SEMESTER II

Discipline : COMPUTER SCIENCE AND ENGINEERING

Programme : M.Tech.

Stream :CSE (COMPUTER SCIENCE AND ENGINEERING)

SECOND SEMESTER: COMPUTER SCIENCE AND ENGINEERING
BRANCH: COMPUTER SCIENCE AND ENGINEERING
STREAM: CSE

Sl. No:	Slot	Course Code	Course Type	Course Category	Course Title (Course Name)	Credit Structure				SS	Total Marks		Credits	Hrs./Week
						L	T	P	R		CIE	ESE		
THEORY														
1	A	R25CST201	DC	DC	Advanced Data Structures And Algorithms	3	0	0	0	4.5	40	60	3	3
2	B	R25CST202	PC	PC	Advanced Operating Systems	3	0	0	0	4.5	40	60	3	3
3	C	R25CSP203	PC	PC	Mini Project	0	0	4	0	2	40	60	2	4
4	D	R25CST2X4	PE	PE	Program Elective-III	3	0	0	0	4.5	40	60	3	3
5	E	R25CST2X5	PE	PE	Program Elective-IV	3	0	0	0	4.5	40	60	3	3
6	F	R25CST2X6	IE	IE	Interdisciplinary Elective	3	0	0	0	4.5	40	60	3	3
PRACTICALS														
7	G	R25CSP207	PC	L	Computing Lab 2	0	0	2	0	1	100	0	1	2
Total										**25.5**			**18**	**21**

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
R25CST201	ADVANCED DATA STRUCTURES AND ALGORITHMS	DISCIPLINE CORE 2	3	0	0	3

Preamble: The course introduces advanced data structures and algorithms in different domains. The goal of this course is to provide a solid background in the design and analysis of the major classes of algorithms. The course helps the learners to develop their own versions for a given computational task and to compare and contrast their performance.

Course Outcomes: After the completion of the course the student will be able to:*

CO 1	Analyse the relevance of amortized analysis and applications. (Cognitive Level: Apply)
CO 2	Illustrate string matching algorithms. (Cognitive Level: Apply)
CO 3	Illustrate advanced data structures like Binomial heap, Fibonacci heap, Disjoint set and string matching algorithms. (Cognitive Level: Apply)
CO 4	Illustrate network flow algorithms and applications. (Cognitive Level: Apply)
CO 5	Make use of probabilistic algorithms and approximation algorithms in computing. (Cognitive Level: Apply)
CO 6	Design, develop and implement software using advanced data structures and algorithms. (Cognitive Level: Create)

* The COs shown are only indicative. For each course, there can be 4 to 6 COs.

Program Outcomes (PO)

Outcomes are the attributes that are to be demonstrated by a graduate after completing the course.

PO1: An ability to independently carry out research/investigation and development work in engineering and allied streams

PO2: An ability to communicate effectively, write and present technical reports on complex engineering activities by interacting with the engineering fraternity and with society at large.

PO3: An ability to demonstrate a degree of mastery over the area as per the specialization of the program. The mastery should be at a level higher than the requirements in the appropriate bachelor program

PO4: An ability to apply stream knowledge to design or develop solutions for real world problems by following the standards

PO5: An ability to identify, select and apply appropriate techniques, resources and state-of-the-art tool to model, analyse and solve practical engineering problems.

PO6: An ability to engage in life-long learning for the design and development related to the stream related problems taking into consideration sustainability, societal, ethical and environmental aspects

PO7: An ability to develop cognitive load management skills related to project management and finance which focus on Entrepreneurship and Industry relevance.

Mapping of course outcomes with program outcomes

	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**
CO 1	2		3	2		1	
CO 2	2		3	2		1	
CO 3	2		3	2		1	
CO 4	2		3	3		1	
CO 5	2		3	2		1	
CO 6	3	2	3	3	3	2	2

Assessment Pattern

Bloom's Category	**End Semester Examination**
Apply	80%
Analyse	20%
Evaluate	
Create	

Mark distribution

Total Marks	**CIE**	**ESE**	**ESE Duration**
100	40	60	2.5 hours

Continuous Internal Evaluation Pattern:

Evaluation shall only be based on application, analysis or design based questions (for both internal and end semester examinations).

Continuous Internal Evaluation : 40 marks

Micro project/Course based project : 20 marks
Course based task/Seminar/Quiz : 10 marks
Test paper, 1 no. : 10 marks
The project shall be done individually. Group projects not permitted.
Test paper shall include minimum 80% of the syllabus.

Course based task/test paper questions shall be useful in the testing of knowledge, skills, comprehension, application, analysis, synthesis, evaluation and understanding of the students.

