evaluation	Digital Assignments, Continuous Assessment Test Final Assessment Test		
PROPOSED BY	Dr. Parvathi (SCSE, VIT Chennai), Dr. Vanchinathan and Dr. Kalyani Desikan		
Recommended by the board of studies	09-03-2016		
Date of approval by the academic council			

Course code	Course title	L	T	P	J	C
ENG5001	Fundamentals of Communication Skills	0	0	2	0	1
Pre-requisite	Not cleared EPT (English Proficiency Test)	Syllabus version				
		v.1.0				

Course Objectives:

To enable learners learn basic communication skills - Listening, Speaking, Reading and Writing and apply them for various purposes in academic and social contexts

Expected Course Outcome:

Ability to communicate effectively in social and academic contexts

Student Learning Outcomes (SLO): 16,18

Module:1	Listening	**8 hours**	**SLO: 16**
Understanding Conversation Listening to Speeches Listening for Specific Information			
Module:2	Speaking	**4 hours**	**SLO: 16**
Exchanging Information Describing Activities, Events and Quantity			
Module:3	Reading	**6 hours**	**SLO: 16,18**
Identifying Information Inferring Meaning Interpreting text			
Module:4	Writing: Sentence	**8 hours**	**SLO: 16**
Basic Sentence Structure Connectives Transformation of Sentences Synthesis of Sentences			
Module:5	Writing: Discourse	**4 hours**	**SLO: 16,18**
Instructions Paragraph Transcoding			
	Total Practical hours:	**30 hours**	

Text Book(s)

1. Redston, Chris, Theresa Clementson, and Gillie Cunningham. *Face2face Upper Intermediate Student's Book.* 2013, Cambridge University Press.

Reference Books

1. Chris Juzwiak . *Stepping Stones: A guided approach to writing sentences and Paragraphs (Second Edition)*, 2012, Library of Congress.
2. Clifford A Whitcomb & Leslie E Whitcomb, *Effective Interpersonal and Team Communication Skills for Engineers*, 2013, John Wiley & Sons, Inc., Hoboken: New Jersey.
3. Arun Patil, Henk Eijkman & Ena Bhattacharya, *New Media Communication Skills for Engineers and IT Professionals,*2012, IGI Global, Hershey PA.

4. 5. 6.	Judi Brownell, *Listening: Attitudes, Principles and Skills*, 2016, 5th Edition, Routledge:USA John Langan, Ten Steps to Improving College Reading Skills, 2014, 6th Edition, Townsend Press:USA Redston, Chris, Theresa Clementson, and Gillie Cunningham. *Face2face Upper Intermediate Teacher's Book*. 2013, Cambridge University Press.

Mode of Evaluation: Online Quizzes, Presentation, Role play, Group Discussions, Assignments, Mini Project

List of Challenging Experiments (Indicative)		**SLO: 16,18**
1.	Familiarizing students to adjectives through brainstorming adjectives with all letters of the English alphabet and asking them to add an adjective that starts with the first letter of their name as a prefix.	2 hours
2.	Making students identify their peer who lack Pace, Clarity and Volume during presentation and respond using Symbols.	4 hours
3.	Using Picture as a tool to enhance learners speaking and writing skills	2 hours
4.	Using Music and Songs as tools to enhance pronunciation in the target language / Activities through VIT Community Radio	2 hours
5.	Making students upload their Self- introduction videos in Vimeo.com	4 hours
6.	Brainstorming idiomatic expressions and making them use those in to their writings and day to day conversation	4 hours
7.	Making students Narrate events by adding more descriptive adjectives and add flavor to their language / Activities through VIT Community Radio	4 hours
8	Identifying the root cause of stage fear in learners and providing remedies to make their presentation better	4 hours
9	Identifying common Spelling & Sentence errors in Letter Writing and other day to day conversations	2 hours
10.	Discussing FAQ's in interviews with answers so that the learner gets a better insight in to interviews / Activities through VIT Community Radio	2 hours
	Total Practical Hours	30 hours

Mode of evaluation: Online Quizzes, Presentation, Role play, Group Discussions, Assignments, Mini Project

Recommended by Board of Studies	22-07-2017		
Approved by Academic Council	No. 46	Date	24-08-2017

Course code	Course title	L	T	P	J	C
STS5001	Essentials of Business etiquettes	3	0	0	0	1
Pre-requisite		Syllabus version				
		2				

Course Objectives:

- Having Computational thinking (Ability to translate vast data into abstract concepts and to understand database reasoning)[SLO 7]
- Having problem solving ability- solving social issues and engineering problems [SLO 9]

Expected Course Outcome:

- Enabling students to use relevant aptitude and appropriate language to express themselves
- To communicate the message to the target audience clearly

Student Learning Outcomes (SLO): 7, 9

Module:1	**Business Etiquette: Social and Cultural Etiquette and Writing Company Blogs and Internal Communications and Planning and Writing press release and meeting notes**	**9 hours**	**SLO: 7**

Value, Manners, Customs, Language, Tradition, Building a blog, Developing brand message, FAQs', Assessing Competition, Open and objective Communication, Two way dialogue, Understanding the audience, Identifying, Gathering Information,. Analysis, Determining, Selecting plan, Progress check, Types of planning, Write a short, catchy headline, Get to the Point –summarize your subject in the first paragraph., Body – Make it relevant to your audience,

Module:2	**Study skills – Time management skills**	**3 hours**	**SLO: 9**

Prioritization, Procrastination, Scheduling, Multitasking, Monitoring, Working under pressure and adhering to deadlines

Module:3	**Presentation skills – Preparing presentation and Organizing materials and Maintaining and preparing visual aids and Dealing with questions**	**7 hours**	**SLO: 7**

10 Tips to prepare PowerPoint presentation, Outlining the content, Passing the Elevator Test, Blue sky thinking, Introduction , body and conclusion, Use of Font, Use of Color, Strategic presentation, Importance and types of visual aids, Animation to captivate your audience, Design of posters, Setting out the ground rules, Dealing with interruptions, Staying in control of the questions, Handling difficult questions

Module:4	**Quantitative Ability -L1 – Number properties and Averages and Progressions and Percentages and Ratios**	**11 hours**	**SLO: 9**

Number of factors, Factorials, Remainder Theorem, Unit digit position, Tens digit position, Averages, Weighted Average, Arithmetic Progression, Geometric Progression, Harmonic Progression, Increase & Decrease or successive increase, Types of ratios and proportions			
Module:5	**Reasoning Ability-L1 – Analytical Reasoning**	**8 hours**	**SLO: 9**
Data Arrangement(Linear and circular & Cross Variable Relationship), Blood Relations, Ordering/ranking/grouping, Puzzle test, Selection Decision table			
Module:6	**Verbal Ability-L1 – Vocabulary Building**	**7 hours**	**SLO: 7,9**
Synonyms & Antonyms, One word substitutes, Word Pairs, Spellings, Idioms, Sentence completion, Analogies			
	Total Lecture hours:	**45 hours**	

Reference Books

1.	Kerry Patterson, Joseph Grenny, Ron McMillan, Al Switzler(2001) Crucial Conversations: Tools for Talking When Stakes are High. Bangalore. McGraw-Hill Contemporary
2.	Dale Carnegie,(1936) How to Win Friends and Influence People. New York. Gallery Books
3.	Scott Peck. M(1978) Road Less Travelled. New York City. M. Scott Peck.
4.	FACE(2016) Aptipedia Aptitude Encyclopedia. Delhi. Wiley publications
5.	ETHNUS(2013) Aptimithra. Bangalore. McGraw-Hill Education Pvt. Ltd.

Mode of Evaluation: FAT, Assignments, Projects, Case studies, Role plays,
3 Assessments with Term End FAT (Computer Based Test)

Recommended by Board of Studies	09/06/2017		
Approved by Academic Council	No 45	Date	15/06/2017

CSE5001	Algorithms: Design and Implementation	L,T,P,J,C 2,0,2,0,3
Preamble	This course is a core course focusing on the design and implementation of algorithms.	
Objectives	1. To focus on the design of algorithms in various domains 2. To provide a foundation for designing efficient algorithms. 3. To provide familiarity with main thrusts of work in algorithms- sufficient to give some context for formulating and seeking known solutions to an algorithmic problem.	
Expected Outcomes	After successfully completing the course the student should be able to 1. Solve a problem using Algorithms and design techniques 2. Solve complexities of problems in various domains 3. Implement algorithm, compare their performance characteristics, and estimate their potential effectiveness in applications	
Student Learning Outcomes	1. Having an ability to apply mathematics and science in engineering applications 2. Having a clear understanding of the subject related concepts and of contemporary issues 14. Having an ability to design and conduct experiments, as well as to analyze and interpret data	

Module	Topics	L Hrs	SLO
1	**Introduction** Algorithm design techniques: Divide and Conquer, Brute force, Greedy, Dynamic Programming. Time complexity (asymptotic notation, recurrence relations)	5	1, 2
2	**Network Flows** Maximum Flows, Min-cost Flows, Max-Flow Min-Cut Theorem, Cycle Cancelling Algorithms, Strongly Polynomial-time Analysis, Minimum Cuts without Flows	6	2
3	**Tractable and Intractable Problems** Class complexity: P, NP, NP-Hard, NP-Complete	3	2
4	**Approximation Algorithms** Limits to Approximability, Vertex Cover problem, Set cover problem, Euclidean TSP	3	1, 2
5	**Search Algorithms for Graphs and Trees** Overview of fundamental algorithms, Dijkstra's algorithm, A* search algorithm	4	1, 2, 14

6	**Computational Geometry** Line Segments, Convex hull finding algorithms	4	1, 2
7	**Linear Programming** Representing problems - shortest paths, maximum flow, and minimum-cost flow as linear programming problems. Simplex algorithm	3	1, 2
8	**Recent Trends**	2	2
List of Lab Experiments 1) Implementation of algorithms for problems that can be solved by one or more of the following strategies: Divide and Conquer, Brute force, Greedy, Dynamic Programming. 2) Implementation of Ford Fulkerson method, Edmonds-Karp algorithm for finding maximum flow in a flow network and applying them for solving typical problems such as railway network flow, maximum bipartite matching 3) Implementation of Dinic's strongly polynomial algorithm for computing the maximum flow in a flow network and applying it for solving typical problems 4) Implementation of push-relabel algorithm of Goldberg and Tarjan for finding maximum flow in a flow network and applying it for solving typical problems 5) Applying linear programming for solving maximum flow problem 6) Applying network flow algorithms for baseball elimination and airline scheduling 7) Given a flow network $G = (V, E, s, t)$,where V is the vertex set, E is	30 Hours	14, 17	

the edge set , s and t are source and destination.

An edge of the flow network is called critical if a decrease in the flow over that edge results in a decrease in the total flow of the flow network. An edge of the flow network is called a bottleneck edge if an increase in the flow over that edge results in an increase in the total flow of the flow network. Assume that you are using to compute the maximum flow of the network.

(a) Write a program (any language) to identify all the critical edges.

(b) Write a program (any language) to identify all bottleneck edges in the network.

8) Implementation of solution techniques for the minimum-cost flow problem

9) Design a polynomial time algorithm to compute the solution of a linear programming problem in two dimensions. Your algorithm should convert each constraint of the problem, into a planar region .Use that algorithm to compute the solution of the following problem. Implement your algorithm in any programming language.

A manufacturer of furniture makes two products: chairs and tables. Processing of these products is done on two machines M1 and M2. A chair requires 2 hours on machine M1 and 6 hours on machine M2. A table requires 5 hours on machine M1 and no time on machine M2. There are 16 hours of time per day available on machine M1 and 30 hours on machine M2. Profits gained by manufacturer from a chair and a table are Rs.1 and Rs. 5 respectively. The problem is to maximize the profit for the manufacturer

10) Implementation of algorithms for the vertex cover problem, set cover problem, TSP

11) Implementation of search algorithms for graphs and trees: fundamental algorithms, Dijkstra's algorithm

12) Consider the problem of barricading n sleeping tigers by a fence of shortest length .Forest officials have tranquilized each tiger. Suggest an algorithm for the purpose. You are allowed to assume any information required for your algorithm. Implement your algorithm in any programming language(using convex hull)

13) A simple polygon is defined as a flat shape consisting of straight non-intersecting line segments or sides that are joined pair –wise to from a

closed path. Let $P = \{p_1, p_2, p_3, ... p_n\}$ be a set of points in the two dimensional plane. (a) Write a program to find the simple polygon of P . (b) Write a program (linear time) to convert that the simple polygon of P to a Convex Hull. Students are free to implement the algorithms using programming languages of their choice.		

Reference Books

1. Cormen, Leiserson, Rivest and Stein, "Introduction to Algorithms", 3rd edition, McGraw-Hill, 2009.
2. J.Kleinberg and E.Tardos. "Algorithm Design", Pearson Education, 2009.
3. E. Horowitz, S. Sahni, S.Rajasekaran,"Fundamentals of Computer Algorithms", 2nd edition, Universities Press, 2011.
4. Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin, "Network Flows: Theory, Algorithms, and Applications", Pearson Education, 2014.
5. George T. Heineman, Gary Pollice, Stanley Selkow, "Algorithms in a nutshell", O'Reilly Media, 2nd edition, 2016.

Algorithms and Complexity

Knowledge Areas that contain topics and learning outcomes covered in the course

Knowledge Area	**Total Hours of Coverage (Theory + Practical)**
CS:AL/Basic Analysis	1
CS:AL/Algorithmic Strategies	4 + 8
CS:SDF/Algorithms and Design	5
CS:AL/Basic Automata Computability and Complexity	3
CS:AL/Fundamental Data Structures and Algorithms	2 + 3
CS:IS/Basic Search Strategies	2
CS:AL/Advanced Data Structures Algorithms and Analysis	17 + 19

Body of Knowledge coverage

KA	Knowledge Unit	Topics Covered	Hours
CS:AL/Basic Analysis CS:AL/Algorithmic Strategies CS:SDF/Algorithms and Design	**Introduction**	Algorithm design techniques: Divide and Conquer, Brute force, Greedy, Dynamic Programming. Time complexity (asymptotic notation, recurrence relations)	5
CS:AL/Advanced Data Structures Algorithms and Analysis	**Network Flows**	Maximum Flows, Min-cost Flows, Max-Flow Min-Cut Theorem, Cycle Cancelling Algorithms, Strongly Polynomial-time Analysis, Minimum Cuts without Flows	6
CS:AL/Basic Automata Computability and Complexity	**Tractable and Intractable Problems**	Class complexity: P, NP, NP-Hard, NP-Complete	3
CS:AL/Advanced Data Structures Algorithms and Analysis	**Approximation Algorithms**	Limits to Approximability, Vertex Cover problem, Set cover problem, Euclidean TSP	3
CS:AL/Fundamental Data Structures and Algorithms CS:IS/Basic Search Strategies	**Search Algorithms for Graphs and Trees**	Overview of fundamental algorithms, Dijkstra's algorithm, A* search algorithm	4
CS:AL/Advanced Data Structures Algorithms and Analysis	**Computational Geometry**	Line Segments, Convex hull finding algorithms	4
CS:AL/Advanced Data Structures Algorithms and Analysis	**Linear Programming**	Representing problems such as shortest paths, maximum flow, and minimum-cost flow as linear programming problems. Simplex algorithm	3
	Recent Trends		2
		Total hours	30

Where does the course fit in the curriculum?

This course

- Is a core course. It is compulsory.
- Is opted by all students.

What is covered in the course?

The course is a core course on algorithms and complexity. It has eight modules. This course includes introduction, network flows, tractable and intractable problems, approximation algorithms, search algorithms for graphs and trees, computational geometry, linear programming and recent trends. The course lays emphasis on techniques used to design algorithms. The goal is for students to be able to apply all of the above to designing solutions for real-world problems.

Module 1: Introduction

This module introduces some algorithm design techniques: Divide and Conquer, Brute force, Greedy, Dynamic Programming. Time complexity is explained by means of asymptotic notation, and recurrence relations.

Module 2: Network Flows

This module discusses maximum flows, min-cost flows, the Max-Flow Min-Cut theorem, cycle cancelling algorithms, strongly polynomial-time analysis, and minimum cuts without flows.

Module 3: Tractable and Intractable Problems

This module discusses tractable and intractable problems by introducing the complexity classes P, NP, NP-Hard, and NP-Complete.

Module 4: Approximation Algorithms

This module discusses the limits to approximability, the vertex cover problem, the set cover problem, and the Euclidean Travelling Salesman Problem.

Module 5: Search Algorithms for Graphs and Trees

This module provides an overview of fundamental algorithms (breadth first search and depth first search), Dijkstra's algorithm, and the A* search algorithm.

Module 6: Computational Geometry

This module discusses line segment properties and some convex hull finding algorithms.

Module 7: Linear Programming

This module discusses representation of problems such as shortest paths, maximum flow, and minimum-cost flow as linear programming problems. It looks at the Simplex algorithm.

Module 8: Recent Trends

This module discusses some recent trends in algorithms.

What is the format of the course?

This course is a face to face course. The number of lecture hours is 30. The number of lab hours is 30. The course will have a combination of lectures, in-class discussion, case studies, guest-lectures, mandatory off-class reading, quizzes, assignments, and tests. The course will have lab sessions. The course will attempt to include video lectures, flipped classroom strategy, virtual

classes, and encourage MOOC learning,

How are students assessed?

- The students will be assessed on the basis of classroom discussion, quizzes, continuous assessment, and the final assessment test. Adequate time will be given to the students to complete their assignments, exercises, lab exercises, quizzes, and tests.
- For the programming assignments, students are strongly encouraged to use programming languages of their choice.