End Semester Examination Pattern:

The end semester examination will be conducted by the University. There will be two parts; Part A and Part B. Part A contain 5 numerical questions with 1 question from each module, having 5 marks for each question. (such questions shall be useful in the testing of knowledge, skills, comprehension, application, analysis, synthesis, evaluation and understanding of the students). Students shall answer all questions.
Part B will contain 7 questions (such questions shall be useful in the testing of overall achievement and maturity of the students in a course, through long answer questions relating to theoretical/practical knowledge, derivations, problem solving and quantitative evaluation), with minimum one question from each module of which student should answer any five. Each question can carry 7 marks.
Total duration of the examination will be 150 minutes.

Course Level Assessment Questions
Course Outcome 1 (CO1):

1. Explain how the accounting method of amortized analysis can be applied to stack operations.

2. Suppose we perform a sequence of n operations on a data structure in which the i^{th} operation costs i if i is an exact power of 2, and 1 otherwise. Use aggregate analysis to determine the amortized cost per operation.

3. What is the total cost of executing n of the stack operations PUSH, POP, and MULTIPOP, assuming that the stack begins with s_0 objects and finishes with s_n objects? Use potential method.

Course Outcome 2 (CO2)

1. Use an aggregate analysis to show that the running time of KMP-MATCHER is $\theta(n)$.

2. Working modulo q = 11, how many spurious hits does the Rabin-Karp matcher encounter in the text T = 3141592653589793 when looking for the pattern P = 26?

3. Compute the prefix function π for the pattern ababbabbabbababbabb

Course Outcome 3(CO3):

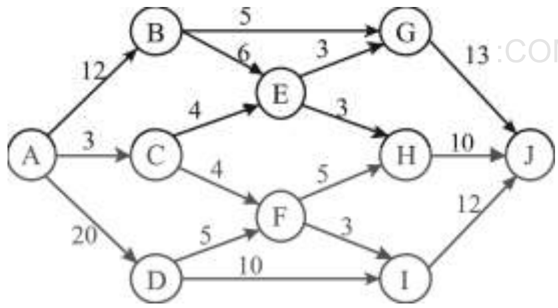
1. Analyse the time complexity of decrease-key operation of Fibonacci heap.

2. Illustrate extract-min operation of Binomial heap.

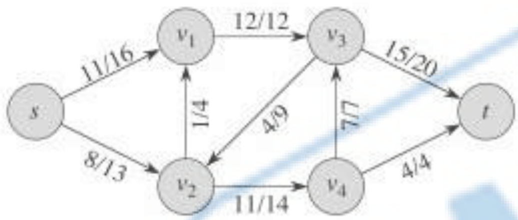
3. Explain the heuristics used in disjoint set data structure to improve the running time.

Course Outcome 4 (CO4):

1. Show the execution of the Edmonds-Karp algorithm on the given flow network (source: A and sink: J).



2. In the following figure, compute flow across the cut ({s, v_2, v_4}, {v_1, v_3, t}). What is the capacity of this cut?



3. State and prove max flow min cut theorem.

Course Outcome 5 (CO5):

1. Illustrate Miller-Rabin primality testing method.

2. Explain probabilistic selection algorithm.

Course Outcome 6 (CO6):

1. Explain the approximation algorithm for subset sum problem.

2. Consider each of the following words as a set of letters: {*arid, dash, drain, heard, lost, nose, shun, slate, snare, thread*}. Show which set cover GREEDY-SET-COVER produces when we break ties in favor of the word that appears first in the dictionary.

Syllabus

Module – 1 (Amortized analysis and String matching)

Overview of asymptotic notations and complexity analysis, Amortized analysis – aggregate analysis, accounting method, potential method.

String matching – introduction, Rabin-Karp algorithm, Knuth-Morris-Pratt algorithm.

Module – 2 (Advanced data structures)

Overview of binary heap operations, Binomial tree and heap, Binomial heap operations, Fibonacci heap structure, Fibonacci heap operations, Disjoint set – overview, linked list representation, disjoint set forests.

Module – 3 (Network flow)

Network flow properties, examples, residual network, augmenting path, cut of network, maxflow-mincut theorem, Ford-Fulkerson algorithm, Edmonds-Karp algorithm, maximum bipartite matching.

Module – 4 (Probabilistic algorithms)

Introduction, types of probabilistic algorithms, Numerical algorithms – Numerical integration, Probabilistic counting, Monte-Carlo algorithms – Verifying matrix multiplication.

Number theory fundamentals – modular arithmetic, modular exponentiation, Euler's Theorem and Fermat's Theorem, Primality testing – Miller-Rabin test.

Las Vegas algorithms – Probabilistic selection and quick sort.

Module – 5 (Approximation algorithms)

Introduction, Vertex-cover problem, Traveling-salesman problem, Set-covering problem, Subset-sum problem.