Session wise plan

Student Outcomes Covered: 1,2,5,7, 9,14,17,18

Sl. No	Topic Covered	Class Hour	Lab Hour	Levels of mastery	Reference Book	Remarks
1	**Introduction** Algorithm design techniques: Divide and Conquer, Brute force	2	4	Familiarity Usage	1, 2, 3	
2	Greedy, Dynamic Programming	2	4	Familiarity Usage	1, 2, 3	
3	Time complexity (asymptotic notation, recurrence relations)	1		Familiarity Usage	1, 2, 3	
4	**Network Flows** Maximum Flows, Min-cost Flows	2	4	Familiarity Usage	1, 2, 4	
5	Max-Flow Min-Cut Theorem	1		Familiarity	1, 4	
6	Cycle Cancelling Algorithms	1	2	Familiarity Usage	4	
7	Strongly Polynomial-time Analysis	1	2	Familiarity Usage	4	
8	Minimum Cuts without Flows	1		Familiarity	4	
9	**Tractable and Intractable Problems** Class complexity: P, NP	1		Familiarity	1, 2, 3, 4	
10	NP-Hard, NP-Complete	2		Familiarity	1, 2, 3, 4	
11	**Approximation Algorithms** Limits to Approximability, Vertex Cover problem	1	1	Familiarity Usage	1, 2	
12	Set cover problem	1	1	Familiarity Usage	1, 2	
13	Euclidean TSP	1	1	Familiarity Usage	1	

14	**Search Algorithms for Graphs and Trees** Overview of fundamental algorithms	1	2	Familiarity Usage	1, 2, 3, 5	
15	Dijkstra's algorithm	1	1	Familiarity Usage	1, 3	
16	A* search algorithm	2		Familiarity	5	
17	**Computational Geometry** Line Segments	2		Familiarity	1	
18	Convex hull finding algorithms	2	2	Familiarity Usage	1, 5	
19	**Linear Programming** Representing problems such as shortest paths, maximum flow, and minimum-cost flow as linear	1	2	Familiarity Usage	1	
	programming problems					
20	Simplex algorithm	2	4	Familiarity Usage	1, 4	
21	**Recent Trends**	2		Familiarity		
Total hours covered		30	30			

CSE5002	Operating systems and Virtualization	L,T,P,J,C 2,0,2,0,3
Objectives	This course introduces to virtualization, operating systems fundamental concepts and its technologies. This course provides the skills to write programs that interact with operating systems components such as Processes, Thread, Memory during concurrent execution. This course provides the skills and knowledge necessary to implement, provisioning and administer server and desktop virtualization.	
Expected Outcomes	After successfully completing the course the student should be able to • Categorize the levels of abstraction in a computer system correspond to implementation layers in both hardware and software. • Apply and design the procedure used for concurrency and memory management. • Comprehend the basics of virtualization and to differentiate types of Virtualization. • Develop and provision server and desktop virtualization • Analyze the inner-working of a Virtual Machine and its Management	
Student Learning Outcomes	• Having a clear understanding of the subject related concepts and of contemporary issues (2) • Having design thinking capability(5) • Having an ability to use techniques, skills and modern engineering tools necessary for engineering practice(17)	

Module	Topics	L Hrs	SLO
1	Computer system architecture a layered view with interfaces - Glenford Myer, Monolithic Linux & Hybrid Windows 10 kernels Layered architecture of operating system and core functionalities	2	2
2	Process Introduction, Process Operations, States, Context switching, Data Structures(Process Control Block (PCB), Process Scheduling: Multi-Level Feedback Queue, Multi-processor Scheduling, Deadlocks and its detection	4	5
3	Memory Introduction, Address Spaces, Memory API, Address Translation, Paging - Faster Translations (TLB), Smaller Tables. Virtual Memory System in x86	4	5
4	Concurrency	4	2

	Introduction, Thread Models, Thread API, Building & Evaluating a Lock, Test And Set, Two phase lock, Classical problems handling using semaphore		
4	Persistence - File Organization: The i-node, Crash Consistency & file security	2	5
5	Virtual Machines Process and System VMs Taxonomy of VMs	2	2
6	Types of Virtualization- Hardware Emulation Full Virtualization with binary translation, Hardware assisted, Operating System Virtualization OS assisted /Para virtualization	4	17
7	Hypervisor Type 1, Type 2	1	17
	Para-virtualization Server Virtualization, Desktop Virtualization	3	17
	Overview VM portability Clones, Templates, Snapshots, OVF, Hot and Cold Cloning Protecting & Increasing Availability Light Weight Virtual machine: Container / Docker	2	2
8	Recent Trends	2	2
		30	

List of Lab Experiments

Each experiment should require the student to submit a system analysis & design document that describes the details of the experiment. The experiment may be submitted before the next lab if not completed within class hours. Collaboration and discussion with co-students on the experiments is encouraged. However plagiarism will be penalized severely as per University regulations.
SLO: 14, 17

1. Retrieval of System data file and its information
2. Write a program to simulate multi-level queue scheduling algorithm

(ex) All the processes in the system are divided into two categories – system processes and user processes. System processes are to be given higher priority than user processes. Use FCFS scheduling for the processes in each queue.

3. Program to demonstrate Process Life Cycle
4. Process Synchronization using Binary and Counting Semaphore
5. Deadlock Avoidance Using Semaphores: Implement the deadlock-free solution to Dining Philosophers problem to illustrate the problem of deadlock and/or starvation that can occur when many synchronized threads are competing for limited resources.

7. Parallel Thread management using pthread library.

Implement a data parallelism using multi- threading. Application should have Thread create, Thread synchronization, Thread termination. In the program, every thread must return the value and must be synchronized in the main function. Final consolidation should be done by main thread (main function)

8. Type II Hypervisor setup and configuration
   Create Virtual Machines (VM)
   Create Snapshot of all Virtual Machines
   Create clone of VM
   Configure Networking
9. Type I Hypervisor setup and configuration
   Virtual machines on Xen Migration of VMs
   Create a Virtual Network Switch
   Create Differencing Hard Disks for use by Virtual Machines

Associate Hard Disks with Virtual Machines

10. VM migration and Backup and recovery virtual machines in type I
11. Implementation of Server virtualization – multiple servers on single system
12. Implementation of Desktop virtualization

Reference Books

1. Thomas Anderson, Michael Dahlin , "Operating Systems: Principles and Practice" , Second Edition, Recursive Books,2014
2. William Stallings," Operating Systems: Internals and Design Principles", 8th Edition, 2014.
3. Remzi H. Arpaci-Dusseau and Andrea C, "Operating Systems: Three Easy Pieces", . Arpaci-Dusseau , University of Wisconsin – Madison,2015
4. A. Silberschatz and P. Galvin. "Operating System Concepts". Eight Edition, John Wiley & Sons, 2008
5. Matthew Portnoy, "Virtualization Essentials", John Wiley & Sons Inc; 2nd Edition edition (8 September 2016)
6. Smith, Nair, "Virtual Machines: Versatile Platforms for Systems and Processes", Morgan Kaufmann Publishers (2005)

Knowledge Areas that contain topics and learning outcomes covered in the course

Knowledge Area	Total Hours of Coverage
CS-OS	30

Body of Knowledge coverage

KA	Knowledge Unit	Topics Covered	Hours

CS	OS/Overview of Operating Systems	Computer system architecture a layered view with interfaces - Glenford Myer, Monolithic Linux & Hybrid Windows 10 kernels. Layered architecture of operating system and core function	**2**
CS	OS/Operating System Principles	Introduction, Process Operations, States, Context switching, Data Structures(Process Control Block (PCB),	**2**
CS	OS/Scheduling and Dispatch	Process Scheduling: Multi-Level Feedback Queue, Multi-processor Scheduling, Deadlocks and its detection	**2**
CS	OS/Memory Management	Address Spaces, Memory API, Address Translation, Paging - Faster Translations (TLB), Smaller Tables. Virtual Memory System in x86	**4**
CS	OS/Concurrency	Thread Models, Thread API, Building & Evaluating a Lock, Test And Set, Two phase lock, Classical problems handling using semaphore	**4**
CS	OS/File Systems	Persistence - File Organization: The i-node, Crash Consistency & file security	**2**
CS	OS/Virtual Machines	Virtual Machines, Types of Virtualization, Hypervisor - Type 1, Type 2, Server Virtualization, Desktop Virtualization, Overview VM portability, Light Weight Virtual machine	**14**

Where does the course fit in the curriculum?

[*In what year do students commonly take the course? Is it compulsory? Does it have pre-requisites, required following courses? How many students take it?*]
This course is a

- An ElectiveCourse
- Suitable from first semester onwards.

What is the format of the course?

[*Is it face to face, online or blended? How many contact hours? Does it have lectures, lab sessions, discussion classes?*]

This Course is designed with 100 minutes of in-classroom sessions per week, 60 minutes of video/reading instructional material per week, 100 minutes of lab hours per week, as well as 200 minutes of non-contact time spent on implementing course related project. Generally this course should have the combination of lectures, in-class discussion, case studies, guest-lectures, mandatory of pre-class reading material, quizzes.

How are students assessed?

[*What type, and number, of assignments are students are expected to do? (papers, problem sets, programming projects, etc.). How long do you expect students to spend on completing assessed work?]*

- Students are assessed on a combination group activities, classroom discussion, and continuous, final assessment tests.
- Additional weightage will be given based on their rank in crowd sourced projects/ Kaggle like competitions and certificate of completion of a related MOOC course.

Session wise Plan

Students Outcome coverage : 2,5,8,11,12,17

Class Hour	Topics Coverage	Levels of Mastery	Reference Book	Remarks
2	Computer system architecture a layered view with interfaces - Glenford Myer, Monolithic Linux & Hybrid Windows 10 kernels. Layered architecture of operating system and core function	Familiarity	6	
2	Introduction, Process Operations, States, Context switching, Data Structures(Process Control Block (PCB),	Familiarity	3	
2	Process Scheduling: Multi-Level Feedback Queue, Multi-processor Scheduling, Deadlocks and its detection	Usage	3	
4	Address Spaces, Memory API, Address Translation, Paging - Faster Translations (TLB), Smaller Tables. Virtual Memory System in x86	Familiarity	3	
4	Thread Models, Thread API, Building & Evaluating a Lock, Test And Set, Two phase lock, Classical problems handling using semaphore	Usage	3	
2	Persistence - File Organization: The i-node, Crash Consistency & file security	Familiarity	4	
12	Virtual Machines, Types of Virtualization, Hypervisor - Type 1, Type 2, Server Virtualization, Desktop Virtualization, Overview VM portability, Light Weight Virtual machine	Usage	5,6	

2	Recent Trends			
	Theory : 30 Hours (2 Credit hours /week D 15 Weeks schedule) Laboratory : 30 Hours (1 Credit hours / week)			

CSE5004	COMPUTER NETWORKS	L T P J C 2 0 2 0 3
Objectives	• Learn the division of network functionalities into layers. • Be familiar with the components required to build different types of networks and protocol • Understand the flow control and congestion control algorithms • Design the subnet and super net for particular organization	
Expected Outcomes	After successfully completing the course the student should be able to • Explore the basics of Computer Networks and Various Protocols. • Administrate a network and flow of information • Plan and design networks	
Student Learning	• Having a clear understanding of the subject related concepts and of contemporary issues (2) • Having an ability to design a component or a product applying all the relevant standards and with realistic constraints (6) • Having an ability to design and conduct experiments, as well as to analyze and interpret data (14)	

MODULE	TOPICS	LHRS	SLO
1	**INTRODUCTION**	6	
	Network models, Addressing: Classful and Classless, Routing Protocols : unicast, multicast, Congestion control, Host configuration: DHCP, DNS		2
2	**NETWORK MANAGEMENT**	4	
	SNMP : Management Components, SMI, MIB, Configuration Management – Fault management – Performance Management – Accounting Management, Case studies		2
3	**SOFTWARE DEFINED NETWORKS**	5	
	SDN Data plane, Control Plane, Application Plane. SDN security attack vectors and SDN Harderning, Overlay model and network model for cloud computing		2
4	**Network Functions Virtualization**	3	
	Concepts, Benefits, requirements, Reference architecture, Management, Functionality and Infrastructure		6
5	**Network Virtualization**	4	
	Virtual LAN, Virtual Private Networks: IPSEC, MPLS, Network Virtualization Architecture and Benefits		6,14
6	**Security**	2	
	Security requirements, Threats to SDN, SDN security, NFV Security and its techniques		2
7	**Network Design Implications of QoS and QoE**	4	

	QoS Architectural Framework, SLA, IP Performance metrics, QoE: Strategies, Measurements, QoE/QoS Mapping models		2
8	**RECENT TRENDS**	**2**	6
	TOTAL HOURS	**30**	
	laboratory exercises 1. Study of different types of Network cables and Practically implement the cross-wired cable and straight through cable using crimping tool. **Using commands and tools** 2. Study of Network Devices in Detail. 3. Study of network IP. 4. Web NMS (SNMP based) 5. Network Simulators (Programs using NS3) . 6. Implementation of routing protocols in MANETs 7. Network trouble shooting 8. Programs using network packet tracers 9. SDN Applications and Use Cases (e.g. Multicasting) 10. Network Virtualization and Slicing (e.g. FlowVisor) 11. Network Function Virtualization (NFV)		14, 17

REFERENCE BOOKS:

1. William Stallings, “Data and Computer Communication”, Sixth Edition, Pearson Education, 2000
2. Behrouz A. Forouzan, ,“TCP/IP Protocol Suite”,Tata McGraw Hill edition, Fourth Edition. 2015.
3. William Stallings, ”Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud" Pearson,2015
4. James F. Kuross, Keith W. Ross, “Computer Networking, A Top-Down Approach Featuring the Internet”, Third Edition, Addison Wesley, 2004.
5. Andrew S. Tanenbaum, “Computer Networks”, Fourth Edition, 2003.
6. Forouzan, A. Behrouz. “Data Communications & Networking (sie)”. Tata McGraw-Hill Education, 2006.
7. Peterson and Bruce S. Davie Larry L.,”Computer Networks – A Systems approach” -, Morgan Kaufmann Publishers, Elsevier, 5th edition, 2012.

COMPUTER NETWORKS

Knowledge Areas that contain topics and learning outcomes covered in the course

Knowledge Area	Total Hours of Coverage
CS-NC	12
CE-NWK	18

Body of Knowledge coverage

[List the Knowledge Units covered in whole or in part in the course. If in part, please indicate which topics and/or learning outcomes are covered. For those not covered, you might want to indicate whether they are covered in another course or not covered in your curriculum at all. This section will likely be the most time-consuming to complete, but is the most valuable for educators planning to adopt the CS2013 guidelines.]

KA	Knowledge Unit	Topics Covered	Hours
CS	**NC**	Protocols and Standards, The OSI model, TCP/IP protocol suite, point-to-point WANs, Switching, Addressing-IPv4, Ipv6 addresses: Classsful Addressing, Subnetting/Supernetting and Classless Addressing, Special addresses, Wired and Wireless LANs, VLANs	**7**
CE	**NWK**	Internet Protocol, Address resolution protocol, Internet control message protocol, Unicast routing protocols(RIP, OSPF, and BGP), Multicasting and Multicast routing protocols (DVMRP, MOSPF, PIM-DM and PIM-SM)	
CS	**NC**	Design issues of transport layer , UDP, TCP, flow control, Error control, congestion control, Client-server paradigm, Host configuration: DHCP, DNS	
CE	**NWK**	SNMP : Management Components, SMI, MIB, Configuration Management – Fault management – Performance Management – Security Management – Accounting Management, Case studies	**4**
CE	**NWK**	SDN Data plane, Control Plane, Application Plane. SDN security attack vectors and SDN Harderning, Overlay model and network model for cloud computing, SDN security Network Functions Virtualization (NFV):Concepts, Benefits, requirements, Reference architecture, Management, Functionality and Infrastructure ,NFV security NETWORK VIRUTALIZAITON:Virtual LAN, Virtual Private Networks: IPSEC, MPLS,Network Virtualization Architecture and Benefits Network Design Implications of QoS and QoE:QoS Architectural	**18**

		Framework, SLA, IP Performance metrics, QoE: Strategies, Measurements, QoE/QoS Mapping models	

What is covered in the course?

[A short description, and/or a concise list of topics - possibly from your course syllabus.(This is likely to be your longest answer)]

This course focuses on the fundamentals of computer networks and communication. One goal is to give some insight into the rationale of why networks are structured the way they are today and to understand the issues facing the designers of next-generation data networks. Much of the course focuses on Networking Standards And Specification, TCP/IP Protocol Suite, Wireless Lan, Software Defined Networking, Network Monitoring & Management.

Module 1 - INTRODUCTION

Network models, Addressing: Classful and Classless, Routing Protocols : unicast, multicast, Congestion control, Host configuration: DHCP,

DNS **Module 2 – SNMP**

SNMP : Management Components, SMI, MIB, Configuration Management – Fault management – Performance Management – Security Management – Accounting Management, Case studies

Module 3 – SOFTWARE DEFINED NETWORK

SDN Data plane, Control Plane, Application Plane. SDN security attack vectors and SDN Harderning, Overlay model and network model for cloud computing, SDN security

Module 4 – Network Functions Virtualization (NFV)

Concepts, Benefits, requirements, Reference architecture, Management, Functionality and Infrastructure ,NFV security

Module 5 – NETWORK VIRUTALIZAITON

Virtual LAN, Virtual Private Networks: IPSEC, MPLS,Network Virtualization Architecture and Benefits

Module 6: Security

Security requirements, Threats to SDN, SDN security, NFV Security and its techniques

Module 7 –Network Design Implications of QoS and QoE

QoS Architectural Framework, SLA, IP Performance metrics, QoE: Strategies, Measurements, QoE/QoS Mapping models

Module 8 – RECENT TRENDS

What is the format of the course?

[*Is it face to face, online or blended? How many contact hours? Does it have lectures, lab sessions, discussion classes?*]

This Course is designed with 100 minutes of in-classroom sessions per week, 100 minutes of lab hours per week. Generally this course should have the combination of lectures, in-class discussion, case studies, guest-lectures, mandatory of pre-class reading material, quizzes.

How are students assessed?

[*What type, and number, of assignments are students are expected to do? (papers, problem sets, programming projects, etc.). How long do you expect students to spend on completing assessed work?]*

- Students are assessed on a combination group activities, classroom discussion, and continuous, final assessment tests.

- Additional weightage will be given based on their rank in crowd sourced projects/ Kaggle like competitions and certificate of completion of a related MOOC course.

Session wise Plan

Students Outcome coverage : 2,5,6,14

Class Hour	Lab Hour	Topics Coverage	Levels of Mastery	Reference Book	Remarks
2		Network models, Addressing: Classful and Classless	Familiarity	1,4	
2	2	Routing Protocols : unicast, multicast Congestion control,	Usage	1,4	Lab component
2	2	Host configuration: DHCP, DNS	Usage	1,4,5	Lab component
2	4	SNMP : Management Components, SMI, MIB, Configuration Management	Familiarity	2,7	Lab component
2		Fault management- Performance Management	Usage	2,7	Lab component
2	6	SDN Data plane, Control Plane, Application Plane.	Familiarity & Usage	3	Lab component
2	4	SDN security attack vectors and SDN Harderning	Familiarity & Usage	3	Lab component
1		Overlay model and network model for cloud computing	Familiarity	3	
1	4	Network Functions Virtualization (NFV): Concepts and Architecture	Familiarity & Usage	3	Lab component
2		Management, Functionality and Infrastructure	Familiarity	3	
2	6	**Network Virtualization:** Virtual LAN, Virtual Private Networks: IPSEC, MPLS,	Familiarity & Usage	3	Lab component
2		Network Virtualization Architecture and Benefits	Familiarity	3	
2		**Security:** Security requirements, Threats to SDN, SDN security, NFV Security and its techniques	Familiarity	3	
2		**Network Design Implications of QoS and QoE:** QoS Architectural Framework, SLA, IP Performance metrics	Familiarity	3	
2		QoE: Strategies, Measurements, QoE/QoS Mapping models	Familiarity	3	
2		**RECENT TRENDS**	Familiarity		
30 Hours	30 Hours				

CSE6004	Cloud Computing Eco-Systems	L,T,P,J,C 2,0,2,4,4
Objectives	• The fundamentals and essentials of Cloud Computing • The ability to adopt Cloud Computing tools and services for real life scenarios • An exposure to use commercial systems such as Google Apps, Microsoft Azure and Amazon Web Services etc. • To impart knowledge in applications of cloud computing	
Expected Outcomes	After successfully completing the course the student should be able to • Deploy real-world applications onto the cloud • Differentiate between Public, Private and hybrid clouds • Formulate devOps based design and development of cloud applications • Appreciate the requirements of various service paradigms in cloud computing • Describe Datacenter requirements for the cloud	
Student Learning Outcomes	• Having a clear understanding of subject related concepts and of contemporary issues. (2) • Having design thinking capability (5) • Having problem solving ability- solving social issues and engineering problems (9)	

Module	Topics	L Hrs	SLO
1	**Introduction to Cloud Computing** Cloud Computing Overview: Characteristics – challenges, benefits, limitations, Evolution of Cloud Computing, Cloud computing architecture, Cloud Reference Model (NIST Architecutre), open group cloud ecosystem reference model	(4Hrs)	2
2	**Cloud Service Models** Service Models, Characteristics, Benefits, Enabling Technologies (IaaS/PaaS/SaaS) Case Study : SaaS : Salesforce.com, Online Collaboration Serivces Iaas : AWS, OpenStack PaaS : IBM Bluemix, GAE	(4Hrs)	2

3	**Cloud Deployment Models and Resource Management** Public/Private/Multi-cloud deployments, Shared Resources – Resource Pool – Usage and Administration Portal – Resource Management – Elastic Environment – Resilient Environment – Security – Workload Distribution – Dynamic provisioning.	(5Hrs)	5
4	**Cloud Eco Systems** The concept of a cloud ecosystem, Actors and Roles in the Cloud Eco System, Cloud adoption vision, Identifying your use cases, Developing your plan, Understanding the implications of Cloud Service Layers, Utilizing cloud to gain strategic advantage	(2Hrs)	9
5.	**Introduction to DevOps** Understanding the Business Needs for Devops, DevOps Culture, Process and Technology in DevOps,DevOps Myths, Path to DevOps Adoption, Plan and Measure, Develop and Test (collaborative and continuous), Release and Deploy Monitor and Optimize (Continuous and Customer Feedback)	(5Hrs)	2 & 5
6.	**DevOps Capabilities** Open stack Architecture, Open stack Compute, Network, Object storage in detail, Automation, Open stack installations.	(4Hrs)	2 & 5
7.	**Cloud Data Centers** Historical Perspective, Datacenter Components, Design Considerations, Power Calculations, Evolution of Data Centers	(4Hrs)	5
8.	Recent Trends	(2Hrs)	2
		30	
List of Experiments 1. Cisco simulator – VLAN design, Routing, Sub netting, Gateway configuration 2. Virtual box based Webserver creation, Images/Snapshots access webpage from 2nd VM on another subnet work 3. EC2 AWS – S3 bucket based static webpages. Use this page as a start page vis EC2 webserver 4. AWS – Local balancing and auto scaling 5. DaaS – Deployment of a basic web app and add additional functionality(Java scripts based) 6. PaaS – IOT – Mobile sensor based IOT application hosted via PaaS environment 7. SaaS – Deployment of any SaaS application for a online collaborative tool *8.* Deployment of Open stack or Virtual box from the scratch *(2 Lab sessions)* *9.* DevOps deployment of library automation etc. on the cloud platform with one complete upgrade of the application *(2 Lab sessions)* 10. Automating Open stack deployment using Chef/Puppet configuration for 4 node/ 5 node/ HA clusters		30	14, 17

Reference Books

1. *Kai Hwang , Geoffrey Fox, Jack J. Dongarra, Morgan Kaufmann, Distributed and Cloud Computing: From Parallel Processing to the Internet of Things, 1st Edition, 2011.*
2. *Gautham Shroff, "Enterprise Cloud Computing: Technology, Architecture, Applications", Cambridge press, 2010.*
3. *Rajkumar Buyya, James Broberg, Andrzej Goscinski," Cloud Computing Principles and Paradigms", John Wiley & Sons, 2011.*
4. *John Rhoton and Risto Haukiojal, "Cloud Computing Architectured : Solution Design Handbook", Recursive Press, 2013.*
5. *Dinkar Sitaram, Geetha Manjunathan, " Moving to the Cloud: Developing Apps in the new world of Cloud Computing", Syngress, 2012.*
6. *Introduction to Cloud Eco Systems*
7. *"DevOps for Dummies" by Sanjeev Sharma*
8. *Mandis Walls, "Building a DevOps Culture", O'relly*
9. *"Handbook on Data Centers" Samee. U. Khan, Albert. Y. Zomaya, Springer*
10. https://www.cloudfoundry.org/
11. *https://puppet.com/blog/implement-a-message-queue-your-cloud-application*
12. *www.cloudpatterns.org*
13. *www.IBMCloud.com*

Session wise Plan

Students Outcome coverage : 2,5,6,7,9,11,14,17

Class Hour	Topics Coverage	Levels of Mastery	Reference Book	Remarks
(4Hrs)	**Introduction to Cloud Computing** Cloud Computing Overview: Characteristics – challenges, benefits, limitations, Evolution of Cloud Computing, Cloud computing architecture, Cloud Reference Model (NIST Architecutre)	Familiar	1,2	
(4Hrs)	**Cloud Service Models** Service Model, Characteristics, Benifits, Enabling Technologies(IaaS/PaaS/SaaS) Case Study : SaaS : Salesforce.com, CRM, Online Collaboration Serivces Iaas : AWS, OpenStack PaaS : IBM Bluemix, GAE	Familiar	1	
(4Hrs)	**Introduction to Public and Private Clouds** Shared Resources – Resource Pool – Usage and Administration Portal – Usage Monitor – Resource Management – Intelligent Automation Engine – Elastic Environment – Resilient Environment – Security – Workload Distribution – Dynamic provisioning.	Usage	2,4,13	
(4Hrs)	**Cloud Eco Systems** The concept of a cloud ecosystem, Actors and Roles of Cloud Eco System, Cloud adoption vision, Identifying your use cases, Developing your plan, Understanding the implications of Cloud Service Layers, Utilizing cloud to gain strategic advantage	Usage	15	
(4Hrs)	**Introduction to DevOps** Understanding the Business Needs for Devops, Recognizing the Business Values of DevOps, DevOps Culture, Process and Technology in DevOps,DevOps Myths	Usage	9	
(4Hrs)	**DevOps Capabilities** Path to DevOps Adoptation, Plan and Measure,	Usage	9,10	

Class Hour	Topics Coverage	Levels of Mastery	Reference Book	Remarks
	Develop and Test (collaborative and continuous), Release and Deploy Monitor and Optimize (Continuous and Customer Feedback)			
(5Hrs)	**Cloud Data Centers** Historical Perspective, Datacenter Components, Design Considerations, Power Calculations, Evolution of Data Centers	Usage	11	
(2Hrs)	**Recent Trends**	Usage	Internet	
Theory : 30 Hours (2 Credit hours /week) **Lab** : 30 Hours (1 Credit hour / week) **Project** : 60 Non-contact Hours (1 credit hours / week) *(15 Weeks schedule)*				

CSE6010	Cloud Application Development and Management	L,T,P,J,C 2,0,2,4,4
Objectives	• To enable student to develop and launch applications in the cloud environment • To understand the various frameworks and APIs that can be used for developing cloud based applications • To use Cloud application management and management tools are used to analyze digital service ecosystems and digital product life-cycles.	
Expected Outcomes	After successfully completing the course the student should be able to • Design, Develop & Deploy real-world applications in the cloud computing platforms they have learnt • Demonstrate the ability to access the various cloud platforms used. • Describe the standardization process of cloud platform and various API's • Describe the methods for managing the data in cloud and demonstrate the concepts of automation, provisioning using puppet tool. • Develop Applications in the cloud platform • Analyze and use of an appropriate framework and APIs for the task • Design dashboards for management across cloud based service	
Student Learning Outcomes	✓ Having a clear understanding of the subject related concepts and of contemporary issues (2) ✓ Having an ability to design a component or a product applying all the relevant standards and with realistic constraints(6) Having problem solving ability- solving social issues and engineering problems (9)	

Module	Topics	L Hrs	SLO
1	**Basic concepts & techniques** Business case for implementing cloud application, Requirements collection for cloud application development, Cloud service models and deployment models, Open challenges in Cloud Computing: Cloud inter-operability and standards, scalability and fault tolerance, security, trust and privacy.	**4**	2
2	**Application development framework** Accessing the clouds: Web application vs Cloud Application, Frameworks: Model View Controller (MVC), Struts, Spring. Cloud platforms in Industry – Google AppEngine, Microsoft Azure, Openshift, CloudFoundry	**6**	6
3	**Cloud service delivery environment and API** Storing objects in the Cloud, Session management, Working with third party APIs: Overview of interconnectivity in Cloud ecosystems. Facebook API, Twitter API, Google API.	**5**	9
4	**Architecting for the Cloud : Best practices** Best practices in architecture cloud applications in AWS cloud, Amazon Simple Queue Service (SQS), RabbitMQ	**3**	2
5	**Cloud applications** Amazon Simple Notification Service (Amazon SNS), multi-player online game hosting on cloud resources, Building content delivery networks using clouds	**3**	2
6	**Managing the data in cloud** Securing data in the cloud, ACL, OAuth, OpenID, XACML, securing data for transport in the cloud, scalability of applications and cloud services.	**4**	2
7	**Automation and provisioning tool** Puppet and Chef – steps for automation: Introduction, files and packages, services and subscriptions, exec and notify, facts, conditional statements and logging.	**3**	2,6
8	**Recent Trends**	**2**	2
		30	

List of Experiments **Software / API / Tools** JDK 1.7/1.8, Eclipse IDE, Dropbox API, Apache tomcat server 7.0/8.0, Google AppEngine API, Servlets, Struts, Spring framework. 1) Design and Development of Web applications using MVC Framework. 2) Installing and Configuring required platform for Google App Engine 3) Studying the feature of GAE PaaS model. 4) Creating and running Web applications (Guest book, MVC) on local host and deploying the same in Google App Engine 5) Design and Development of Web applications using Struts. 6) Design and Development of Web applications using Spring framework. 7) Developing an ASP.NET based web application on Azure platform 8) Creating an application in Dropbox to store data securely. Develop a source code using Dropbox API for updating and retrieving files. 9) Installing Cloud Foundry in a local host and exploring CF commands. 10) Cloud application development using IBM Bluemix Cloud. 11) Installing and Configuring Dockers in local host and running multiple images on a Docker Platform. 12) Configuring and deploying VMs/Dockers using Chef/Puppet Automation tool.	**30**	14, 17

Reference Books

1. Rajkumar buyya, Christian vecchiola, S Thamarai Selvi , "Mastering cloud computing", Tata McGraw Hill Education Private Limited, 2013
2. Anthony T .Velte, Toby J. Velte, Robert Elsenpeter, "Cloud Computing a Practical Approach", Tata McGraw-HILL, 2010 Edition.
3. Barrie sosinsky, "Cloud computing bible, Wiley publishing
4. James Loope, "Managing Infrastructure with puppet", O'REILLY , June 2011
5. https://cloud.google.com/appengine/docs
6. https://www.chef.io/solutions/cloud-management/
7. https://aws.amazon.com/documentation

8. https://dev.twitter.com/overview/documentation
9. https://developers.facebook.com/
10. https://www.cloudfoundry.org/
11. https://puppet.com/blog/implement-a-message-queue-your-cloud-application

Knowledge Area	Total hours of coverage
CS:SF (System fundamental)	25
CS:PD (Parallel and distributed)	5
Total	30

What is covered in the course?
[A short description, and/or a concise list of topics - possibly from your course syllabus.(This is likely to be your longest answer)]

Cloud application development and management deals with the concepts that are essential to build and deploy various applications in various cloud platforms.

Module 1 - Basic concepts & techniques

This module introduces the fundamental elements of cloud computing and the open challenges in the cloud computing environment

Module 2 – Application Development framework

This module provides various ways to access the cloud services, discusses about the MVC framework and steps to deploy cloud applications in the various cloud platforms like Amazon Web Services, Google AppEngine, Microsoft Azure, Openshift and CloudFoundry.

Module 3 – Cloud Service Delivery Environment and API's

This module explains about managing sessions in a cloud and projects the working of various API's

Module 4 - Architecting for the Cloud : Best practices

This module explains the best practices in architecture cloud applications

Module 5 - Cloud applications

This module deals with the application-oriented concepts of cloud computing such as multi-player online game hosting on cloud resources, Building content delivery networks using clouds and Resource cloud mashups

Module 6 - Managing the data in cloud

Cloud management provides capabilities for managing faults, configuration, accounting, performance, and security. This module provides complete details about the essential things

which are needed to manage the data in cloud.

Module 7 - Automation and provisioning tool

This module explains puppet tool. Puppet is a configuration management framework with an object-oriented twist. It provides declarative language syntax and an abstraction layer that allow you to write heavily reusable and understandable configuration definitions.

Module 8 – Emerging Trends

What is the format of the course?

[*Is it face to face, online or blended? How many contact hours? Does it have lectures, lab sessions, discussion classes?*]

This Course is designed with 100 minutes of in-classroom sessions per week, 60 minutes of video/reading instructional material per week, 100 minutes of lab hours per week, as well as 200 minutes of non-contact time spent on implementing course related project. Generally this course should have the combination of lectures, in-class discussion, case studies, guest-lectures, mandatory of pre-class reading material, quizzes.

How are students assessed?

[*What type, and number, of assignments are students are expected to do? (papers, problem sets, programming projects, etc.). How long do you expect students to spend on completing assessed work?]*

- Students are assessed on a combination group activities, classroom discussion, and continuous, final assessment tests.

- Additional weight-age will be given based on their rank in crowd sourced projects/ Kaggle like competitions and certificate of completion of a related MOOC course.

Additional topics

[List notable topics covered in the course that you do not find in the CS2013 Body of knowledge]

Other comments

[optional]

Session wise Plan

Students Outcome coverage : 2,6,7,11.

Class Hour	Lab Hour	Topics Coverage	Levels of Mastery	Reference Book	Remarks
2	-	**Basic concepts & techniques** Business case for implementing cloud, Requirements collection for cloud application development, Cloud service models and deployment models	Familiar	1	
2	-	Cloud open challenges: Cloud inter-operability and standards, scalability and fault tolerance, security, trust and privacy	Familiar	1	
3	-	**Cloud Application Development** Accessing the clouds: Platforms, Web applications, web API's, Web browsers, Frameworks: Model View Controller (MVC), Struts, Spring. Cloud platforms in Industry	Usage	5	
2	-	Google AppEngine, Microsoft Azure, Openshift, CloudFoundry	Usage	5,7,10	
3	-	**Cloud service delivery environment and API** Storing objects in the Cloud: Session management, Working with third party APIs: Overview of interconnectivity in Cloud ecosystems.	Usage		
2	-	Facebook API, Twitter API, Flickr API, Google Maps API.	Usage	8,9,5	
3	-	**Online cloud applications** Best practices in architecting cloud applications in AWS cloud, Massively multiplayer online game hosting on cloud resources	Usage		
2	-	Building content delivery networks using clouds , Resource cloud mashups	Usage		

Class Hour	Lab Hour	Topics Coverage	Levels of Mastery	Reference Book	Remarks
3	-	**Managing the data in cloud** Declaring data types, securing data in the cloud: data location in the cloud, data control in the cloud,	Usage	11	
2	-	securing data for transport in the cloud, looking at data, scalability and cloud services, sorting out metadata	Usage	4	
3	-	**Automation and provisioning tool** Puppet – steps for automation: Introduction, files and packages, services and subscriptions, exec and notify, facts, conditional statements and logging.	Usage	3	
2	-	Puppeteering: Inheritance and parameterized classes, virtual resources, variables.	Usage	4	
2	-	Recent Trends		11	

Theory : 30 Hours (2 Credit hours /week)
Lab : 30 Hours (1 Credit hour / week)
Project : 60 Non-contact Hours (1 credit hours / week)
(15 Weeks schedule)

CSE6003	Web Services	L,T,P,J,C 2,0,2,0, 3
Objectives	• To provide a basic conceptual understanding of web enterprise architectures. • To explore distributed remote communication. • To make understand the basic concepts of Service Oriented Architecture. • To explore XML, web services, web service security and its implementation. • To understand micro services and enterprise application patterns.	
Expected Outcomes	After successfully completing the course the student should be able to • Design, Develop & Demonstrate web services (U) • Recognize meaning of web enterprise architecture and SOA. (A) • Describe the micro services and enterprise application patterns.(F)	
Student Learning	• Having a clear understanding of the subject related concepts and of contemporary issues (2) • Having an ability to design a component or a product applying all the relevant standards and with realistic constraints (6) • Having problem solving ability- solving social issues and engineering problems (9)	

Module	Topics	L Hrs	SLO
1	**WEB APPLICATION ARCHITECTURE** Web Architecture: MVC, middleware – Design considerations, Issues in web application design: Security issues and interoperability issues (WS-I).	**3**	2
2	**DISTRIBUTED REMOTE COMMUNICATION** RPC, Java RMI, message queuing, Data Serialization –MQTT, RabbitMQ, JMS – JSON – AVRO, Thrift, protocol buffer.	**5**	2
3	**SERVICE ORIENTED ARCHITECTURE** Introducing SOA – SOA triangle, layered architecture of SOA, BPO – Business Process Outsourcing - Web service composition and coordination.	**3**	2
4	**BUILDING SOA** Web service creation and accessing - WSDL, SOAP, UDDI, XINS, JSON-RPC, JSON-WSP, REST-ful web services,mashup	**6**	6,9

5	SEMANTIC WEB Services RDF, RDFS, OWL, SPARQL	**2**	6
6	**MICROSERVICES** Evolution, Modeling services, Integration, Deployment, Testing, Monitoring, Security. Implementation of microservices.	**5**	2
7	**ENTERPRISE APPLICATION PATTERNS** Concurrency patterns, Session state patterns. Web service security – protocols.	**4**	2,9
8	**Recent Trends**	**2**	
		30	
Lab - List of Experiments **Software / API / Tools / IDE** Netbeans Microsoft visual Studio 2010 1. Creation of .NET web service and consumed by .NET client (console, window and web).(2) 2. Creation of Java web service consumed by Java client.(2) 3. Interoperability in web services with java web service and .NET client.(2) 4. Interoperability in web services with .NET web service and java client.(2) 5. Creation of RESTful web services.(2) 6. Consuming a realtime web service.(2) 7. Creation and consuming .NET web service without IDE.(2) 8. Web service composition using BPEL.(4) 9. Web services with array methods.(2) 10. Web services with database connectivity methods. (2) 11. Application based on web service security(2) 12. Creation of ontology (4) 13. Application using SPARQL (2)		**30**	14, 17

Reference Books

1. J.D.Meier,Alex Homer,"Web Application Architecture guide, Patterns and Practices", Microsoft 2008.
2. Thomas Erl,"Service-Oriented Architecture: Concepts, Technology, and Design", Pearson Education, 2005.
3. Andrew S. Tenenbaum, Marteen Van Steen,"Distributed Systems, Principles and Paradigms", Second Edition, Pearson, Prentice Hall, 2007.
4. Sam Newman, " Building Micro Services ", O' Reilly, , 2015.

5. Martin Fowler, David Rice, Matthew Foemmel, Edward Hieatt, Robert Mee, Randy Stafford,"Patterns of Enterprise Application Architecture", Addison Wesley, 2002.
6. Heather Kreger," Web Services conceptual architecture", , IBM, 2001
7. Sacha Krakowiak," Middleware Architecture with Patterns and Frameworks", 2009
8. Leonard Richardson, Sam Ruby, "Restful Web Services", O'Reilly Media; First Edition edition (May 15, 2007)
9. Ben Smith," Beginning JSON", Apress, 2015
10. Mark O' Neill ,"Web services security" , McGraw Hill, 2003
11. Kapil Pant," Business Process Orchestration for SOA using BPMN and BPEL", Packt publishing, 2008
12. Gustavo Alonso, Fabio Casati, Harumi Kuno, Vijay Machiraju,"Web Services - Concepts, Architectures and Applications", Springer Verlag, 2004
13. Fensel, D., Facca, F.M., Simperl, E., Toma, I., "Semantic Web Services", Springer, 2011
14. Leon Shklar, Richard Rosen,"Web Application Architecture, Principles, Protocols and Practices", John Wiley and Sons, 2003.

Knowledge Areas that contain topics and learning outcomes covered in the course

Knowledge Area	Total Hours of Coverage
CS: SF(System fundamentals) computational paradigm	30

What is covered in the course?

[A short description, and/or a concise list of topics - possibly from your course syllabus.(This is likely to be your longest answer)]

This course deals with developing web based enterprise applications using web services. It deals with how Service Oriented Architecture evolved. A flavor of XML is given to understand web services. Micro services is also introduced. The patterns involved during enterprise application development is also discussed. Web service security and related protocols is also introduced.

Module 1 - WEB APPLICATION ARCHITECTURE

Web Architecture: MVC, middleware – Design considerations, Issues in web application design: Security issues and interoperability issues (WS-I).

Module 2 - DISTRIBUTED REMOTE COMMUNICATION

RPC, Java RMI, message queuing, Data Serialization –MQTT, RabbitMQ, JMS – JSON – AVRO, Thrift, protocol buffer.

Module 3 - SERVICE ORIENTED ARCHITECTURE

Introducing SOA – SOA triangle, layered architecture of SOA, BPO – Business Process Outsourcing - Web service composition and coordination.

Module 4 - BUILDING SOA

Web service protocol stack, WSDL, SOAP, UDDI, XINS, JSON-RPC, JSON-WSP, REST-ful web services.

Module 5 : SEMANTIC WEB
RDF, RDFS, OWL, SPARQL

Module 6 - MICROSERVICES
Evolution, Modeling services, Integration, Deployment, Testing, Monitoring, Security. Implementation of microservices.

Module 7 - ENTERPRISE APPLICATION PATTERNS
Concurrency patterns, Session state patterns. Web service security – protocols.

Module 8 - Recent Trends Mashup

What is the format of the course?
This Course is designed with 100 minutes of in-classroom sessions per week, 100 minutes of lab hours per week. Generally this course should have the combination of lectures, in-class discussion, case studies, guest-lectures, mandatory of pre-class reading material, quizzes.