Course Plan (For 3 credit courses, the content can be for 40 hrs and for 2 credit courses, the content can be for 26 hrs. The audit course in third semester can have content for 30 hours).

No	Topic	No. of Lectures (37)
1	**Module – 1 (Amortized analysis and String matching)**	
1.1	Overview of asymptotic notations and complexity analysis	1
1.2	Amortized analysis – aggregate analysis	1
1.3	accounting method	1
1.4	potential method	1
1.5	String matching – introduction	1
1.6	Rabin-Karp algorithm	1
1.7	Knuth-Morris-Pratt algorithm (1)	1
1.8	Knuth-Morris-Pratt algorithm (2)	1
2	**Module – 2 (Advanced data structures)**	
2.1	Overview of binary heap operations	1
2.2	Binomial tree and heap	1
2.3	Binomial heap operations (1)	1
2.4	Binomial heap operations (2)	1
2.5	Fibonacci heap structure	1
2.6	Fibonacci heap operations (1)	1
2.7	Fibonacci heap operations (2)	1
2.8	Disjoint set – overview, linked list representation	1
2.9	disjoint set forests	1
3	**Module – 3 (Network flow)**	
3.1	Network flow properties, examples	1
3.2	residual network, augmenting path, cut of network	1
3.3	maxflow-mincut theorem	1
3.4	Ford-Fulkerson algorithm	1
3.5	Edmonds-Karp algorithm	1
3.6	Maximum bipartite matching	1
4	**Module – 4 (Probabilistic algorithms)**	
4.1	Introduction, types of probabilistic algorithms	1
4.2	Numerical algorithms – Numerical integration, Probabilistic counting	1
4.3	Monte-Carlo algorithms – Verifying matrix multiplication	1
4.4	Number theory fundamentals – modular arithmetic, modular exponentiation	1
4.5	Euler's Theorem and Fermat's Theorem	1
4.6	Primality testing – Miller-Rabin test (1)	1
4.7	Primality testing – Miller-Rabin test (2)	1
4.8	Las Vegas algorithms – Probabilistic selection and quick sort	1
5	**Module – 5 (Approximation algorithms)**	

5.1	Introduction	1
5.2	Vertex-cover problem	1
5.3	Traveling-salesman problem	1
5.4	Set-covering problem	1
5.5	Subset-sum problem (1)	1
5.6	Subset-sum problem (2)	1

Reference Books

1. T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, "Introduction to Algorithms", MIT Press, 3rd edition, 2009.

2. Gilles Brassard and Paul Bratley, "Fundamentals of algorithms", Prentice-hall of India Private Limited, 2001.

3. Rajeev Motwani, Prabhakar Raghavan, "Randomized Algorithms", Cambridge University Press, 2000.

4. Dexter C. Kozen, "The Design and Analysis of Algorithms", Springer.

5. Jon Kleinberg and Eva Tardos, "Algorithm Design", Pearson Education, 2006.

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
R25CST202	**ADVANCED OPERATING SYSTEMS**	**PROGRAM CORE 3**	**3**	**0**	**0**	**3**

Preamble: Study of this course enables the learners to understand the configuration and functions of OS Kernel and have an overview on concepts implemented in modern operating systems. The course focus on providing information on the design and implementation of the Linux kernel modules. This course will help the learners to suggest solutions/ modify the existing architectural features.

Course Outcomes: After the completion of the course the student will be able to

CO 1	Illustrate the concepts of process management and process scheduling mechanisms employed in the Linux operating system. **(Cognitive Knowledge Level: Apply)**
CO 2	Describe the set of interfaces by which the process running in user space can interact with the system and how the Kernel manages the various interrupts. **(Cognitive Knowledge Level: Apply)**
CO 3	Apply various synchronization methods to write race free code. **(Cognitive Knowledge Level: Apply)**
CO 4	Demonstrate how the kernel handles memory and implementation of the file system. **(Cognitive Knowledge Level: Apply)**
CO 5	Analyze how kernel manages block devices and their requests and identify the issues to be considered in writing portable codes **(Cognitive Knowledge Level: Analyze)**
CO 6	Design and implement different kernel modules. **(Cognitive Knowledge Level: Create)**

Program Outcomes (PO)

Outcomes are the attributes that are to be demonstrated by a graduate after completing the course.

PO1: An ability to independently carry out research/investigation and development work in engineering and allied streams

PO2: An ability to communicate effectively, write and present technical reports on complex engineering activities by interacting with the engineering fraternity and with society at large.

PO3: An ability to demonstrate a degree of mastery over the area as per the specialization of the program. The mastery should be at a level higher than the requirements in the appropriate bachelor program

PO4: An ability to apply stream knowledge to design or develop solutions for real world problems by following the standards

PO5: An ability to identify, select and apply appropriate techniques, resources and state-of-the-art tool to model, analyse and solve practical engineering problems.