How are students assessed?

- Students are assessed on a combination group activities, classroom discussion, and continuous, final assessment tests.

- Additional weightage will be given based on their rank in crowd sourced projects/ Kaggle like competitions and certificate of completion of a related MOOC course.

Session wise Plan

Students Outcome coverage: 2,4,6,9

Class Hour	Lab Hour	Topics Coverage	Levels of Mastery	Reference Book	Remarks
1	-	**WEB APPLICATION ARCHITECTURE** MVC, middleware	Usage	7	
1	-	Design considerations: Presentation layer, business layer, data layer, service layer	Usage	1	
1	-	Issues in web application design: Security issues and interoperability issues (WS-I).	Usage	1	
1	-	**DISTRIBUTED REMOTE COMMUNICATION** RPC, Java RMI, message queuing	Usage	3	
1	-	Java RMI, Message Queuing	Usage	3	
1	-	Data Serialization –MQTT, RabbitMQ, JMS	Usage	3	
2	-	JSON – AVRO, Thrift, protocol buffer.	Usage	9	
1	-	Introducing SOA – SOA triangle, layered architecture of SOA	Usage	2	
1	-	BPO – Business Process Outsourcing	Usage	11	
1	-	Web service composition and orchestration.	Usage	12	
1		**BUILDING SOA** WSDL	Usage	1,6	
1	-	SOAP	Usage	1,6	
1	-	UDDI, XINS	Familiarity	1,6	
1		JSON-RPC, JSON-WSP	Usage	9	
2		REST-ful web services.	Usage	8	
2	-	SEMANTIC WEB RDF, RDFS, OWL, SPARQL	Familiarity	13	

Class Hour	Lab Hour	Topics Coverage	Levels of Mastery	Reference Book	Remarks
2	-	**MICROSERVICES** Evolution, Modeling services, Integration	Familiarity	4	
2	-	Deployment, Testing, Monitoring, Security.	Familiarity	4	
1	-	Implementation	Familiarity	5	
1		**ENTERPRISE APPLICATION PATTERNS** Concurrency patterns	Usage	5	
1		Session state patterns	Usage	5	
2		Web service security – protocols.	Usage	10	
2		Recent Trends	Usage	13	
30 Hours	30 Hours				

Multicore Architectures

CSE5006	Multicore Architectures	L,T,P,J,C 2,0,2,0,3
Preamble	This course is to provide knowledge on multicore architectures that lays the foundation for the development of High Performance Applications through OpenMP, CUDA parallel programming platforms. It enables to analyse the performance of HPC applications using various profiling tools	
Objective of the course	• To provide knowledge on basics of Multicore architectures • To understand concepts of parallel computers and its programming models • To design and develop parallel programs • To practice parallel programming using OpenMP, CUDA parallel programming platforms • To apply program optimizations on parallel programs • To analyse the performance using profiling tools • To explore various contemporary tools and recent trends in field of multicore architectures	
Expected Outcome	After successfully completing the course the student should be able to 1) Describe various parallel programming models 2) Design and develop High Performance Applications using contemporary tools 3) Improve performance of applications through program optimizations 4) Analyse performance of parallel applications	
Student Learning Outcome	2. Having a clear understanding of the subject related concepts and of contemporary issues 11. Having interest in lifelong learning 14. Having an ability to design and conduct experiments, as well as to analyze and interpret data	

Module	Topics	L Hrs	SLO
1	**Introduction to Multi-Core Architectures** Evolution of multicores through Moor's Law, Comparisons of single core, multi-core, multi-processing and hyper threading	2	2
2	**Parallel Computers and programming** Threading Concepts, Communication Architectures and Communication Costs, Thread Level Parallelism(TLP), Instruction Level Parallelism(ILP), Comparisons, Cache Hierarchy and Memory-level Parallelism, Cache Coherence, Parallel programming models, Shared Memory and Message Passing, Vectorization	5	2
3	**OpenMP programming (Open multi-processing)** Introduction to OpenMP, Parallel constructs, Runtime Library routines, Work-sharing constructs, Scheduling clauses, Data environment clauses, atomic, master Nowait Clause, Barrier Construct	5	2
4	**CUDA Programming(Compute Unified Device Architecture)** 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	6	2
5	**Performance Analysers** Trace analyzer and collector (ITAC), VTune Amplifier XE, Energy Efficient Performance, Integrated Performance Primitives (IPP)	4	14
6	**Contemporary tools** MKL (Math Kernel Library), Threading Building Blocks, CUDA Tools	3	14
7	**HTC and MTC** HTC (High Throughput Computing), MTC (Many Task Computing), Top 500 Super computers in the world, Top 10 Super Computer architectural details, Exploring Linpack.	3	14
8	**Recent Trends**	2	11
		30	
Project		**60** [Non Contact hrs]	17

Projects may be given as group projects Design and development of High Performance applications through parallel programming platforms in the following areas Network Security Data Compression Image Processing Bio-Medical Information retrieval Natural Language Processing Health care Applications		

Reference Books

1. Rob Farber, "CUDA Application Design and Development", Morgan Kaufmann Publishers, 2013
2. Shameem Akhter and Jason Roberts, "Multi-Core Programming", 1st edition, Intel Press, 2012
3. Cameron Hughes, Tracey Hughes, "Professional Multicore Programming Design and Implementation for C++ Developers", Wiley, 2008
4. Robert Oshana, "Multicore Software Development Techniques: Applications, Tips, and Tricks", Newnes,1 edition, 2015
5. David B. Kirk , Wen-mei W. Hwu, "Programming Massively Parallel Processors: A Hands-on Approach (Applications of GPU Computing Series)", 1st edition, Morgan Kaufmann, 2010.

Knowledge areas that contain topics and learning outcomes covered in the course

Knowledge Area	Total Hours of Coverage
Systems Fundamentals (SF)	30

Body of Knowledge coverage

[List the Knowledge Units covered in whole or in part in the course. If in part, please indicate which topics and/or learning outcomes are covered. For those not covered, you might want to indicate whether they are covered in another course or not covered in your curriculum at all. This section will likely be the most time-consuming to complete, but is the most valuable for educators planning to adopt the CS2013 guidelines.]

KA	Knowledge Unit	Topics Covered	Hours
SF	Computer Organization	Evolution of multicores through Moor's Law, Comparisons of single core, multi-core, multi-processing and hyper threading.	2

SF	Parallelism	Threading Concepts, Communication Architectures and Communication Costs, TLP, ILP, Comparisons, Cache Hierarchy and Memory-level Parallelism, Cache Coherence, Parallel programming models, Shared Memory and Message Passing, Vectorization	5
SF	Parallel Programming Language	Introduction to OpenMP, Parallel constructs, Runtime Library routines, Work-sharing constructs, Scheduling clauses, Data environment clauses, atomic, master Nowait Clause, Barrier Construct	5
	Heterogeneous architecture and its programming	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	6
SF	Program Analyzer	Trace analyzer and collector (ITAC), VTune Amplifier XE, Energy Efficient Performance, Integrated Performance Primitives (IPP)	4
SF	Contemporary Tools	MKL (Math Kernel Library), Threading Building Blocks, CUDA Tools	3
SF	HTC and MTC	HTC (High Throughput Computing), MTC (Many Task Computing), Top 500 Super computers in the world, Top 10 Super Computer architectural details, Exploring Linpack.	3
SF	Recent Trends	HTC (High Throughput Computing), MTC (Many Task Computing), Top 500 Super computers in the world, Top 10 Super Computer architectural details, Exploring Linpack.	2
		Total hours	**30**

Where does the course fit in the curriculum?
[*In what year do students commonly take the course? Is it compulsory? Does it have pre-requisites, required following courses? How many students take it?*]

This course is a

- An elective Course.
- Suitable from first semester onwards.

What is covered in the course?
[A short description, and/or a concise list of topics - possibly from your course syllabus.(This is likely to be your longest answer)]

Module 1: Introduction to Multi-Core architecture

Evolution of multi-cores through Moor's Law, Comparisons of single core, multi-core, multi-processing and hyper threading.

Module 2: Parallel Computers & its programming

Threading Concepts, Communication Architectures and Communication Costs, TLP, ILP, Comparisons, Cache Hierarchy and Memory-level Parallelism, Cache Coherence, Parallel programming models, Shared Memory and Message Passing, Vectorization

Module 3: OpenMP Programming

Introduction to OpenMP, Parallel constructs, Runtime Library routines, Work-sharing constructs, Scheduling clauses, Data environment clauses, atomic, master Nowait Clause, Barrier Construct

Module 4: CUDA Programming

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

Module 5: **Performance Analysers**

Trace analyzer and collector (ITAC), VTune Amplifier XE, Energy Efficient Performance, Integrated Performance Primitives (IPP)

Module 6: Contemporary Tools

MKL (Math Kernel Library), Threading Building Blocks, CUDA Tools

Module 7: HTC and MTC

HTC (High Throughput Computing), MTC (Many Task Computing), Top 500 Super computers in the world, Top 10 Super Computer architectural details, Exploring Linpack.

Module 8: Recent trends

What is the format of the course?

[*Is it face to face, online or blended? How many contact hours? Does it have lectures, lab sessions, discussion classes?*]

This Course is designed with 100 minutes of in-classroom sessions per week. Generally this course should have the combination of lectures, in-class discussion, case studies, guest-lectures, mandatory off-class reading material, assignments and quizzes.

How are students assessed?

[*What type, and number, of assignments are students are expected to do? (papers, problem sets, programming projects, etc.). How long do you expect students to spend on completing assessed work?]*

- Students are assessed based on group activities, classroom discussion, assignments, quiz,

projects, continuous (CAT) assessment test, and final assessment test.

Additional topics

[List notable topics covered in the course that you do not find in the CS2013 Body of Knowledge]

Cuda Programming, Top 10 Super Computers in the world and Benchmarks

Other comments

Nil

Session wise plan

Student Outcomes Covered: 2, 5, 9, 17

Sl. No.	Class Hour	Topic Covered	levels of mastery	Reference Book	Remarks
1	2	Evolution of multi-cores through Moor's Law, Comparisons of single core, multi-core, multi-processing and hyper threading.	Usage	2	
2	5	Threading Concepts, Communication Architectures and Communication Costs, TLP, ILP, Comparisons, Cache Hierarchy and Memory-level Parallelism, Cache Coherence, Parallel programming models, Shared Memory and	Usage	2	

		Message Passing, Vectorization			
3	5	Introduction to OpenMP, Parallel constructs, Runtime Library routines, Work-sharing constructs, Scheduling clauses, Data environment clauses, atomic, master Nowait Clause, Barrier Construct	Usage	2	Assignments
4	6	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	Usage	1	Assignments
5	4	Trace analyzer and collector (ITAC), VTune Amplifier XE, Energy Efficient Performance,	Usage	1	

		Integrated Performance Primitives (IPP)			
6	3	MKL (Math Kernel Library), Threading Building Blocks, CUDA Tools	Usage	2	
7	2	HTC (High Throughput Computing), MTC (Many Task Computing), Top 500 Super computers in the world, Top 10 Super Computer architectural details, Exploring Linpack.	Familiarity		
8		Recent Trends			
	30 Hours (2 Credit hours /week ✍ 15 Weeks schedule)				

CSE6001	**Big Data Frameworks**	L,T,P,J,C 2,0,2,4,4
Objective	The course objective is to impart an understanding of the challenges in storing and processing big data and how to use different big data frameworks effectively to store and process big data.	
Expected Outcomes	After successfully completing the course the student should be able to a) Discuss the challenges in Big Data. b) Describe the need of different big data frameworks. c) Write MapReduce programming in both Hadoop and Spark Framework. d) Write programs in Spark Streaming, SPARK SQL and GraphX	
SLO's	2. Having a clear understanding of the subject related concepts and of contemporary issues 9. Having problem solving ability- solving social issues and engineering problems 14. Having an ability to design and conduct experiments, as well as to analyze and interpret data	

Module	Topics	LHrs	SLO
1	**INTRODUCTION TO BIG DATA** Data Storage and Analysis - Characteristics of Big Data – Big Data Analytics - Typical Analytical Architecture – Requirement for new analytical architecture – Challenges in Big Data Analytics – Need of big data frameworks	3	2
2	**Hadoop Framework** Hadoop – Requirement of Hadoop Framework - Design principle of Hadoop –Comparison with other system - Hadoop Components – Hadoop 1 vs Hadoop 2 – Hadoop Daemon's – HDFS Commands – Map Reduce Programming: I/O formats, Map side join, Reduce Side Join, Secondary sorting, Pipelining MapReduce jobs	6	2,9
3	**Hadoop Ecosystem** Introduction to Hadoop ecosystem technologies: Serialization: AVRO, Co-ordination: Zookeeper, Databases: HBase, Hive, Scripting language: Pig, Streaming: Flink, Storm	3	2,9

4	**Spark Framework** Overview of Spark – Hadoop vs Spark – Cluster Design – Cluster Management – performance,Application Programming interface (API): Spark Context, Resilient Distributed Datasets, Creating RDD, RDD Operations, Saving RDD - Lazy Operation – Spark Jobs	5	2,9

5	**Interactive Data Analysis with Spark Shell** Writing Spark Application - Spark Programming in Scala, Python, R, Java - Application Execution	3	2,9
6	**Spark SQL and GraphX** SQL Context – Importing and Saving data – Data frames – using SQL – GraphX overview – Creating Graph – Graph Algorithms	5	2,9, 14
7	**Spark Streaming** Overview – Errors and Recovery – Streaming Source – Streaming live data with spark	3	2,9
8	**Recent Trends**	2	2
Lab(IndicativeListofExperiments(inthe areasof) 1. HDFS Commends 2. MapReduce Program to show the need of Combiner 3. MapReduce I/O Formats –Text, key-value 4. MapReduce I/O Formats - NLine, Multiline 5. Sequence file Input / Output Formats 6. Secondary sorting 7. Distributed Cache & Map Side Join, Reduce side Join 8. Building and Running a Spark Application 9. Wordcount in Hadoop and Spark 10. Manipulating RDD 11. Inverted Indexing in Spark 12. Sequence alignment problem in Spark 13. Implementation of Matrix algorithms in Spark 14. Spark Sql programming 15. Building Spark Streaming application		**30**	14, 17

Project# Generally ateamproject[5to10 members] # Concepts studied inXXXX should have beenused # Down to earthapplication and innovative idea should have beenattempted # ReportinDigital formatwithall drawings usingsoftware package tobesubmitted. # Assessment ona continuous basis with a min of3 reviews. Projects may be given as group projects The following is the sample project that can be given to students to be implemented: 1. Predicting forest cover 2. Anomaly detection 3. Text Analytics 4. Co-occurrence of terms in social networks using GraphX 5. HITS algorithm 6. Geospatial and Temporal data analytics	**60**[Non Contact hrs]	17

ReferenceBooks

1. Mike Frampton, "Mastering Apache Spark", Packt Publishing, 2015.
2. Tom White, "Hadoop: The Definitive Guide", O'Reilly, 4th Edition, 2015.
3. Nick Pentreath, Machine Learning with Spark, Packt Publishing, 2015.
4. Mohammed Guller, Big Data Analytics with Spark, Apress, 2015
5. Donald Miner, Adam Shook, "MapReduce Design Pattern", O'Reilly, 2012

Big Data Frameworks

Knowledge Areas that contain topics and learning outcomes covered in the course

Knowledge Area	**Total Hours of Coverage**
CS: AL(Algorithms and Complexity) / CE: CAO	3
CS: PL(Programming Languages) / CE: CAO	24
CS: DS / CE: DSC	3

Body of Knowledge coverage

[List the Knowledge Units covered in whole or in part in the course. If in part, please indicate which topics and/or learning outcomes are covered. For those not covered, you might want to indicate whether they are covered in another course or not covered in your curriculum at all. This section will likely be the most time-consuming to complete, but is the most valuable for educators planning to adopt the CS2013 guidelines.]

KA	Knowledge Unit	Topics Covered	Hours
CE: AR	Memory System Organization and Architecture	Data Storage and Analysis - Characteristics of Big Data – Big Data Analytics - Typical Analytical Architecture – Requirement for new analytical architecture – Challenges in Big Data Analytics – Need of big data frameworks	3
CE: PD CS: PL / CE: PRF	Parallel algorithms, Analysis and Programming Language Pragmatics	Hadoop – Requirement of Hadoop Framework - Design principle of Hadoop –Comparison with other system - Hadoop Components – Hadoop 1 vs Hadoop 2 – Hadoop Daemon's – HDFS Commands – Map Reduce Programming: I/O formats SQL Context – Importing and Saving data – Data frames Writing Spark Application - Spark Programming in Scala, Python, R, Java - Application Execution	12
CS: PL / CE: PRF	Advanced Programming Constructs	Map side join, Reduce Side Join, Secondary sorting, Pipelining MapReduce jobs Overview – Errors and Recovery – Streaming Source – Streaming live data with spark Introduction to Hadoop ecosystem technologies: Serialization: AVRO, Co-ordination: Zookeeper, Databases: HBase, Hive, Scripting language: Pig, Streaming: Flink, Storm	12

CE/NC CS: DS / CE: DSC	Social networking Graphs and Trees	using SQL – GraphX overview – Creating Graph – Graph Algorithms	3
		Total Hours	30

What is covered in the course?
[A short description, and/or a concise list of topics - possibly from your course syllabus.(This is likely to be your longest answer)]

Part 1: Introduction to Big Data
This part of the course gives introduction to the basics of big data, characteristics of big data, challenges involved and the need for big data frameworks

Part II: Hadoop Framework
Describes the Hadoop Architecture and compares it with legacy distributed computing. This part of the course also introduces data storage in Hadoop and writing MapReduce code. The essential ecosystems of Hadoop are introduced in this part.

Part III: Spark and Streaming
This part of the course, introduces Spark tool, Graph algorithms and streaming. Spark will lead to interactive data analysis and supports streaming.

What is the format of the course?
[*Is it face to face, online or blended? How many contact hours? Does it have lectures, lab sessions, discussion classes?*]

This Course is designed with 100 minutes of in-classroom sessions per week, 100 minutes of lab hours per week, as well as 200 minutes of non-contact time spent on implementing course related project. Generally this course has the combination of lectures, in-class discussion, case studies, guest-lectures, mandatory off-class reading material, quizzes.

How are students assessed?
[*What type, and number, of assignments are students are expected to do? (papers, problem sets, programming projects, etc.). How long do you expect students to spend on completing assessed work?]*

- Students are assessed on a combination group activities, classroom discussion, projects, & continuous and final assessment tests.

- Additional weightage will be given based on their rank in crowd sourced projects.

- Students can earn additional weightage based on certificate of completion of a related MOOC course or any online course completion.

Session wise plan

S.No	Topic Covered	Class Hour	Levels of mastery	Reference Book
1	Data Storage and Analysis - Characteristics of Big Data – Big Data Analytics - Typical Analytical Architecture – Requirement for new analytical architecture – Challenges in Big Data Analytics – Need of big data frameworks	3	Familiarity	2
2	Hadoop – Requirement of Hadoop Framework - Design principle of Hadoop –Comparison with other system - Hadoop Components – Hadoop 1 vs Hadoop 2 – Hadoop Daemon's – HDFS Commands – Map Reduce Programming: I/O formats, Map side join, Reduce Side Join, Secondary sorting, Pipelining MapReduce jobs	6	Usage	2, 5
3	Introduction to Hadoop ecosystem technologies: Serialization: AVRO, Co-ordination: Zookeeper, Databases: HBase, Hive, Scripting language: Pig, Streaming: Flink, Storm	3	Familiarity	2, 3
4	Overview of Spark – Hadoop vs Spark – Cluster Design – Cluster Management – performance,Application Programming interface (API): Spark Context, Resilient Distributed Datasets, Creating RDD, RDD Operations, Saving RDD - Lazy Operation – Spark Jobs	5	Familiarity	1, 3, 4
5	Writing Spark Application - Spark Programming in Scala, Python, R, Java - Application Execution	3	Usage	1, 3, 4
6	SQL Context – Importing and Saving data – Data frames – using SQL – GraphX overview – Creating Graph – Graph Algorithms	5	Usage	1, 3, 4
7	Overview – Errors and Recovery – Streaming Source – Streaming live data with spark	3	Usage	1, 3, 4
8	**Recent Trends**	2		
Total hours		*30*		

CSE6005	Machine Learning	L, T, P, J, C 2,0,2,4,4
Objective	It introduces theoretical foundations, algorithms, methodologies, and applications of Machine Learning and also provide practical knowledge for handling and analysing data sets covering a variety of real-world applications.	
Expected Outcomes	After successfully completing the course the student should be able to 1. Recognize the characteristics of machine learning that make it useful to solve real-world problems. 2. Identify real-world applications of machine learning. 3. Identify and apply appropriate machine learning algorithms for analyzing the data for variety of problems. 4. Implement different machine learning algorithms for analyzing the data 5. Design test procedures in order to evaluate a model 6. Combine several models in order to gain better results 7. Make choices for a model for new machine learning tasks based on reasoned argument	
SLO's	7. Having computational thinking (Ability to translate vast data in to abstract concepts and to understand database reasoning) 14.Having an ability to design and conduct experiments, as well as to analyze and interpret data 17. Having an ability to use techniques, skills and modern engineering tools necessary for engineering practice	

Module	Topics	L Hrs	SLO
1	**INTRODUCTION TO MACHINE LEARNING** Introduction, Examples of Various Learning Paradigms, Perspectives and Issues, Version Spaces, Finite and Infinite Hypothesis Spaces, PAC Learning, VC Dimension.	3	7
2	**Supervised Learning** Decision Trees: ID3, Classification and Regression Trees, Regression: Linear Regression, Multiple Linear Regression, Logistic Regression, Neural Networks: Introduction, Perceptron, Multilayer Perceptron, Support vector machines: Linear and Non-Linear, Kernel Functions, K-Nearest Neighbours	9	7

3	**Ensemble Learning** Model Combination Schemes, Voting, Error-Correcting Output Codes, Bagging: Random Forest Trees, Boosting: Adaboost, Stacking	3	7
4	**Unsupervised Learning** Introduction to clustering, Hierarchical: AGNES, DIANA, Partitional: K-means clustering, K-Mode Clustering, Expectation Maximization, Gaussian Mixture Models	5	7
5	**Probabilistic Learning** Bayesian Learning, Bayes Optimal Classifier, Naïve Bayes Classifier, Bayesian Belief Networks	3	7, 14
6	**Learning Association Rules** Mining Frequent Patterns - basic concepts -Apriori algorithm, FP-Growth algorithm, Association-based Decision Trees	3	7, 17
7	**Machine Learning in Practice** Design, Analysis and Evaluation of Machine Learning Experiments, Other Issues: Handling imbalanced data sets	2	7
8	**Recent Trends**	2	7
Lab (Indicative List of Experiments (in the areas of) 1. Implement Decision Tree learning 2. Implement Logistic Regression 3. Implement classification using Multilayer perceptron 4. Implement classification using SVM 5. Implement Adaboost 6. Implement Bagging using Random Forests 7. Implement K – means Clustering to Find Natural Patterns in Data 8. Implement Hierarchical clustering 9. Implement K – mode clustering 10. Implement Association Rule Mining using FP Growth 11. Classification based on association rules 12. Implement Gaussian Mixture Model Using the Expectation Maximization 13. Evaluating ML algorithm with balanced and unbalanced datasets 14. Comparison of Machine Learning algorithms 15. Implement k – nearest neighbour algorithm		**30**	14

Project# Generally a team project [5 to 10 members] # Concepts studied in XXXX should have been used # Down to earth application and innovative idea should have been attempted # Report in Digital format with all drawings using software package to be submitted. # Assessment on a continuous basis with a min of 3 reviews.	**60** [Non Contact hrs]	17
Projects may be given as group projects The following is the sample project that can be given to students to be implemented: 1. Solving Data Science problems from Kaggle website 2. Applying Machine Learning algorithms in the field of biometrics for reliable and robust identification of humans from their personal traits, mainly for security and authentication purposes 3. Applying Machine Learning for OCR, Video Analytics 4. Applying Machine Learning algorithms in the field of Natural Language Processing for document clustering and sentiment analysis 5. Applying Machine Learning for Fraud Detection, Customer segmentation etc. Note: Students can down load real time data sets for different Machine Learning Tasks from https://archive.ics.uci.edu/ml/datasets.html and http://sci2s.ugr.es/keel/datasets.php#sub1 and do the projects		

Reference Books

1. Ethem Alpaydin,"Introduction to Machine Learning", MIT Press, Prentice Hall of India, Third Edition 2014.
2. Mehryar Mohri, Afshin Rostamizadeh, Ameet Talwalkar "Foundations of Machine Learning", MIT Press, 2012.
3. Tom Mitchell, "Machine Learning", McGraw Hill, 3rd Edition,1997.
4. Charu C. Aggarwal, "Data Classification Algorithms and Applications", CRC Press, 2014.
5. Charu C. Aggarwal, "DATA CLUSTERING Algorithms and Applications", CRC Press, 2014.
6. Kevin P. Murphy "Machine Learning: A Probabilistic Perspective", The MIT Press, 2012
7. Jiawei Han and Micheline Kambers and Jian Pei, "Data Mining –Concepts and Techniques", 3rd edition, Morgan Kaufman Publications, 2012.

Machine Learning

Knowledge Areas that contain topics and learning outcomes covered in the course

Knowledge Area	**Total Hours of Coverage**
CS: IS(Intelligent System)	30

Body of Knowledge coverage
[List the Knowledge Units covered in whole or in part in the course. If in part, please indicate which topics and/or learning outcomes are covered. For those not covered, you might want to indicate whether they are covered in another course or not covered in your curriculum at all. This section will likely be the most time-consuming to complete, but is the most valuable for educators planning to adopt the CS2013 guidelines.]

KA	**Knowledge Unit**	**Topics Covered**	**Hours**
CS: IS	IS/Basic Machine Learning	**Introduction to Machine Learning**	3
CS: IS	IS/Advanced Machine Learning	Supervised Learning Ensemble Learning Unsupervised Learning Probabilistic Learning Learning Association Rules Machine Learning in Practice Recent Trends	27
		Total hours	**30**

What is covered in the course?
[A short description, and/or a concise list of topics - possibly from your course syllabus.(This is likely to be your longest answer)]

Part 1: Introduction to Machine Learning

Introduction, Examples of Various Learning Paradigms, Perspectives and Issues, Version Spaces, Finite and Infinite Hypothesis Spaces, PAC Learning, VC Dimension.

Part II: Supervised Learning

This chapter covers supervised learning algorithms for classification tasks. The algorithms covered are the following: Decision Trees: ID3, Classification and Regression Trees, Regression: Linear Regression, Multiple Linear Regression, Logistic Regression, Neural Networks:

Introduction, Perceptron, Multilayer Perceptron, Support vector machines: Linear and Non-Linear, Kernel Functions, K-Nearest Neighbours

Part III: Ensemble Learning

This chapter covers ensemble learning algorithms for classification tasks. Model Combination Schemes, Voting, Error-Correcting Output Codes, Bagging: Random Forest Trees, Boosting: Adaboost, Stacking

Part IV: Unsupervised Learning

This chapter covers unsupervised learning algorithms for clustering tasks. The algorithms covered are the following: Introduction to clustering, Hierarchical: *AGNES, DIANA*, Partitional: *K-means clustering, K-Mode Clustering*, Expectation Maximization, Gaussian Mixture Models

Part V: Probabilistic Learning

This chapter covers learning algorithms based on Bayesian theory.Bayesian Learning, Bayes Optimal Classifier, Naïve Bayes Classifier, Bayesian Belief Networks

Part VI: Learning Association Rules

This chapter covers learning association rules from the data. The algorithms covered are the following: Mining Frequent Patterns - basic concepts -Apriori algorithm, FP-Growthalgorithm, Association-based Decision Trees

Part VII: Machine Learning in Practice

This chapter covers necessary points to be taken when applying machine learning algorithms on the data. Also discuss about evaluation metrics and methods for comparison of Machine learning algorithms.

Part VIII: Recent Trends

What is the format of the course?

[*Is it face to face, online or blended? How many contact hours? Does it have lectures, lab sessions, discussion classes?*]

This Course is designed with 100 minutes of in-classroom sessions per week, 100 minutes of lab hours per week, as well as 200 minutes of non-contact time spent on implementing course related project. Generally, this course should have the combination of lectures, in-class discussion, case studies, guest-lectures, mandatory off-class reading material, quizzes.

How are students assessed?

[*What type, and number, of assignments are students are expected to do? (papers, problem sets, programming projects, etc.). How long do you expect students to spend on completing assessed work?]*

- Students are assessed on a combination group activity, classroom discussion, projects, and continuous, final assessment tests.

- Additional weightage will be given based on their rank in crowd sourced projects/ Kaggle like competitions.

- Students can earn additional weightage based on certificate of completion of a related MOOC course.

Additional topics
[List notable topics covered in the course that you do not find in the CS2013 Body of Knowledge]

Other comments
[optional]

Session wise plan

Student Outcomes Covered: 2, 5,7,9

Class Hour	Lab Hour	Topic Covered	levels of mastery	Reference Book	Remarks
1		Introduction, Examples of Various Learning Paradigms	Familiarity	1,2	
1		Perspectives and Issues	Familiarity	1, 2	
1		Version Spaces, Finite and Infinite Hypothesis Spaces, PAC Learning, VC Dimension	Familiarity	1,2	
2		Decision Trees: ID3, Classification and Regression Trees	Usage	1	
2		Regression: *Linear Regression, Multiple Linear Regression, Logistic Regression*	Usage	1	
1		Neural Networks: *Introduction, Perceptron*		3	
1		Multi-layer Perceptron	Usage	3	
1		Support vector Machines - Linear	Usage	1,4	
1		Support vector Machines – Non-Linear, kernel functions		1,4	

1		K-nearest neighbour	Usage	3	
1		Model Combination Schemes, Voting, Error-Correcting Output Codes, Stacking	Usage	1,4	
1		Bagging: Random Forest Trees	Usage	1,4	
1		Boosting: Adaboost	Usage	1,4	
2		Introduction to clustering, Hierarchical Clustering: *AGNES, DIANA*	Usage	5	
2		Partitional K-means clustering, K-mode Clustering	Usage	5	
1		Expectation Maximization, Gaussian Mixture Models	Usage	5	
2		Bayesian Learning, Bayes Optimal Classifier, Naïve Bayes Classifier	Usage	3	
1		Bayesian Belief Networks	usage	3	
1		Mining Frequent Patterns - basic concepts –Apriori algorithm	Usage	7	
1		FP-Growth algorithm	Usage	7	
1		Association-based Decision Trees	Usage	1,6	
1		Design, Analysis and Evaluation of Machine Learning Experiments	Usage	6	

1		Comparison of Machine Learning algorithms, Other Issues: Handling imbalanced data sets		6	
2		Recent Trends			

CSE6006	NoSQL Databases	L,T,P,J,C 2,0,2,4,4
Objective	This course will explore the origins of NoSQL databases and the characteristics that distinguish them from traditional relational database management systems. This covers the architectures and common features of the main types of NoSQL databases (key-value stores, document databases, column-family stores, graph databases) Finally, discuss the criteria that decision makers should consider when choosing between relational and non-relational databases and techniques for selecting the NoSQL database that best addresses specific use cases.	
Expected Outcomes	After successfully completing the course the student should be able to 1. Explain the detailed architecture, define objects, load data, query data and performance tune NoSQL databases 2. Define NoSQL, its characteristics, history and primary benefits using NoSQL Databases. 3. Define the major types of NoSQL databases including a primary use case and advantages/disadvantages of each type. 4. Analyze semi-structured data and choose an appropriate storage structure	
SLO's	2- Having a clear understanding of the subject related concepts and of contemporary issues 7- Having computational thinking (Ability to translate vast data in to abstract concepts and to understand database reasoning) 12- Having adaptive thinking and adaptability	

Module	Topics	L Hrs	SLO
1	**INTRODUCTION TO NOSQL CONCEPTS** Data base revolutions: First generation, second generation, third generation, Managing Transactions and Data Integrity, ACID and BASE for reliable database transactions, Speeding performance by strategic use of RAM, SSD, and disk, Achieving horizontal scalability with database sharding, Brewer's CAP theorem.	4	2
2	**NOSQL DATA ARCHITECTURE PATTERNS** NoSQL Data model: Aggregate Models- Document Data Model- Key-Value Data Model- Columnar Data Model, Graph Based Data Model – Graph Data Model, NoSQL system ways to handle big data problems, Moving Queries to data, not data to the query, hash rings to distribute the data on clusters, replication to scale reads, Database distributed queries to data nodes.	4	12

3	**KEY –VALUE DATA STORES** From array to key –value databases, Essential features of key – value Databases, Properties of keys, Characteristics of Values, Key-Value Database Data Modeling Terms, Key-Value Architecture and implementation Terms, Designing Structured Values, Limitations of Key-Value Databases, Design Patterns for Key-Value Databases, Case Study: Key-Value Databases for Mobile Application Configuration	5	7
4	**DOCUMENT ORIENTED DATABASE** Document, Collection, Naming, CRUD operation, querying, indexing, Replication, Sharding, Consistency Implementation: Distributed consistency, Eventual Consistency, Capped Collection, Case studies: document oriented database: MongoDB and/or Cassandra	5	7
5	**COLUMNAR DATA MODEL - I** Data warehousing schemas: Comparison of columnar and row-oriented storage, Column-store Architectures: C-Store and Vector-Wise, Column-store internals and, Inserts/updates/deletes, Indexing, Adaptive Indexing and Database Cracking.	3	7
6	**COLUMNAR DATA MODEL - II** Advanced techniques: Vectorized Processing, Compression, Write penalty, Operating Directly on Compressed Data Late Materialization Joins , Group-by, Aggregation and Arithmetic Operations, Case Studies	3	7
7	**DATA MODELING WITH GRAPH** Comparison of Relational and Graph Modeling, Property Graph Model Graph Analytics: Link analysis algorithm- Web as a graph, PageRank-Markov chain, page rank computation, Topic specific page rank (Page Ranking Computation techniques: iterative processing, Random walk distribution Querying Graphs: Introduction to Cypher, case study: Building a Graph Database Application- community detection	4	7
8	**Recent Trends**	2	2

Lab (Indicative List of Experiments (in the areas of)	**30**	**14**

1. Import the Hubway data into Neo4j and configure Neo4j. Then, answer the following questions using the Cypher Query Language:

    a) List top 10 stations with most outbound trips (Show station name and number of trips)
    b) List top 10 stations with most inbound trips (Show station name and number of trips)
    c) List top 5 routes with most trips (Show starting station name, ending station name and number of trips) (4) List the hour number (for example 13 means 1pm -2pm) and number of trips which start from the station "B.U. Central"
    d) List the hour number (for example 13 means 1pm -2pm) and number of trips which end at the station "B.U. Central"

2. The flight data can be found at http://stat-computing.org/dataexpo/2009/the-data.html . You need to download just one year and from there you can sample a subset of at least 10000 records. You can use the data from a full year if you want but we recommend using a smaller dataset for simplicity.

    Hint: If you need to unzip the data file, you can use the command: bzip2 –d datafile from a terminal. For example, for the 2008, you download the file and unzip it using: bzip2 -d 1987.csv.bz2. The airport data can be found at http://stat-computing.org/dataexpo/2009/supplemental-data.html .

    1) Download the flight dataset and airport dataset.
    (2) Clean the dataset (for example: remove columns you do not need, remove records with missing information, remove duplicate records and so on).
    (3) Give the header to csv files
    (4) Import the data into Neo4j.
    (5) Write the queries to answer following questions:
        (5.1) List top 10 airports with most outbound flights.
        (5.2) List top 10 airports with most inbound flights.
        (5.3) List top 5 routes with most flights in weekdays.
        (5.4) List top 5 routes with most flights in weekends.
        List the hour number (for example 13 means 1pm -2pm) and number of flights, which depart from a specific airport in your data (e.g., Boston Logan Airport).
        List the hour number (for example 13 means 1pm -2pm) and number of flights, which arrive at specific airport in your data (e.g., Boston Logan Airport).

    In your report, you should answer the following questions:
    (a) List the year of the flights that you downloaded and prepared for this assignment. You can get a sample set from one-year data. However, the number of flights cannot be smaller than 10k.
    (b) Describe how you clean the data (Which columns you remove and why? Which rows you remove and why?). Hint: You can clean your data by writing a small program in Java, Python, C, Matlab or any kind of programming language.
    (c) Describe the header you give to the csv files.
    (d) Write down the command for importing data.
    (e) Write and execute the queries from step (5) above.

3. Download a zip code dataset at http://media.mongodb.org/zips.json. Use mongoimport to import the zip code dataset into MongoDB.

After importing the data, answer the following questions by using aggregation pipelines:

(1) Find all the states that have a city called "BOSTON".

(2) Find all the states and cities whose names include the string "BOST".

(3) Each city has several zip codes. Find the city in each state with the most number of zip codes and rank those cities along with the states using the city populations.

(4) MongoDB can query on spatial information.

Assume we have a spatial position as [-72, 42], and in the range of 2 (it can be [-71.5, 41.5] or [-72.5, 42.5] or somewhere else), there may exist a number of zip codes . Try to find the states in that range. You should return the total populations and the number of cities of each state in that range. Rank the states based on the number of cities.

(5) Consider a certain rectangular area, in which the vertices are [-80 , 30] , [-90 , 30] , [-90 , 40] and [-80 , 40]. Find and report the top 10 largest cities (by population) in this area.

4. Create a database that stores road cars. Cars have a manufacturer, a type. Each car has a maximum performance and a maximum torque value. Use ifconfig to determine a machine's IP address. To check if Cassandra is running in the background, run: **ps aux | grep cassandr[a]**

Do the following:

5. Test Cassandra's replication schema and consistency models.

6. Network Partition without Replication

7. Network Partition with Replication and Weak Consistency

8. Network Partition with Replication and Quorum Consistency

9. Cars have different powertrains. Each type can be described with different parameters:

10. Internal combustion engine: fuel type, displacement, maximum torque, maximum power

11. Electric motor: maximum torque, maximum power

12. Both: all of the above and the combined maximum torque and power values

13. The class hierarchy for different powertrain types

14. Extend the cars column family to store the powertrain of each car.

15. Write a query that collects the cars with an internal combustion engine.

16. Write a query that collects the cars with an internal combustion engine or an electric motor.

Project Projects may be given as group projects The following is the sample project that can be given to students to be implemented: 1. Analyzing and Visualizing social networks like Facebook and twitter using NoSQL Databases. 2. Using Sample datasets from http://www.rdatamining.com/resources/data, UCLA Repository, kaggle dataset etc., and analyzing those using NoSQL databases. 3. Twitter provides a fire hose of data. Automatically filtering, aggregating, analyzing such data can allow a way to harness the full value of the data, extracting valuable information. The idea of this project is investigating stream processing technology to operate on social streams. 4. Project on Combining Database management and Cloud storage system. 5. CarTel. In the CarTel project, we are building a system for collecting and managing data from automobiles. There are several possible CarTel related projects: a) One of the features of CarTel is a GUI for browsing geo - spatial data collected from cars. We currently have a primitive interface for retrieving parts of the data that are of interest, but developing a more sophisticated interface or query language for browsing and exploring this data would make a great project. b) One of the dangers with building a system like CarTel is that it collects relatively sensitive personal information about user's location and driving habits. Protecting this information from casual browsers, insurance companies, or other undesired users is important. However, it is also important to be able to combine different user's data together to do things like intelligent route planning or vehicle anomaly detection. The goal of this project would be to find a way to securely perform certain types of aggregate queries over CarTel data without exposing personally identifiable information. **Reference Books** 1. Guy Harrison, "Next Generation database: NoSQL New SQL and Big Data", Apress, I[st] Edition, 2015 2. Daniel G. McCreary and Ann M. Kelly "Making Sense of NoSQL" Manning publisher, Edition illustrated, 2013 3. Shanshak Tiwari, "Professional NoSQL", Wrox, I[st] Edition, 2011 4. Christopher D. manning, Prabhakar Raghavan, Hinrich Schutze, "An introduction to Information Retrieval", Cambridge University Press, 2008 5. Daniel Abadi, Peter Boncz, Stavros Harizopoulos, "The Design and Implementation of Modern Column-Oriented Database Systems", Now Publisher, 2013. 6. Kristina Chodorow, "Mongo DB the Definitive Guide" O'Reilly Media, 2013.	60	17

2. Knowledge Areas that contain topics and learning outcomes covered in the course

Knowledge Area	Theory Hours
CS: IM (Distributed Databases)	30
Total	30

2.1 Body of Knowledge coverage

KA	Knowledge Unit	Topics Covered	Hours
CS: IM	Distributed Databases	**Introduction to Nosql Concepts** **Nosql Data Architecture Patterns** **Key –Value Data Stores** **Document Oriented Database** **Columnar Data Model - I** **Columnar Data Model - II** **Data Modeling With Graph** **Recent Trends**	30
		Total hours	30

What is covered in the course?