PO6: An ability to engage in life-long learning for the design and development related to the stream related problems taking into consideration sustainability, societal, ethical and environmental aspects

PO7: An ability to develop cognitive load management skills related to project management and finance which focus on Entrepreneurship and Industry relevance.

Mapping of course outcomes with program outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7
CO 1	2		2	3	2	2	
CO 2	2		2	3	2	2	
CO 3	2		2	3	3	2	
CO 4	2		2	3	2	3	
CO 5	3		3	2	3	2	
CO 6	3	3	3	3	3	2	2

Assessment Pattern

Bloom's Category	End Semester Examination
Apply	70%-80%
Analyze	30%-40%
Evaluate	
Create	

Mark distribution

Total Marks	CIE	ESE	ESE Duration
100	40	60	2.5 hours

Continuous Internal Evaluation Pattern:

Evaluation shall only be based on application, analysis or design-based questions (for both internal and end semester examinations).

Continuous Internal Evaluation Pattern:

Evaluation shall only be based on application, analysis or design based questions (for both internal and end semester examinations).

Continuous Internal Evaluation : 40 marks

Micro project/Course based project : 20 marks
Course based task/Seminar/Quiz : 10 marks
Test paper, 1 no. : 10 marks
The project shall be done individually. Group projects not permitted.
Test paper shall include minimum 80% of the syllabus.
Course based task/test paper questions shall be useful in the testing of knowledge, skills, comprehension, application, analysis, synthesis, evaluation and understanding of the students.

End Semester Examination Pattern:

The end semester examination will be conducted by the University. There will be two parts; Part A and Part B. Part A contain 5 numerical questions with 1 question from each module, having 5 marks for each question. (such questions shall be useful in the testing of knowledge, skills, comprehension, application, analysis, synthesis, evaluation and understanding of the students). Students shall answer all questions.
Part B will contain 7 questions (such questions shall be useful in the testing of overall achievement and maturity of the students in a course, through long answer questions relating to theoretical/practical knowledge, derivations, problem solving and quantitative evaluation), with minimum one question from each module of which student should answer any five. Each question can carry 7 marks.
Total duration of the examination will be 150 minutes.

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. How does Linux process scheduler ensure a fair share of its computational power among the executing process.
2. By default the child process starts execution in Linux. Discuss the reason behind this.
3. How is TASK_INTERRUPTIBLE state different from TASK_UNINTERRUPTIBLE state ? Suppose a process is waiting for a specific event to happen. Which state should it be assigned?

Course Outcome 2 (CO2):

1. Illustrate the role of system calls in Linux.
2. Can we use work queues in an interrupt context? Justify your answer.

3. In interrupt processing, if the deferred work needs to run in interrupt context and should guarantee that no two of same type can run concurrently, which bottom half can be preferred. Write notes on its implementation and usage.

Course Outcome 3(CO3):

1. What trouble arises if a kernel code which has acquired a spin lock is interrupted by an interrupt handler trying to acquire the same spin lock? Is it possible to avoid such a situation ? How ?
2. If your code needs to sleep, which is often the case when synchronizing with user-space: Explain which synchronization method is most suitable for the above scenario?
3. List out the various aspects of scalability and contention.

Course Outcome 4 (CO4):

1. Differentiate vmalloc() and kmalloc()? Which of these must be used for allocating memory for hardware devices? Why?
2. Explain the design of the slab layer. Also explain the advantage of slab layer allocation of kernel objects.
3. Illustrate the importance of zones in Linux memory management.

Course Outcome 5 (CO5):

1. Before the 2.6 kernel, buffer head was used for block I/O whereas in the later kernels, it has been replaced by biostructure as the basic container for block I/O. Why?
2. Discuss the significance of an I/O scheduler. What drawback of the deadline I/O scheduler is overcome by the anticipatory I/O scheduler?
3. Describe the measures to avoid data alignment issues.

Course Outcome 6 (CO6):

1. Design and implement a process scheduler.

Syllabus

Module 1: Process Management and Scheduling

Introduction to the Linux kernel, Process Management – Process, Process descriptor and the task structure, Process creation, The Linux implementation of threads, Process termination. Process Scheduling – Multitasking, Linux's process scheduler, Policy, Linux scheduling algorithm, Preemption and context switching, Real-time scheduling policies.

Module 2: System calls and Interrupts

System Calls - Communicating with the Kernel, Syscalls, System call handler, System call implementation, System call context . Interrupts and Interrupt Handlers – Interrupts, Interrupt handlers, Top halves versus bottom halves, Registering an interrupt handler, Writing an interrupt handler, Interrupt context, Interrupt control. Bottom Halves – Task queues, Softirqs, Tasklets, Work queues.

Module 3: Kernel Synchronization

Kernel Synchronization – Critical regions and race conditions, Locking, Deadlocks, Contention and scalability. Kernel Synchronization Methods – Atomic operations, Spin locks, Semaphores, Mutexes, Completion variables, BKL: The Big Kernel Lock, Sequential locks, Preemption disabling.

Module 4: Memory Management and Virtual File System

Memory Management – Pages, Zones, kmalloc(), kfree(), vmalloc(), Slab layer – design, Per-CPU allocations. The Virtual File system – VFS objects, data structures, relationship and functionalities.

Module 5: Block I/O Layer and Portability

The Block I/O Layer – Buffers and buffer heads, Request queues, I/O schedulers – Types, Scheduler selection. Portability – Word size and data types, Data alignment, Byte order, Time, Processor ordering

Course Plan

No	Topic	No. of Lectures (40 hrs)
1	**Module 1: Process Management and Scheduling**	**8**
1.1	Introduction to the Linux kernel , Process	1
1.2	Process descriptor and the task structure	1
1.3	Process creation, The Linux implementation of threads	1
1.4	Process termination, Multitasking	1
4.5	Linux's process scheduler, Policy	1
1.6	Linux scheduling algorithm	1
1.7	Preemption and context switching	1
1.8	Realtsime scheduling policies	1
2	**Module 2: System Calls and Interrupts**	9
2.1	Communicating with the Kernel, Syscalls	1
2.2	System Call Handler, System Call Implementation	1
2.3	System call context	1
2.4	Interrupts, Interrupt handlers, Top halves versus bottom halves	1
2.5	Registering an Interrupt Handler	1
2.6	Writing an Interrupt Handler	1
2.7	Interrupt Context, Interrupt Control	1
2.8	Bottom Halves – Task Queues, Softirqs	1
2.9	Tasklets, Work Queues	1
3	**Module 3: Kernel Synchronization**	9
3.1	Critical regions and race conditions	1
3.2	Locking, Deadlocks	1
3.3	Contention and scalability	1
3.4	Kernel Synchronization Methods – Atomic operations	1
3.5	Spin locks	1
3.6	Semaphores	1
3.7	Mutexes, Completion variables	1
3.8	BKL: The Big Kernel Lock	1
3.9	Sequential locks, Preemption disabling	1
4	**Module 4: Memory Management and Virtual File System**	7
4.1	Pages, Zones	1
4.2	kmalloc()	1
4.3	kfree(), vmalloc()	1
4.4	Slab Layer - Design	1
4.5	Per-CPU Allocations	1
4.6	The Virtual File system – VFS objects	1
4.7	Data structures, relationship and functionalities	1
5	**Module 5 : The Block I/O Layer**	7

5.1	Buffers and buffer heads	1
5.2	Request queues	1
5.3	I/O Schedulers	1
5.4	Types, Scheduler Selection	1
5.5	Portability – Word size and data types	1
5.6	Data Alignment, Byte Order	1
5.7	Time, Processor ordering	1

Reference

1. Robert Love, “Linux Kernel Development”, 3/e, Addison-Wesley, 2010.
2. Daniel Bovet, Marco Cesati, “Understanding the Linux Kernel”, 3/e, OReilly Media Inc., 2005.
3. Linux Kernel Architecture – Wolfgang Mauerer.
4. Reilly Christian Benvenuti, “Understanding Linux Network Internals”, 1/e, OReilly Media Inc.,2005.
5. Jonathan Corbet, Alessandro Rubini, Greg Kroah-Hartman, “Linux Device Drivers”, 3/e, OReilly Media Inc., 2005
6. Operating Systems Concepts, 9th Edition- Silberschatz, Galvin, Gagne

COURSE CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
R25CSP203	**MINI PROJECT**	**PROJECT**	**0**	**0**	**4**	**2**

Mini project can help to strengthen the understanding of student's fundamentals through application of theoretical concepts and to boost their skills and widen the horizon of their thinking. The ultimate aim of an engineering student is to resolve a problem by applying theoretical knowledge. Doing more projects increases problem solving skills.

The introduction of mini projects ensures preparedness of students to undertake dissertation. Students should identify a topic of interest in consultation with PG Programme Coordinator that should lead to their dissertation/research project. Demonstrate the novelty of the project through the results and outputs.The progress of the mini project is evaluated based on three reviews, two interim reviews and a final review. A report is required at the end of the semester.

Evaluation Committee - Programme Coordinator, One Senior Professor and Guide.