Module I - INTRODUCTION TO NOSQL CONCEPTS

Data base revolutions: First generation, second generation, third generation, Managing Transactions and Data Integrity, ACID and BASE for reliable database transactions, Speeding performance by strategic use of RAM, SSD, and disk, Achieving horizontal scalability with database sharding, Brewer's CAP theorem.

Module II - NOSQL DATA ARCHITECTURE PATTERNS

NoSQL Data model: Aggregate Models- Document Data Model- Key-Value Data Model- Columnar Data Model, Graph Based Data Model – Graph Data Model, NoSQL system ways to handle big data problems, Moving Queries to data, not data to the query, hash rings to distribute the data on clusters, replication to scale reads, Database distributed queries to data nodes.

Module III - KEY –VALUE DATA STORES

From array to key –value databases, Essential features of key – value Databases, Properties of keys, Characteristics of Values, Key-Value Database Data Modeling Terms, Key-Value Architecture and implementation Terms, Designing Structured Values, Limitations of Key-Value Databases, Design Patterns for Key-Value Databases, Case Study: Key-Value Databases for Mobile Application Configuration

Module IV – DOCUMENT ORIENTED DATABASE

Document, Collection, Naming, CRUD operation, querying, indexing, Replication, Sharding, Consistency Implementation: Distributed consistency, Eventual Consistency, Capped Collection, Case studies: document oriented database: MongoDB and/or Cassandra

Module V - COLUMNAR DATA MODEL-I

Data warehousing schemas: Comparison of columnar and row-oriented storage, Column-store Architectures : C-Store and Vector-Wise, Column-store internals and Inserts/updates/deletes, Indexing, Adaptive Indexing and Database Cracking.

Module VI - COLUMNAR DATA MODEL-II

Advanced techniques: Vectorized Processing, Compression,Write penalty, Operating Directly on Compressed Data Late Materialization Joins , Group-by, Aggregation and Arithmetic Operations, Inserts/updates/deletes, Indexing, Adaptive Indexing and Database Cracking, Case Studies

Module VII - DATA MODELING WITH GRAPH

Comparison of Relational and Graph Modeling, Property Graph Model

Graph Analytics: Link analysis algorithm- Web as a graph, PageRank- Markov chain, page rank computation, Topic specific page rank (Page Ranking Computation techniques: iterative processing, Random walk distribution Querying Graphs: Introduction to Cypher, case study: Building a Graph Database Application- community detection

Module VIII – Recent Trends

What is the format of the course?

This Course is designed with 100 minutes of in-classroom sessions per week, 100 minutes of lab hours per week, as well as 200 minutes of non-contact time spent on implementing course related project. Generally this course should have the combination of lectures, in-class discussion, case studies, guest-lectures, mandatory off-class reading material, and assignment.

How are students assessed?

- Students are assessed on a combination of group activities, classroom discussion, projects and continuous, final assessment tests.
- Additional weightage will be given to students working with projects based on different databases, and competitions and projects handling with large databases.
- Students can earn additional weightage based on certificate of completion of a related MOOC course.

Session wise plan

Class Hour	Lab Hour	Topics Covered	Level of mastery	Text/Reference Book	Remarks
2		Data base revolutions : First generation,second generation, third generation,Managing Transactions and Data Integrity ACID and BASE for reliable database transactions	Familiarity	1,2	
2	2	Speeding performance by strategic use of RAM, SSD, and disk Achieving horizontal scalability with database sharding, Brewer's CAP theorem	Familiarity	1,2	
2		NoSQL Data model: Aggregate Models, Document Data model, Key-value Data model, Columnar Data model, Graph Based Data model Graph data model	Familiarity	1,2	
2	2	Ways that NoSQL systems handle big data problems Moving queries to the data, not data to the queries Using hash rings to evenly distribute data on a cluster Using replication to scale reads Letting the database distribute queries evenly to data nodes	Usage, Assessment	1,2	
2		Key-value data stores From Arrays to Key-Value Databases Essential Features of Key-Value Databases Properties of Keys Characteristics of Values Key-Value Database Data Modeling Terms Key-Value Architecture Terms	Familiarity,	1,2	
3	4	Key-Value Implementation Terms Key Design and Partitioning Designing Structured Values Limitations of Key-Value Databases Design Patterns for Key-Value Databases Case Study: Key-Value Databases	Usage, Assessment	1,2	

		for Mobile Application Configuration			
3	4	Document Oriented Database Document, Collection, Naming, CRUD operation, querying, indexing, Replication, Sharding, Consistency Implementation: Distributed consistency, Eventual Consistency	Usage, Assessment	1,2	
2	4	Advanced Topics: Capped Collection, Case studies : document oriented database: MongoDB and/or Cassandra	Usage, Assessment	1,2	
2		Data warehousing schemas: Comparison of columnar and row-oriented storage, Column-store Architectures : C-Store and Vector-Wise	Familiarity	1,2	
4	6	Column-store internals and advanced techniques Vectorized Processing Compression ,Write penalty Operating Directly on Compressed Data Late Materialization Joins , Group-by, Aggregation and Arithmetic Operations , Inserts/updates/deletes Indexing, Adaptive Indexing and Database Cracking	Usage, Assessment	1,2	
2	4	Data modeling with Graph : Comparison of Relational and Graph Modeling,Property Graph Model,Graph Analytics: Link analysis algorithm- Web as a graph, PageRank- Markov chain, page rank computation, Topic specific page rank.	Usage, Assessment	1,2	
2	4	Building a Graph Database Application- community detection Querying Graphs: An Introduction to Cypher	Usage, Assessment	1,2	

2		Recent trends	Familiarly		
30 Hours/2 hours per week	30 Hours/2 Hours per week				

CSE6026	IoT on Cloud	L,T,P,J,C 2,0,2,4,4
Preamble	This course is to understand the working of Internet of Things. It introduces students to technology that enables IoT, application of IoT, cloud support for IoT and access data using mobile computing devices. This will serve as foundation for the cyber physical systems, Internet of services leading to Industry 4.0 changes.	
Objective of the course	• Introduction to fundamentals of IoT • Application of IoT in various domain • Hardware and software that enable IoT • Upload data on cloud for further analysis and visualisation • Access the IoT data from cloud using mobile computing devices. • Learn to use of tools such as Apache servers, WebAPI, • Design product for automation various domain such as for Home, Industry.	
Expected Outcome	2 - Having a clear understanding of the subject related concepts and of contemporary issues 5- Having design thinking capability 9- Having problem solving ability- solving social issues and engineering problems	

Module	Topics	L Hrs	SLO
1	**Introduction to IoT** Things in IoT, IoT protocols, IoT communication model, IoT communication APIs, IoT enabling Technologies.	3	2
2	**Application of IoT** Home, Cities, Environment, Energy, Retail, Logistics, Agriculture, Industry, Health, Life style, M2M – Machine to Machine, Difference between IoT and M2M. Industry 4.0 concepts - cyber physical system, Security aspects in IoT	4	5,9

3	**IOT Supported hardware** Introduction to wireless sensor network, RFID, Sensors, Overview of IoT supported Hardware platforms (Any two hardware can be handled) Raspberry pi, Arduino and Intel Galileo boards, Beaglebone, ARM Cortex Processors.	5	5,9
4	**Communication in IOT** Interface protocol , Serial, SPI, I2C, 6LoWPAN, 802.11wifi, 802.15 Bluetooth, 802.15.4 ZigBee, RTLS, GPS, CoAP – Constrained application protocol, RPL – routing protocol for lossy networks - MQTT.	7	5,9
5	**IOT Physical Servers and cloud** Data retrieval from cloud – data storage to cloud – data dissemination from sensors of IoT – Cloud Access to Sensors with IBM Bluemix, Juju Framework, Google Cloud Engine – data visualization	3	2,5
6	**IoT on Cloud** DevOps – Contained based software - Designing a RESTful Web API – PubNub API for IoT to cloud – mobile device as IoT – Mobile cloud access	3	2,9
7	**IoT Operating Systems** Introduction to Contiki OS and RIOT OS – Proto threads – tasking – Simple IoT application with border router, implementation of CoAP in Contiki OS.	3	2,9
8	**Recent Trends**	2	2
Lab. Experiments (in the areas of) 1. Installation of Raspbian OS or Ubuntu ARM OS on a Rasberry Pi Platform 2. Setting the networking parameters for Raspbian OS like Ethernet, WLAN, Bluetooth, etc 3. Enabling Security or SELinux in Raspbian OS or Ubuntu OS 4. Accessing IBM Bluemix from IoT Devices 5. Data dissemination from Sensor nodes (any make) 6. Data visualization using d3.js or any other tool 7. Contiki OS Installation and Simple IoT network configuration using Contiki 8. Border Router using Contiki OS 9. Implementation of CoAP protocol using Contiki OS 10. Energy, power, duty cycle calculation of IoT devices in Contiki OS 11. Simple application deployment in Google Cloud Engine or Juju Framework 12. Simple application deployment with PubNub cloud services.		**30**	14
Project # Generally a team project [5 to 10 members]		**60** [Non	17

# Concepts studied in XXXX should have been used # Down to earth application and innovative idea should have been attempted # Report in Digital format with all drawings using software package to be submitted. [Ex. 1. Design of a traffic light system using sequential circuits OR 2. Design of digital clock] # Assessment on a continuous basis with a min of 3 reviews. **Sample project titles:** Smart grid Vehicle charging using IOT Energy measurement and storage at cloud Water measurement and storage at cloud Analysis and presentation IOT data stored at cloud Smart Parking Flash flood prediction Real time monitoring of flood Remote Monitoring & Sensing Remote Controlling, Performance Analysis IOT industries and what they are doing, selling Healthcare application	Contact hrs]	

Reference Books

1. Arshdeep Bahga, Vijay Madisetti, Internet of Things: A hands-on Approach, University Press, 2015 (1 st edition)
2. Adrian McEwen & Hakim Cassimally, Designing the Internet of Things by Nov 2013, (1st edition)
3. Claire Rowland, Elizabeth Goodman, Martin Charlier, Ann Light, Algred Lui Designing Connected Products: UX for the consumer internet of things, O'Reilly, (1 st edition).
4. Rethinking the Internet of things: A Scalable Approach to Connecting
5. Learning Internet of Things by Peter Waher, Packt Publishing, 2015
6. Designing the Internet of Things, by Adrian Mcewen, Hakin Cassimally , Wiley India Private Limited
7. Cloud Computing, Thomas Erl, Pearson Education, 2014
8. Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud, William Stallings, Addison-Wesley Professional; 1 edition

IoT on Cloud

Knowledge areas that contain topics and learning outcomes covered in the course

Knowledge Area	Total Hours of Coverage
CS:HCI(Foundation, New Interactive Technologies), / CE:HCI(History and Overview, Foundations of human-computer interaction, I/O technologies, Intelligent systems), CE:ESY Embedded Microcontrollers, Low-power computing)	14
CS:AR (Interfacing and Communication)/CE:CAO(Interfacing and communication, Device subsystems, Processor systems design)	7
CS:SF (Cross-Layer Communications)/ CE:CAO(Interfacing and communication, Device subsystems, Processor systems design)	3
CS:PBD(Web Platform)(Mobile Platforms)/ CE:HCI (Foundations of human-computer interaction), CE:NWK (Mobile Platforms Wireless and mobile computing)	6

Body of Knowledge coverage

[List the Knowledge Units covered in whole or in part in the course. If in part, please indicate which topics and/or learning outcomes are covered. For those not covered, you might want to indicate whether they are covered in another course or not covered in your curriculum at all. This section will likely be the most time-consuming to complete, but is the most valuable for educators planning to adopt the CS2013 guidelines.]

KA	Knowledge Unit	Topics Covered	Hours
CS:HCI	Foundations	• Home, Cities, Environment, Energy, Retail, Logistics, Agriculture, Industry, Health, Life	4

CE:HCI	History and Overview Foundations of human-computer interaction	style, M2M – Machine to Machine, Difference between IoT and M2M. Industry 4.0 concepts - cyber physical system, Security aspects in IoT	
CS:HCI CE:HCI CE:ESY	New Interactive Technologies I/O technologies Intelligent systems Embedded microcontrollers Low-power computing	• Things in IoT, IoT protocols, IoT communication model, IoT communication APIs, IoT enabling Technologies. • Introduction to wireless sensor network, RFID, sensors, Overview of IoT supported Hardware platforms (Any two hardware can be handled) Raspberry pi, Arduino and Intel Galileo boards, Beaglebone, ARM Cortex Processors. • Recent trends	10
CS:AR CE:CAO	Interfacing and Communication Interfacing and communication Device subsystems Processor systems design	• Interface protocol , Serial, SPI, I2C, 6LoWPAN, 802.11wifi, 802.15 Bluetooth, 802.15.4 Zigbee, RTLS, GPS, CoAp – Constrained application protocol,	7
CE:HCI CE:CAO	Cross-Layer Communications Device subsystems Processor systems design	• Data retrieval from cloud – data storage to cloud – data dissemination from sensors of IoT – Cloud Access to Sensors with IBM Bluemix, Juju Framework, Google Cloud Engine – data visualization	3
CS:PBD CE:HCI	Web Platform Foundations of human-computer interaction	DevOps – Contained based software - Designing a RESTful Web API – PubNub API for IoT to cloud – mobile device as IoT – Mobile cloud access	3
CE-OS	Operating Systems	• **IoT Operating Systems** Introduction to Contiki OS and RIOT OS – Proto threads – tasking – Simple IoT application with border router, implementation of CoAP in Contiki OS.	3
		Total hours	**30**

Where does the course fit in the curriculum?
[*In what year do students commonly take the course? Is it compulsory? Does it have pre-requisites, required following courses? How many students take it?*]

What is covered in the course?
[A short description, and/or a concise list of topics - possibly from your course syllabus.(This is likely to be your longest answer)]

What is the format of the course?
[*Is it face to face, online or blended? How many contact hours? Does it have lectures, lab sessions, discussion classes?*]

This Course is designed with 100 minutes of in-classroom sessions per week. Generally this course should have the combination of lectures, in-class discussion, case studies, guest-lectures, mandatory off-class reading material, assignments and quizzes.

How are students assessed?
[*What type, and number, of assignments are students are expected to do? (papers, problem sets, programming projects, etc.). How long do you expect students to spend on completing assessed work?]*

- Students are assessed based on group activities, classroom discussion, assignments, quiz, projects, Midterm assessment test, and final assessment test.

Additional topics
[List notable topics covered in the course that you do not find in the CS2013 Body of Knowledge]

Other comments

Session wise plan

Student Outcomes Covered: 2, 5, 9, 17

Sl. No.	Class Hour	Lab Hour	Topic Covered	levels of mastery	Reference Book	Remarks
1	1		Things in IoT, IoT protocols,	Familiarity	1	
2	2		IoT communication model, IoT communication APIs, IoT	Usage	1	

			enabling Technologies.			
3	2		Home, Cities, Environment, Energy, Retail, Logistics, Agriculture, Industry, Health, Life style,	Familiarity	1	
4	2		M2M – Machine to Machine, Difference between IoT and M2M. Industry 4.0 concepts - cyber physical system, Security aspects in IoT	Familiarity	1,2	LAB Component
5	1		Introduction to wireless sensor network, RFID,	Familiarity	1,2	
6	4	6	(Any two hardware can be handled) Raspberry pi, Arduino and Intel Galileo boards, Beaglebone, ARM Cortex Processors.	Usage	1, INTEL Galileo Arduino Lab manual	LAB Component
7	1	4	Interface protocol , Serial, SPI, I2C,	Usage	1	LAB Component
8	3	4	6LoWPAN, 802.11wifi, 802.15 Bluetooth, 802.15.4 Zigbee	Usage	1	LAB Component
9	3	4	RTLS, GPS, CoAp – Constrained application protocol, RPL – routing protocol for lossy networks.	Usage	1	LAB Component
10	1	2	Data retrieval from cloud – data storage to cloud – data dissemination from sensors of IoT	Usage	1	LAB Component
11	2	4	Cloud Access to Sensors with IBM Bluemix, Juju Framework, Google Cloud Engine	Usage	1	LAB Component
12	1	2	Data visualization	Usage	1	
13	2	4	**IoT on Cloud** DevOps – Contained based software - Designing a RESTful Web API – PubNub API for IoT to cloud – mobile device as IoT – Mobile cloud access	Usage	1	LAB Component
14	3		**IoT Operating Systems** Introduction to Contiki OS and RIOT OS – Proto threads – tasking – Simple IoT application	Usage	1	Lab Component

			with border router, implementation of CoAP in Contiki OS.			
15	2		**Recent Trends**	Familiarity		
	30 Hours	30 Hours				

CSE6027	Mobile Cloud Computing	L T P J C 2 0 2 4 4
Objectives	This Course provides a comprehensive overview of how to integrate cloud and mobile technology. It is an emerging field and this course explores how distributed resources can be shared by mobile users in very different ways and issues arising therefrom.	
Expected Outcome	After successfully completing the course the student should be able to 1. Describe the usefulness of integrating cloud and mobile technologies. 2. Sketch the architecture and discuss the characteristics of mobile cloud computing. 3. Examine the issues of security and privacy due to user cooperation in mobile cloud. 4. Develop, Design and Deploy mobile Cloud Based Applications	
Student learning Outcome (SLO	• Having a clear understanding of the subject related concepts and of contemporary issues (2) • Having design thinking capability (5) • Having an ability to design a component or a product applying all the relevant standards and with realistic constraints(6)	

Module	Topics	L Hrs	SLO
1	**Mobile clouds: introduction and background** Introduction-Mobile vs Desktop devices -App Store, Google Play, WindowsStore-Developmentenvironments-PhoneGAP- Native vs. web applications – Mobile Connectivity Evolution: From Single to Multiple Air Interface Devices –Mobile cloud architecture and its need	5	2
2.	**Sharing Device Resources and allocation in Mobile Clouds** Resource sharing - Loudspeakers, Microphones - Image Sensors, Displays, General Purpose Sensors, Keyboards, Data Pipes, Mobile Apps-Sharing Mass Memory-Sharing Processing Units, Sharing Batteries-Resource allocation strategy-Task scheduling, Middleware-Energy aware resource allocation	5	5,6

3	**Integrating technologies for mobile clouds** Wireless Communication Technologies - Cellular Communications Systems-Short-Combined Air Interface, User Cooperation in Mobile Clouds - Data Integrity in Mobile Cloud-warehousing and analysing social data using cloud-Social compute cloud	4	2,5
4.	**Mobile Cloud Formation and Maintenance** Mobile Cloud Stages- Service Discovery for Mobile Clouds-Social Networks and Mobile Clouds- Mobile Networks Supporting-Distributed Storage in Mobile Clouds Standards-Building mobile clouds	4	2,5
5.	**Application deployment on mobile clouds** Forced Cooperation – Overlay Network – Crowd sourced Information, Mobile Apps information backup- Direct Mobile Cloud-TechnicallyenabledCooperation–Bimetric application-vehicle monitoring application-Cooperative Access	4	2,6
6.	**Green and Social aspects of mobile clouds** Energy Consumption for the Sequential Local Exchange (SLE) - Energy Consumption for the Parallel Local Exchange (PLE) - Cooperative Streaming - Comparison of the Different Approaches SLE and PLE.	3	6
7.	**Security in Mobile Cloud** Security in mobile social cloud-Different level of security, Security issues. Trust in mobile computing:Trust properties, component of trust, type of trust, trust issues	3	5
8.	**Recent Trends**	2	2
Suggested projects to gain subject knowledge: Google Ara-(xamarin framework) 1.Vehicle Tracking Using Driver Mobile Gps Tracking 2.Android Employee Tracker 3.Develop a MIDlet that has a Text Field and Label GUI components.		**60** [Non Contact hrs]	14

Reference Books

1. Mobile Cloud Computing: Architectures, Algorithms and Applications- Debashis De CRC press 2015
2. Mobile Clouds Exploiting Distributed Resources in Wireless, Mobile and Social Networks-Frank H.P. Fitzek and Marcos D. Katz
3.Valentino Lee, Heather Schneider, and Robbie Schell, Mobile Applications: Architecture, Design, and Development , Prentice Hall , 2004
4.Brian Fling, Mobile Design and Development O'Reilly Media,2009 5.Maximiliano Firtman Programming the Mobile Web , O'Reilly Media, 2010. 6.Christian Crumlish and Erin Malone Designing Social Interfaces, O'Reilly Media , 2009

Experiments: SLO: 17

1) Using Eclipse Graphical Layout Builder to understand complete Lifecycle mobile Apps
2) Tapping a Button displays a message and Controlling width & height of UI Widgets
3) Combining Horizontal & Linear layouts and Capturing Text from an Edit Text Object
4) Display the encrypted text back in the Text Field (so pressing enter should give you back the original text) Display the length of the entered text using the Label.
5) Integrating Mobile to Cloud Environment & Sharing the resource into the cloud.
6) Processing on Cloud Computing infrastructure: Hadoop/MapReduce.
7) Scalable multimedia processing application with Amazon Web Services
8) Event Data Collection and Processing by Crowdsourcing to Smartphones
9) Collaborative Experiment using Mobile Cloud Computing
10) Massive Processing Computation using Mobile Cloud Computing
11) Mobile multimedia Cloud services on Android devices eg. Traffic Manament (or) Natural Disaster Alert System
12) Mobile Application Performance and Energy Consumption Aspects through Hardware and Emulation.

Mobile Cloud Computing

Knowledge Areas that contain topics and learning outcomes covered in the course

Knowledge Area	**Total Hours of Coverage**
CS:PDB(Platform Based Development)	10
CS:PL(Programming Languages)	20
CS:IAS(Information Assurance and Security)	5
CS-HCI (Human Conputer Interface)	5
CS- GV - Graphics and Visualization	10
CS-SE – software engineering	10

Body of Knowledge coverage

KA	Knowledge Unit	Topics Covered	Hours
CS: PDB	Development Environment	Platforms, Hierarchy, Architecture	5
CS: PDB	Frameworks	Configuration, Tools. Interface.	5
CS: PL	Applications	Programming, native level programming	20
CS: IAS	Risk management	Security, Threads	5
CS-HCI	Social information networking	SVN Repository & IDE. Simple GAE application	5
CS-GV	Visualization, Computer Animation	Mobile Apps information backup- Direct Mobile Cloud- Technically enabled Cooperation – Direct Mobile Cloud- CoopLoc - Cooperative Access	10
CS-SE	Tools and Environments, Software Reliability	Combined Air Interface – Inter flow Network Coding, User Cooperation in Mobile Clouds - Security, Privacy and Data Integrity in Mobile Cloud	10
		Total hours	60

Where does the course fit in the curriculum?

The course may be offered from 2nd semester onwards

What is covered in the course?

The course covers various aspects **Mobile Cloud Computing** for both web apps and native apps for Android using Eclipse and the Android SDK, to write native apps for iPhones, iPod Touches, and iPads using Xcode and the iOS SDK

What is the format of the course?

Face-to-face lectures, for 2 contact hours/week.

This Course is designed with 100 minutes of in-classroom and 100 min in lab sessions per week, as well as 200 minutes of non-contact time spent on implementing course related project. Generally this course should have the combination of lectures, in-class discussion, case studies, guest-lectures, mandatory off-class reading material, quizzes.

How are students assessed?

Written homework assignments, course projects, midterm exam, final exam.

- Students are assessed on a combination group activities, classroom discussion, projects, and continuous, final assessment tests.
- Students can earn additional weight-age based on certificate of completion of a related MOOC course (Coursera online certification).

CSE6028	Cloud Security and Audit	L,T,P,J,C 2,0,0,4,3
Objectives	The objective of the course is to impart 1. Understanding of the security challenges in the cloud environment. 2. Understanding of the issues regarding privacy and manage risks associated with it. 3. Knowledge of security standards and the audit processes to follow and ensure better cloud security.	
Expected Outcomes	After successfully completing the course the student should be able to understand • State the security challenges of cloud infrastructure. • Illustrate the application security mechanisms. • Use the standards to define a management policy. • Design solutions for risk management and security threats. • Prepare a cloud security audit report.	
Student Learning Outcomes	• Having a clear understanding of the subject related concepts and of contemporary issues (2) • Having an ability to design a component or a product applying all the relevant standards and with realistic constraints (6) • Having problem solving ability, solving social issues and engineering problems (9)	

Module	Topics	L Hrs	SLO
1	**Cloud Security Fundamentals** Cloud computing security challenges – cloud computing security architecture – data security life-cycle - Security Patterns and architectural elements - Planning key Strategies for secure operation	**2**	2
2	**Cloud Application Security** Encryption techniques – homomorphic encryption - securing data Redaction - secure bitcoin – Public key infrastructure (PKI) – key management - open web application security project (OWASP) Cloud Top 10 Security Risks - Security as a service (SECaaS).	**5**	6
3	**Cloud Infrastructure Security** Virtualization security – securing hypervisor - securing virtual machines - designing virtual network for security - Network Security in the cloud - software-defined security - secure isolation strategy - anti-fragile cloud infrastructure - Failure as a service.	**5**	9

4	**Security Management & Privacy** Managed Security Service Provider(MSSP): Availability management – configuration management - vulnerability management - identity management. - Privacy: privacy, compliance and the cloud - privacy enhancing encryption	**4**	9
5	**Risk Management & Security Threats** Risk management – principles - assessing the risk – strategies for managing risk – risk analysis framework – security threats - intrusion detection	**3**	6
6	**Cloud Standards and Compliance** Cloud security alliance – cloud controls matrix - cloud security standards guidance – security compliance - NIST – PCI data security standards – SAS 70 - ISO 27001 – HIPAA – ITIL - FISMA - FIPS 140-2.	**4**	6
7	**Audit** Cloud-Based IT Audit Process – System and Infrastructure lifecycle management for the cloud - goverance, risk management and compliance (GRC) – cloud audit assurance – auditing – record generation, reporting and management - tamper-proofing audit logs service level agreement (SLA) – legal safeguards - cloud morphing.	**5**	6
8	**Recent Trends**	**2**	2
	Projects 1. Altor VF – security appliances 2. Appgate security server – application layer firewall 3. Biobeans – secure identification and authentication solutions 4. Ciphercloud – data privacy 5. Defensepro – intrusion protection system(IPS) 6. Incapsula – DDoS protection, load balancing, fail-over services. 7. Cloudbric – malicious attack	**60** [Non Contact hrs]	17
	Total Hrs	**30**	

Reference Books

1. Tim Mather, Subra Kumaraswamy, Shahed Latif, **Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance**, O'Reilly Media Inc, 2009.
2. Dave Shackleford, **Virtualization security:Protecting virtualized environments**, John Wiley & sons, 2013.
3. Vic (J.R.) Winkler, **Securing the Cloud: Cloud Computer Security Techniques and Tactics,** Syngress; 1st edition (April 29, 2011).

4. Raghu Yeluri, Enrique castro-leon, **Building the infrastructure for cloud security:A Solutions view**, Apress, 2014.
5. Krutz, R.L. (2010), **Cloud Security A Comprehensive Guide to Secure Cloud Computing**, Wiley
6. Ben Halpert **, "Auditing Cloud Computing: A Security and Privacy Guide**: ", John Wiley & Sons, 2011.
7. Shao ying zhu, Richard Hill, **Guide to security assurance for cloud computing**, Springer 2015.
8. John Rittinghouse, James F.Ransome, **Cloud computing implementation, Management, security**, CRC Press, 2010.
9. Stefan Rass, Daniel Slamanig, **Cryptography for security and Privacy in cloud computing**, Artech House, 2014.
10. OWASP - https://www.owasp.org/images/4/47/Cloud-Top10-Security-Risks.pdf

Knowledge Areas that contain topics and learning outcomes covered in the course

Knowledge Area	**Total Hours of Coverage**
IAS: (Information Assurance and Security)	20
PD: (Parallel and Distributed Computing)	10
Total	30

What is covered in the course?

Module I: Cloud Security Fundamentals

This module discusses about the cloud security architecture challenges and its life cycle and provides the key strategies for secure operation in cloud security.

Module II: Cloud Application Security

This module describes about various encryption techniques, key management and top 10 security risks .

Module III: Cloud Infrastructure Security

This module discuss about virtualization and network security aspects. It also discuss about the current terms about anti-fragile and failure as a service.

Module IV: Security Management & Privacy

This module discuss about various managed security service providers, privacy in cloud security and compliance.

Module V: Risk Management & Security Threats

This module describes about various risk factors and the current security threats in the cloud environment.

Module VI: Cloud Standards and Compliance

This module discuss about various security alliance. It discuss about the standards and compliance in cloud security

Module VII: Audit

It explain about the various audit process involved in cloud security and to familiarise in the GRC and SLA's

Session wise plan

Student Outcomes Covered: 2, 6, 9, 10, 11

Class Hour	Topic Covered	Levels of mastery	Reference Book	Remarks
1	Cloud computing security challenges - cloud computing security architecture – data security life-cycle	Familiarity	3,5	
1	Security Patterns and architectural elements - Planning key Strategies for secure operation	Familiarity	3,5	
1	Encryption techniques – homomorphic encryption	Usage	7,8	
2	securing data Redaction - secure bitcoin - PKI – key management	Usage	7,8	
2	open web application security project (OWASP) Cloud Top 10 Security Risks - Security as a service.	Familiarity	7,8	
2	Virtualization security – securing hypervisor - securing virtual machines - designing virtual network for security	Familiarity	2,4	
2	Network Security in the cloud -	Familiarity	2,4	

	software-defined security - secure isolation strategy			
1	antifragile cloud infrastructure - Failure as a service	Familiarity	2,4	
1	Managed Security Service Provider(MSSP): Availability management – configuration management	Familiarity	1,7,9	
1	vulnerability management - identity management	Familiarity	1,7,9	
2	privacy, compliance and the cloud - privacy enhancing encryption	Familiarity	1,7,9	
1	Risk management – principles - assessing the risk	Familiarity	3,5	
1	strategies for managing risk – risk analysis framework	Familiarity	3,5	
1	security threats - intrusion detection	Familiarity	3,5	
2	Cloud security alliance – cloud controls matrix - cloud security standards guidance – security compliance	Familiarity	5	
2	NIST – PCI data security standards – SAS 70 - ISO 27001 – HIPAA – ITIL - FISMA - FIPS 140-2	Familiarity	5	
1	Cloud-Based IT Audit Process – System and Infrastructure lifecycle management for the cloud	Familiarity	6	
2	goverance, risk management and compliance (GRC) – cloud audit assurance – auditing – record generation, reporting and management	Familiarity	6	
2	tamper-proofing audit logs service level agreement (SLA) – legal safeguards - cloud morphing.	Familiarity	6	
2	Recent Trends	Familiarity		
Theory : 30 Hours (2 Credit hours /week) **Project** : 60 Non-contact Hours (1 credit hours / week) *(15 Weeks schedule)*				

CSE6029	CLOUD STORAGE TECHNOLOGIES	L,T,P,J,C 2,0,2,4,4
Objectives	The objective of the course is to impart an understanding of the need of cloud storage, its architecture the different cloud storage systems, their comparative merits and suitability to task specific deployment. The course will discuss the various enabling technologies for cloud storage	
Expected Outcome	Students who complete this course successfully are expected to: 1. Propose a plan to store, maintain, and deliver the massive amounts of media, software, documents, and other digital objects. 2. Create a comparative report on how to Store/ access data using different cloud storage technologies for a range of applications. 3. Explain the various architectures of different cloud storage. 4. Describe approaches to virtualize and integrate multiple storage platforms. 5. Devise a plan to secure and manage the storage infrastructure.	
Student Learning	1-Having an ability to apply mathematics and science in engineering applications 2-Having a clear understanding of the subject related concepts and of contemporary issues 7-Having computational thinking	

Module	Topics	L Hrs	SLO
1	**Essentials of Storages** Types of Storage - Data Processing: Local, Directly Attached, Remote - File Systems & Volume Management - Distributed/Clustered File Systems, Concatenation, Partitioning, etc. – RAIDs: Descriptions, Comparison - I/O & Capacity: Concatenation, Vertical Tier, Disk Splitting - Storage Taxonomy - Storage Levels: Block, File, Object - Backup - Snapshots, Disaster Recovery/Protection, Recovery Time – Caching – Replication	4	1, 2
2	**Storage Technology Devices** Storage Network Adapters (HBAs, SNICs and HCAs) – NAS appliances, servers and gateways – Storage devices: JBOD and disk array subsystem, Modular subsystem, Frame storage subsystem, Solid state disks, memory	4	1, 7

	disks, and SAN cache – Network Components: Hubs and concentrators – Switches and directors – Bridges, gateways and router devices – Intelligent switches, multifunction switches and storage domain controllers.		
3	**Fibre Channel Storage Area Networks & Intelligent Storage System** Components of FC SAN – FC Connectivity – Switched fabric ports – Fibre channel architecture – Link services – Fabric services. Components of intelligent storage system: front-end, back-end and physical disk – Storage provisioning: traditional and virtual – Types of intelligent storage systems	5	1, 7
4	**IP SAN, FcoE and Network Attached Storage** Components of iSCSI, connectivity, topologies, protocol stack, PDU, discovery, names, sessions, command sequencing – FCIP protocol stack and topology – Components of FCoE, frame structure and enabling technologies - Converged Enhanced Ethernet (CEE), transmission selection, congestion notification. Components of NAS – NAS I/O operation – NAS implementation – NAS file sharing protocols	6	1, 7
5	**Storage Virtualization** Virtualization in I/O path, limitations and requirements – Storage virtualization definition – Implementation consideration – Storage virtualization on block and file level – Storage virtualization on various levels of storage network: server, storage devices and network	3	1, 2
6	**Securing the Storage Infrastructure** Information security framework – Risk triad – Storage security domains – Security implementation in storage networking – Securing storage infrastructure in virtualized and cloud environments	3	1, 2

7	**Managing the Storage Infrastructure** Monitoring the storage infrastructure - Storage infrastructure management activities - Storage infrastructure management challenges - Developing an ideal solution - Information lifecycle management - Storage Tiering	3	1, 2
8	**Recent Trends**	2	2
Total		30	
List of Laboratory Experiments (in the areas of) 1. Configuring network files system. 2. Adding virtual hard drives using VMware 3. Creating RAID Array using VMware. 4. Simulating Drive failures. 5. Creating the clusters 6. Replicating Data in Cloud 7. Encrypting and decrypting data in Cloud 8. Implementing Deduplication in cloud storage 9. Storage tiering 10. Storage backup exercise - Design and develop data backup utility that copies all files in one folder to another storage location specified by the user 11. Cloud storage platforms : Design and implementation of Intelligent storage system 12. Study of disk drive performance using DiskSim simulation (opensource) tool		**30**	14

Reference Books

1. Greg Schulz: Cloud and virtual data storage networking, ISBN-13: 978- 1439851739, Auerbach Publications, 2011.
2. Hitachi Data Systems Academy: Storage Concepts: Storing and Managing Digital Data (Volume 1). ISBN-13: 978-0615656496, 2012.
3. EMC education services. Information Storage and Management: Storing, Managing, and Protecting Digital Information in Classic, Virtualized, and Cloud Environments, ISBN-13: 978-1118094839, Wiley, 2012.
4. Richard Barker, Paul Massiglia 2002, Storage area network essentials, Wiley New York
5. IBM Redbook “Introduction to Storage Area networks and System Networking” http://www.redbooks.ibm.com/redbooks/pdfs/sg245470.pdf

Knowledge Areas that contain topics and learning outcomes covered in the course

Knowledge Area	Total Hours of Coverage
CS: AR (Memory System Organization and Architecture)	1
CS:AR (Interfacing and Communication)	1
CS:IAS (Foundational concepts in security)	1
CS:IAS (Security policy and governance)	1
CS:IM (Distributed databases)	1
CS:OS (Virtual machines)	1
CS:OS (Device management)	1
CS:OS (Cloud computing)	2

Body of Knowledge coverage

[List the Knowledge Units covered in whole or in part in the course. If in part, please indicate which topics and/or learning outcomes are covered. For those not covered, you might want to indicate whether they are covered in another course or not covered in your curriculum at all. This section will likely be the most time-consuming to complete, but is the most valuable for educators planning to adopt the CS2013 guidelines.]

KA	Knowledge Unit	Topics Covered	Hours
CS: AR	Memory system organization and architecture	Storage systems and their technology	1
CS: AR	Interfacing and communication	External storage, drives, bus protocols, communication networks as another layer of remote access, RAID architecture	1
CS: IAS	Foundational concepts in security	Concepts of risk and attack vectors, authentication and authorization, access control	1
CS: IM	Distributed databases	Distributed data storage, data replication	1

CS:OS	Virtual machines	Types of virtualization, virtual file system	1
CS:OS	Device management	Recovery from failures	2
CS:PD	Cloud Computing	Cloud based storage	2
		All other topics	21
		Total hours	**30**

What is the format of the course?
[*Is it face to face, online or blended? How many contact hours? Does it have lectures, lab sessions, discussion classes?*]

This Course is designed with 100 minutes of in-classroom sessions per week and 100 minutes of lab hours per week as well as 200 minutes of non-contact time spent on implementing course related project. Generally this course should have the combination of lectures, in-class discussion, case studies, guest-lectures, mandatory off-class reading material, quizzes.

How are students assessed?
[*What type, and number, of assignments are students are expected to do? (Papers, problem sets, programming projects, etc.). How long do you expect students to spend on completing assessed work?]*

- Students are assessed on a combination group activities, classroom discussion, projects, and continuous, final assessment tests.

- Students can earn additional weightage based on certificate of completion of a related MOOC course.

Session wise Plan

Students Outcome coverage: 1, 2, 6,7,14.