Sl. No	Type of evaluations	Mark	Evaluation criteria
1	Interim evaluation 1	20	
2	Interim evaluation 2	20	
3	Final evaluation by a Committee	35	Will be evaluating the level of completion and demonstration of functionality/ specifications, clarity of presentation, oral examination, work knowledge and involvement
4	Report	15	the committee will be evaluating for the technical content, adequacy of references, templates followed and permitted plagiarism level(not more than 25%)
5	Supervisor/Guide	10	
	Total Marks	100	

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
R25CSP207	**COMPUTING LAB 2**	**LABORATORY 2**	**0**	**0**	**2**	**1**

Preamble: This course provides an in-hand experience related to various database management systems. Also equips them to design and implement a database application built over the concepts. This course helps the learners to develop applications that manage data efficiently with the help of suitable data models and techniques.

Course Outcomes: After the completion of the course the student will be able to

CO#	Course Outcomes
CO1	Able to perform the distributed processing of large data sets across clusters of computers using simple programming models with the help of Hadoop. **(Cognitive Knowledge Level: Apply)**
CO2	Perform data summarization and ad hoc querying using Hive. **(Cognitive Knowledge Level: Create)**
CO3	Operates on document databases and techniques using DynamoDB. **(Cognitive Knowledge Level: Apply)**
CO4	Capable of implementing XML and XML queries for data management. **(Cognitive Knowledge Level: Create)**
CO5	Apply emerging technologies in column store along with Cassandra. **(Cognitive Knowledge Level: Apply)**

Program Outcomes (PO)

Outcomes are the attributes that are to be demonstrated by a graduate after completing the course.

PO1: An ability to independently carry out research/investigation and development work in engineering and allied streams

PO2: An ability to communicate effectively, write and present technical reports on complex engineering activities by interacting with the engineering fraternity and with society at large.

PO3: An ability to demonstrate a degree of mastery over the area as per the specialization of the program. The mastery should be at a level higher than the requirements in the appropriate bachelor program

PO4: An ability to apply stream knowledge to design or develop solutions for real world problems by following the standards

PO5: An ability to identify, select and apply appropriate techniques, resources and state-of-the-art tool to model, analyse and solve practical engineering problems.

PO6: An ability to engage in life-long learning for the design and development related to the stream related problems taking into consideration sustainability, societal, ethical and environmental aspects

PO7: An ability to develop cognitive load management skills related to project management and finance which focus on Entrepreneurship and Industry relevance.

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7
CO1	2	2	2	3	3	2	2
CO2	2	2	2	3	3		
CO3	2	2	2	3	3		
CO4	2	2	2	3	3	2	
CO5	2	2	2	3	3		

Continuous Internal Evaluation Pattern:

The laboratory courses will be having only Continuous Internal Evaluation and carries 100 marks. Final assessment shall be done by two examiners; one examiner will be a senior faculty from the same department.

Continuous Evaluation: 60 marks

Final internal assessment: 40 marks

Lab Report:

All the students attending the Lab should have a Fair Report. The report should contain details of experiments such as Objective, Algorithm/Design, Description, Implementation, Analysis, Results, and Outcome. The report should contain a print out of the respective code with inputs addressing all the aspects of the algorithm described and corresponding outputs. All the experiments noted in the fair report should be verified by the faculty regularly. The fair report, properly certified by the faculty, should be produced during the time of the final assessment.

Syllabus

Basic concepts of Big Data, Configuration of Hadoop, Parallel Database, Distributed Database, Semi-structured Data and XML Databases: XML Data Model – XML Schema- DTD-XSD – XPath and XQuery, JDOQL (Java Data Object-based Query Language), No SQL Databases: Key value stores – DynamoDB, Column Based- HBase, Cassandra, Web Page ranking algorithm.

Practice Questions

1. Study and Configure Hadoop for Big Data.
2. Study of NoSQL Databases such as Hive/HBase/Cassandra/DynamoDB.
3. Design Data Model using NoSQL Databases such as Hive/HBase/Cassandra/DynamoDB.
4. Implement any one Partitioning technique in Parallel Databases.
5. Implement Two Phase commit protocol in Distributed Databases.
6. Design Persistent Objects using JDO and implement min 10 queries on objects using JDOQL in Object DB NOSQL DATABASE.
7. Create XML, XML schemas, DTD for any database application and implement min 10 queries using XQuery FLOWR expression and XPath.
8. Design database schemas and implement min 10 queries using Hive/ HBase/ Cassandra column-based databases.
9. Design database schemas and implement min 10 queries using Dynamo DB key-value based databases.
10. Implement Web Page ranking algorithm.
11. Create a database infrastructure like GitHub backend DB for a project team to collaborate for coding, code review, code commenting and approval, code rejection for changes workflow.

Reference Books:

1. Tom White, Hadoop: The Definitive Guide, O'Reilly Media 4th Edition, April 2015.
2. Joe Fawcett, Danny Ayers, Liam R. E. Quin, Beginning XML, 5/e, John Wiley & Sons, 2012.
3. Jeff Carpenter, Eben Hewitt, “Cassandra: The Definitive Guide”, 3/e, O’Reilly, 2020.
4. Tanmay Deshpande, “DynamoDB Cookbook”, Packt Publishing, September 2015.

SEMESTER II

PROGRAM ELECTIVE III

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
R25CST2X4	**BIG DATA ANALYTICS**	**PROGRAM ELECTIVE 3**	3	0	0	3

Preamble: This course helps the learner to understand the basic concepts of data analytics. This course covers mathematics for data analytics, Big data and its applications, techniques for managing big data and data analysis & visualization using R programming tool. It enables the learners to perform data analysis on a real-world scenario using appropriate tools.

Prerequisite: Nil

Course Outcomes: After the completion of the course the student will be able to

CO 1	Illustrate the concept for data analytics using basic mathematics **(Cognitive Knowledge Level: Apply)**
CO 2	Illustrate the concepts of data analytics **(Cognitive Knowledge Level: Apply)**
CO 3	Apply various Big Data Analytics Techniques using R **(Cognitive Knowledge Level: Apply)**
CO 4	Access and Process Data on Distributed File System and to Manage Job Execution in Hadoop Environment **(Cognitive Knowledge Level: Apply)**
CO 5	Analyze the Big Data using Advanced analytical methods such as text analysis **(Cognitive Knowledge Level: Apply)**
CO 6	Design, Develop, Implement and Present innovative ideas on distributed algorithms and techniques. **(Cognitive Knowledge Level: Create)**

Program Outcomes (PO)

Outcomes are the attributes that are to be demonstrated by a graduate after completing the course.

PO1: An ability to independently carry out research/investigation and development work in engineering and allied streams

PO2: An ability to communicate effectively, write and present technical reports on complex engineering activities by interacting with the engineering fraternity and with society at large.

PO3: An ability to demonstrate a degree of mastery over the area as per the specialization of the program. The mastery should be at a level higher than the requirements in the appropriate bachelor program

PO4: An ability to apply stream knowledge to design or develop solutions for real world problems by following the standards

PO5: An ability to identify, select and apply appropriate techniques, resources and state-of-the-art tool to model, analyse and solve practical engineering problems.

PO6: An ability to engage in life-long learning for the design and development related to the stream related problems taking into consideration sustainability, societal, ethical and environmental aspects

PO7: An ability to develop cognitive load management skills related to project management and finance which focus on Entrepreneurship and Industry relevance.

Mapping of course outcomes with program outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7
CO 1	2		3	2	2	1	
CO 2	2		3	2	2	1	
CO 3	2		3	3	3	1	
CO 4	2		3	3	3	1	
CO 5	3		3	3	3	2	
CO 6	3	2	3	3	3	2	2

Assessment Pattern

Bloom's Category	End Semester Examination
Apply	70%-80%
Analyze	30%-40%
Evaluate	
Create	

Mark distribution

Total Marks	CIE	ESE	ESE Duration
100	40	60	2.5 hours

Continuous Internal Evaluation Pattern:

Evaluation shall only be based on application, analysis or design-based questions (for both internal and end semester examinations).

Continuous Internal Evaluation: 40 marks

i. Preparing a review article based on peer reviewed original publications (minimum 10 publications shall be referred) : 15 marks

ii. Course based task / Seminar/ Data collection and interpretation : 15 marks

iii. Test paper (1 number) : 10 marks

Test paper shall include minimum 80% of the syllabus.

Course based task/test paper questions shall be useful in the testing of knowledge, skills, comprehension, application, analysis, synthesis, evaluation and understanding of the students.

End Semester Examination Pattern:

The end semester examination will be conducted by the respective College.
There will be two parts; Part A and Part B.
Part A will contain 5 numerical/short answer questions with 1 question from each module, having 5 marks for each question. Students should answer all questions. Part B will contain 7 questions (such questions shall be useful in the testing of overall achievement and maturity of the students in a course, through long answer questions relating to theoretical/practical knowledge, derivations, problem solving and quantitative evaluation), with minimum one question from each module of which student should answer any five. Each question can carry 7 marks
Total duration of the examination will be 150 minutes.
Note: The marks obtained for the ESE for an elective course shall not exceed 20% over the average ESE mark % for the core courses. ESE marks awarded to a student for each elective course shall be normalized accordingly.
For example if the average end semester mark % for a core course is 40, then the maximum eligible mark % for an elective course is 40+20 = 60 %.

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Data Science has a big role to play than BI in big data analytics. Give your view.
2. The number of members of a millionaires' club were as follows:

Year	2011	2012	2013	2014	2015	2016

Members	23	24	27	25	30	28

i. What is the average growth rate of the membership?

ii. Based on the results of (i), how many members would one expect in 2018?