Class Hour	Lab Hour	Topics Coverage	Levels of Mastery	Reference Book	Remarks
2	-	**Essentials of Storages** Types of Storage - Data Processing: Local, Directly Attached, Remote - File Systems & Volume Management - Distributed/Clustered File Systems, Concatenation, Partitioning, etc.	Familiar	1	
2	-	RAIDs - Descriptions, Comparison, I/O & Capacity - Concatenation, Vertical Tier, Disk Splitting. Storage Taxonomy - Storage Levels - Block, File, Object Backup - Snapshots, Disaster Recovery/Protection, Recovery Time.	Usage	1	
2	-	**Storage Technology Devices** Servers and processors (nodes) – Storage Network Adapters (HBAs, SNICs and HCAs) – NAS appliances, servers and gateways – Storage devices: JBOD and disk array subsystem, Modular subsystem, Frame storage subsystem, Solid state disks, memory disks, and SAN cache.	Usage	4,5	
2	-	Network Components: Hubs and concentrators – Switches and directors – Bridges, gateways and router devices – Intelligent switches, multifunction switches and storage domain controllers.	Usage	4,5	
2	-	**Intelligent Storage System** Components of intelligent storage system: front-end, back-end and physical disk – Storage provisioning:	Familiar	2,3	

Class Hour	Lab Hour	Topics Coverage	Levels of Mastery	Reference Book	Remarks
		traditional and virtual – Types of intelligent storage systems.			
3	-	**Fibre Channel Storage Area Networks** Components of FC SAN – FC Connectivity – Switched fabric ports – Fibre channel architecture – Link services – Fabric services.	Familiar	4,5	
2	-	**IP SAN and FCoE** Components of iSCSI, connectivity, topologies, protocol stack, PDU, discovery, names, sessions, command sequencing – FCIP protocol stack and topology.	Familiar	1,2	
2	-	Components of FcoE, frame structure and enabling technologies - Converged Enhanced Ethernet (CEE), transmission selection, congestion notification.	Familiar	2	
3	-	**Network Attached Storage** Components of NAS – NAS I/O operation – NAS implementation – NAS file sharing protocols.	Usage	4	
2	-	**Storage Virtualization** Virtualization in I/O path, limitations and requirements – Storage virtualization definition – Implementation consideration.	Usage	3	
2	-	Storage virtualization on block and file level – Storage virtualization on various levels of storage network: server, storage devices and network.	Usage	3	
3	-	**Securing the Storage Infrastructure** Information security framework – Risk triad – Storage security domains –	Usage	4	

Class Hour	Lab Hour	Topics Coverage	Levels of Mastery	Reference Book	Remarks
		Security implementation in storage networking – Securing storage infrastructure in virtualized and cloud environments.			
2	-	**Managing the Storage Infrastructure** Monitoring the storage infrastructure - Storage infrastructure management activities - Storage infrastructure management challenges.	Familiar	2	
1	-	Developing an ideal solution - Information lifecycle management - Storage Tiering.	Familiar	2	
Theory : 30 Hours (2 Credit hours /week) **Lab**: 30 Hours (1 Credit hour / week) **Project** : 60 Non-contact Hours (1 credit hours / week) *(15 Weeks schedule)*					

CSE6030	Design Thinking	L,T,P,J,C 2,0,0,4,3
Objectives	The intent of the course is to develop a creative thinking skill among students, which is a key to innovation. This includes the ability to gain deep insights about users of the product, to define and reframe problems, to generate alternative approaches and to finalize the solution for solving the problem.	
Expected Outcome	On successful completion of the course the student will be able to, 1. Identify the market needs and the potential stakeholders for the product. 2. Design a product prototype for the company satisfying company's and stakeholders' demands. 3. Perform a feasibility study and design a product that meets requirements.	
Student Learning Outcomes (SLOs)	• Having design thinking capability. (5) • Having problem solving ability- solving social issues and engineering problems. (9) • Having a clear understanding of professional and ethical responsibility (10)	

Module	Description	L Hrs	SLO
1	**Introduction** Why Design thinking? – Standish group chaos report – Software crisis – Software failures - Key steps to Design Process – Design Ability – How to Design?	6	5
2	**Stakeholder Analysis** Converting Need to Demand – Identifying Stakeholders – Insight, Observation and Empathy – Mental Matrix	4	5,9,10
3	**Design** Principles of Good Design – Articulating Design – Design to think – Design to win – Design to please – Designing to use	4	5,9,10
4	**Meeting the Corporate Expectation** Storytelling – Modelling systems – Designing together – Impact of Human Psychology on Design – Design Activism – Design for tomorrow	4	5,9,10
5	**New Social contract** Product to customers – Usability test – Business canvas model – Applicability to cloud users – Economic feasibility	4	5,9,10
6	**Agile Software Development** Lean and Design Thinking – People centric design – Pair Programming – Test Driven Development – Refactoring – Continuous Integration – Planning and Reviews – Clean code	3	5,9,10

7	**Branding** Basics – Creation, Voice casting and Interpretation – Simulation and Evaluation	3	5,9,10
8	**Recent Trends**	2	10
Project # Compulsory team project of size 3 or 4 members # Innovative product ideas in the specified domain should be attempted # Prototype, documentation, financial report should be submitted. # Assessment on a continuous basis with a minimum of 2 reviews. **Indicative Project Domains:** 1. Healthcare 2. Agriculture 3. Cybercrime 4. Social media crime 5. Grocery shopping 6. Mobile marketing 7. Digital analytics 8. Digital marketing		**60** [Non Contact hrs]	17

Reference Books

1) Nigel Cross, "Design thinking", Berg Publishers, 2011.
2) Roger L. Martin, "The Design of Business: Why Design Thinking is the Next Competitive Advantage", Harvard Business Press, 2009.
3) Robert C. Martin, "Agile Software Development, Principles, Patterns, and Practices", Pearson Education publishers, 2003.
4) Business canvas model : http://www.businessmodelgeneration.com/downloads/business_model_canvas_poster.pdf
5) Standish survey: https://www.projectsmart.co.uk/white-papers/chaos-report.pdf
6) Design Activism: http://theprotocity.com/designing-activism-an-interview-with-ann-thorpe/
7) Principles of Design: http://www.archdaily.com/198583/dieter-rams-10-principles-of-%25e2%2580%259cgood-design%25e2%2580%259d
8) Design Thinking http://ocw.mit.edu/courses/engineering-systems-division/esd-051j-engineering-innovation-and-design-fall-2012/lecture-notes-and-videos/
9) Kristin Fontichiaro "Design Thinking", Cherry Lake Publishing, 2015

Design Thinking

Description of course

Design thinking course deals with the ways of converting the market needs to productive products through the analysis of various alternative approaches in every step of development. The course will give the students better insight to analyze the methods of analyzing market, considering alternate design approaches, identifying various stakeholders, developing a user-friendly prototype satisfying customers and company. Design Thinking helps students develop realistic and innovative solutions for the real world problems. Students will be exposed to this thinking through the development of a sample prototype as part of the course.

Knowledge Areas that contain topics and learning outcomes covered in the course

Knowledge Area	Total Hours of Coverage
CE: CSE(Computer Systems Engineering)	14
CE: SWE(Software Engineering)	6
CE: SPR (Social and Professional Issues)	10

Body of Knowledge coverage

KA	Knowledge Unit	Topics Covered	Hours
CE: CSE	CE:CSE1 Lifecycle	Key steps to Design Process	2
CE: CSE	CE:CSE2 Requirements analysis and elicitation	Converting Need to Demand – Identifying Stakeholders – Insight, Observation and Empathy – Mental Matrix	4
CE: CSE	CE:CSE3 Specification	Principles of Good Design – Articulating Design	2
CE: CSE	CE:CSE4 Architectural design	Design to think – Design to win – Design to please – Designing to use	2
CE: CSE	CE:CSE5 Testing	Usability test	1
CE: CSE	CE:CSE8 Concurrent design	Lean and Design Thinking – People centric design – Pair Programming – Test Driven Development — Continuous Integration – Planning and Reviews	2

CE: CSE	CE: CSE11 Reliability and fault tolerance	Refactoring – Clean code	1
CE: SPR	CE:SPR3 Professional and ethical responsibilities	Design Activism – Design for tomorrow Product to customers	3
CE: SPR	CE:SPR4 Risks and Liabilities	Impact of Human Psychology on Design	1
CE: SPR	CE:SPR5 Intellectual property	Basics – Creation, Voice casting and Interpretation – Simulation and Evaluation	3
CE: SPR	CE: SPR8 Economics in computing	Economic feasibility	1
CE: SWE	CE: SWE8 Software Project Management	Design Ability – How to Design? Storytelling – Modelling systems – Designing together Business canvas model – Applicability to cloud users	4
CE: SWE	CE: SWE0 History and Overview	Standish group chaos report 2015 – Software crisis – Software failures - Why Design thinking?	2

Session wise Plan

Class Hour	Lab Hour	Topics Coverage	Levels of Mastery	Reference Book	Remarks
1		Why Design thinking?	Familiarity	1	
1		Standish group chaos report	Familiarity	5	
3		Software crisis, Software failures Key steps to design process	Familiarity	8	
1		Design ability, How to design	Familiarity	1	
4		Converting Need to Demand Identifying Stakeholders Insight, Observation and Empathy Mental Matrix	Familiarity	8	
1		Principles of Good Design Articulating Design	Familiarity	7	
3		Design to think, Design to win Design to please, Designing to use	Familiarity	1, 2	

Class Hour	Lab Hour	Topics Coverage	Levels of Mastery	Reference Book	Remarks
1		Storytelling	Familiarity	3	
1		Modelling systems Designing together	Familiarity	1	
2		Impact of Human Psychology on Design Design Activism Design for tomorrow	Familiarity	1,6,8	
1		Product to customers Usability test	Ugage	3	
3		Business canvas model Applicability to cloud users Economic feasibility	Usage	4	
3		Lean and Design Thinking People centric design Pair Programming Test Driven Development Refactoring, Continuous Integration Planning and Reviews Clean code	Usage	3	
3		Basics of Branding Creation, Voice casting and Interpretation Simulation and Evaluation	Familiarity	8	
2		Recent Trades			
Theory : 30 Hours (2 Credit hours /week, 15 Weeks schedule) Project : 60 Non-contact hours (1 credit hour / week)					

CSE6031	Cyber Space and Information Technology Laws	L,T,P,J,C 2,0,0,0,2
Objectives	• To provide the basic understanding of the legal issues arising in the Cyber Space • To provide an insight into the myriad security and privacy issues in Cyberspace • To provide a comprehensive understanding about the laws applicable to the Cyber Space • To analyze the grey areas and the emerging trends in Cyber Space.	
Expected Outcomes	After successfully completing the course the student should be able to • List the basic issues arising in the cyber space • Discuss about the contemporary technologies that create unparalleled predicaments in cyber space • Outline the IT Laws that are applied to the cyber space	
Student Learning Outcomes	• Having a clear understanding of the subject related concepts and of contemporary issues (2) • Having problem solving ability- solving social issues and engineering problems (9) • Having a clear understanding of professional and ethical responsibility (10)	

Module	Topics	L Hrs	SLO
1	**Basic concepts & Techniques** Cyber Space – History Of Cyber Space – Information Technology Act, 2000 - Outline of the Act- Aims and Objectives of the Act- Applicability of the Act	3	2
2	**Jurisdictional Issues** Types of Jurisdiction – Jurisdiction for Internet cases -Territorial Jurisdiction and Cyber Space – Minimum Contacts theory and Purposeful Availment theory – French Yahoo case, Dow Jones case- IT Act on Jurisdiction	4	2,9
3	**Digital Signatures** Asymmetric Cryptography - Public and Private Key-Hash functions- Verification -Legality of the Digital Signature- Certifying Authorities- Digital Signature Certificates. Validity of expired and local certificates	5	2,10

4	**eGovernance** Legality of eGovernance - security issues and legislative control of eGovernance- Scope Of e Governance in India and other developed countries – Digital Evidence	**4**	2,10
5	**eCommerce** Online Contracts- Click wrap and Shrink wrap Contracts- Time and Delivery of the Contract- eCommerce websites- Online Payments- Indian law of electronic contracts	**5**	2
6	**Intellectual Property and Cyber Space** Copyright and the Internet – Linking – Inlining – Framing- peer to Peer networking, Commissioned work Trademark Issues- Domain Name – ICANN Policy – Uniform Dispute Resolution Policy- Dispute Resolution Procedure Patent-ability and Computer Software – Sui generis regime. Indian position on computer related patents, Copyright	**6**	2,9
7	**Security Issues and Cyber Crimes** Cyber privacy –Crimes against property – Crimes against the Person- Crimes against the State- Crimes against the computer network- Financial Crimes- Cyber Security and safeguards	**1**	10
8	**Recent Trends** Online Dispute Resolution / eCourts	**2**	2
		30	

Reference Books

1) S.K. Verma and Raman Mittal, "Legal Dimensions of Cyber Space", Universal Law Publishing, 2004
2) Sachin Rastogi, " Insights into E - Contracts in India", LexisNexis, Ist Edition, 2013
3) Karnika Seth, "Computers, Internet and New Technology Laws", LexisNexis, 2013
4) Kamath Nandan, "Law Relating to Computers Internet & E-commerce (A Guide to Cyber laws & the Information Technology Act, Rules, Regulations and Notifications along with Latest Case Laws)", 2012, Universal Law Publishing, 2016
5) Christpher Millard, "Cloud Computing Law", OUP-UK, 2013
6) Chris Reed , "Computer Law", Oxford University Press, 7th Edition, 2012

What is the format of the course?
[*Is it face to face, online or blended? How many contact hours? Does it have lectures, lab sessions, discussion classes?*]

There will be 50 minutes of class with 2 hours every week and the generally this course should have the combination of lectures, in-class discussion, case studies, guest-lectures, mandatory of pre-class reading material, quizzes.

How are students assessed?

- Students are assessed on a combination group activities, classroom discussion, and continuous, final assessment tests.
- Presentations will also be assessed

Additional topics
[List notable topics covered in the course that you do not find in the CS2013 Body of Knowledge]

Other comments
[optional]

CSE6032	Cloud Computing Paradigm on Software Engineering	L,T,P,J,C 2,0,0,4,3
Objectives	• To provide a basic understanding of core concepts of software engineering on the cloud platform • To familiarize with the state of the art software engineering approaches for developing cloud based applications • To provide the conceptual understanding of cloud-based application architecture • To understand the field of software engineering as relevant to the emerging new paradigm of cloud computing	
Expected Outcomes	On completion of this course, students will be capable of • Creating cloud based application by using the appropriate software process model • Developing test plan to test the cloud application • Deploying enumerating techniques for cloud based applications • Enhancing and Maintaining the system in Cloud	
Student Learning Outcomes	• Having a clear understanding of the subject related concepts and of contemporary issues (2) • Having an ability to design a component or a product applying all the relevant standards and with realistic constraints (6) • Having problem solving ability- solving social issues and engineering problems (9)	

Module	Topics	L Hrs	SLO
1	**Introduction and Impact of cloud paradigm on software engineering** Introduction to cloud computing platform-traditional software engineering process-creating web services –Impact of cloud computing on software engineering- software process model for cloud platform	**3**	2
2	**Cloud induced Transformations in software engineering** Cloud-Sponsored Transformations for IT, Leveraging Clouds for Global Software Development (GSD), Combination of Agility and Cloud Infrastructure for Next-Generation Software Engineering, Convergence of Service and Cloud Paradigms	**5**	6
3	**Software Development Life Cycle for Cloud Platform** Impact of cloud services on software development life cycle, Cloud based development using classic life cycle model, Business Requirements Engineering for Developing Cloud Computing Services	**4**	2
4	**Software Design Strategies for Cloud Adoption** Feature-Driven Design of SaaS Architectures, Impact of Cloud Adoption on Agile Software Development, Technical Strategies and Architectural Patterns for Migrating Legacy Systems to the Cloud	**4**	2

5	**Software Reuse – Cloud** Components and Frameworks in Cloud Era, Goal based requirements elicitation for service reuse in cloud computing. Reuse of Architecture - Reuse across multiple architectures Reuse Assets - Software Asset Re-Use : Migration of Data-Intensive Legacy system to the cloud computing. Reuse in Cloud Application - Reusing Transaction models for dependable cloud	**6**	9
6	**Performance of Cloud Based Software Applications** Efficient Practices and Frameworks for Cloud-Based, Application Development, Methodology for Identifying the Relationships Between Performance Factors for Cloud Computing Applications	**4**	6
7	**Testing perspectives for cloud based applications** Testing in the cloud : Strategies, risks and benefits,	**2**	9
8	**Recent Trends**	**2**	2
		30	
Indicative list of Projects 1) Agriculture Updates by SMS A Cloud Computing Approach 2) Cloud Computing for Agent-Based Urban Transportation Systems Project 3) A Hybrid Cloud Approach for Secure Authorized Deduplication 4) Privacy-Preserving Public Auditing for Shared Data in the Cloud 5) PACK: Prediction-Based Cloud Bandwidth and Cost Reduction System 6) Cloud-based Online Banking 7) Cloud File/Data storage 8) Performance evaluation of Cloud applications			17

Reference Books:

1. Software engineering frameworks for the cloud computing paradigm, by Zaigham Mahmood, Saqib Saeed, Springer London,2015
2. Software Reuse in the Emerging Cloud Computing Era, edited by Yang, Hongji, Harrisburg PA, United states,2012
3. Migrating Legacy Applications: Challenges in Service Oriented Architecture and Cloud Computing Environment , edited by Ionita, Anca Daniela, 2012

Cloud Computing Paradigm on Software Engineering

Knowledge Areas that contain topics and learning outcomes covered in the course

Knowledge Area	Total Hours of Coverage
CS:SE CE:PD	10 20

What is covered in the course?

Part 1: Introduction to Cloud Computing Transformation and Software Engineering
Introduction of Software Engineering process, cloud computing paradigms and cloud infrastructures are discussed.

Part II: SDLC and design for Cloud
Study of Requirements and Cloud services on SDLC using Agile technology, SaaS Architecture and patterns.

Part III: Performance of Cloud Aided Software Engineering
Study of software system for the development of Cloud Applications using multi-scheduling, and identifying the factors affecting the performance of Cloud Applications.

Part IV: Testing Cloud based applications

Study of strategies, risks and benefits of cloud testing.

What is the format of the course?

[*Is it face to face, online or blended? How many contact hours? Does it have lectures, lab sessions, discussion classes?*]

This Course is designed with 100 minutes of in-classroom sessions per week, 60 minutes of video/reading instructional material per week, as well as 200 minutes of non-contact time spent on implementing course related project. Generally this course should have the combination of lectures, in-class discussion, case studies, guest-lectures, mandatory off-class reading material, quizzes.

How are students assessed?
[*What type, and number, of assignments are students are expected to do? (papers, problem sets, programming projects, etc.). How long do you expect students to spend on completing assessed work?]*

- Students are assessed on a combination group activities, classroom discussion, projects, and continuous, final assessment tests.
- Additional weightage will be given based on their rank in sourced projects.
- Students can earn additional weightage based on certificate of completion of a related MOOC course.

SESSION WISE PLAN

Student Outcomes Covered: 2,6,7,9,14,17

Class Hour	Topic Covered	levels of mastery	Reference Book	Remarks
2	Introduction to cloud computing platform-traditional software engineering process	Familiarity	1	
1	enabled artifacts-creating web services –Impact of cloud computing on software engineering	Usage	1	
1	software process model for cloud platform	Familiarity	1	
2	Cloud-Sponsored Transformations for IT, Leveraging Clouds for Global Software Development (GSD)	Usage	1	
2	Combination of Agility and Cloud Infrastructure for Next-Generation Software Engineering, Convergence of Service and Cloud Paradigms	Familiarity	1	
2	Impact of cloud services on software development life cycle, Cloud based development using classic life cycle model	Usage	1	
2	Business Requirements Engineering for Developing Cloud Computing Services	Familiarity	1	
1	Feature-Driven Design of SaaS Architectures	Familiarity	1	
1	Impact of Cloud Adoption on Agile Software Development	Usage	1	
2	Technical Strategies and Architectural Patterns for Migrating Legacy Systems to the Cloud	Usage	1	

2	Components and Frameworks in Cloud Era, Goal based requirements elicitation for service reuse in cloud computing. Reuse of Architecture - Reuse across multiple architectures	Familiarity	2	
2	Reuse Assets - Software Asset Re-Use : Migration of Data-Intensive Legacy system to the cloud computing. Reuse in Cloud Application - Reusing Transaction models for dependable clou	Usage	2	
2	Efficient Practices and Frameworks for Cloud-Based, Application Development	Familiarity	1	
1	Methodology for Identifying the Relationships Between Performance Factors for Cloud Computing Applications	Usage	1	
1	Methodology for Identifying the Relationships Between Performance Factors for Cloud Computing Applications	Usage	1	
2	Testing in the cloud : Strategies	Usage	1	
2	risks and benefits	Familiarity	1	
2	Recent Trends	Usage		
Theory : 30 Hours (2 Credit hours /week) **Project** : 60 Non-contact Hours (1 credit hours / week) *(15 Weeks schedule)*				