3. A On-line store has integrated Big Data across multiple IT systems to gather customer transaction and interactions data in order to predict customer recommendations better. Provide your view on how BI and Data science handles the recommendations.

Course Outcome 2 (CO2):

1. Assume you have more than two populations to consider? Which hypothesis testing method would you adopt. State the initial and alternate hypothesis for the same, How to accept/reject these hypotheses?
2. Differentiate between type-1 error and type-2 error. Do you think one is always more serious than the other. Justify.
3. How various t-tests can be done in R? Give the R functions for the same and their usage.

Course Outcome 3(CO3):

1. With a neat sketch illustrate the architecture diagram of Hadoop File System.
2. Illustrate the working of a Map Reduce program with example.
3. Explain the Avro and File-Based Data structures.

Course Outcome 4 (CO4):

1. List and explain various Hadoop ecosystem components.
2. Explain the data analytic architecture with a diagram.
3. Give a code in Hive which illustrate the functions to read and write data in to a database.

Course Outcome 5 (CO5):

1. Given a movie review dataset. Propose a technique by which sentiments can be analysed from rated movie reviews.
2. Which numerical statistic is intended to reflect how important a 'word' is to a document in a collection or corpus? Justify your answer and sort out the same using the following example.

   Given a document containing 100 words wherein the word ‘cat’ appears 3 times. Assume there are ‘10 million’ documents and the word ‘cat’ appears in ‘one thousand’ of these.

Course Outcome 6 (CO6):

1. Assume that a non-stationary time series with seasonal fluctuations of last 100 months is given. Applying various methods illustrate how prediction can be done from this series for next 12 months.

Syllabus

Module 1: Introduction to Data Analytics

Mathematics for Data Analytics - Descriptive statistics - Measures of central tendency and dispersion, Association of two variables, Probability calculus - probability distributions, Inductive statistics - Point estimation, Interval estimation, Hypothesis Testing - Basic definitions, t-test

Introduction To Big Data: Big data characteristics, Features of Big Data, Evolution of Big data, Analyst Perspective on Data Repositories , State of the Practice in Analytics, BI Versus Data Science.

Module 2: Introduction to R

Review of basic data analytic methods using R : Introduction to R, R graphical user interface-data import and export-attribute and data type. Exploratory data analysis-Visualization, Dirty data, single and multiple variables, data exploration vs presentation. Statistical methods for evaluation-Hypothesis testing, difference of means, Wilcoxon rank sum test, type I and II errors, power and sample size,

Module 3: Hadoop & HDFS

History of Hadoop, Apache Hadoop, Analysing Data with Unix tools, Analysing Data with Hadoop, Hadoop Streaming, Hadoop Echo System, IBM Big Data Strategy, Introduction to Infosphere BigInsights and Big Sheets.

The Design of HDFS, HDFS Concepts, Command Line Interface, Hadoop file system interfaces, Data flow, Data Ingest with Flume and Scoop and Hadoop archives, Hadoop I/O: Compression, Serialization, Avro and File-Based Data structures

Module 4: Map reduce & Hadoop Ecosystem

Anatomy of a Map Reduce Job Run, Failures, Job Scheduling, Shuffle and Sort, Task Execution, Map Reduce Types and Formats, Map Reduce Features.

Pig : Introduction to PIG, Execution Modes of Pig, Comparison of Pig with Databases, Grunt, Pig Latin, User Defined Functions, Data Processing operators. Hive : Hive Shell, Hive Services, Hive Metastore, Comparison with Traditional Databases, HiveQL.

Module 5: Advanced Analytical Theory, Methods and Text Analysis

Advanced analytical theory and methods: Time Series Analysis- Overview of Time Series Analysis, Box-Jenkins Methodology ARIMA Model, Autocorrelation Function (ACF), Autoregressive Models, Moving Average Models ARMA and ARIMA Models Building and Evaluating an ARIMA Model, Reasons to Choose and Cautions.

Course Plan

No	Topic	No. of Lectures (40 hrs)
1	**Module 1: Introduction to Data Analytics**	**8**
1.1	Mathematics for Data Analytics - Descriptive statistics	1
1.2	Measures of central tendency and dispersion, Association of two variables	1
1.3	Probability calculus - probability distributions, Inductive statistics	1
1.4	Point estimation, Interval estimation, Hypothesis Testing - Basic definitions, t-test	1
1.5	Introduction To Big Data: Big data characteristics	1
1.6	Features of Big Data,	1
1.7	Evolution of Big data, Analyst Perspective on Data Repositories	1
1.8	State of the Practice in Analytics BI Versus Data Science	1
2	**Module 2: Introduction to R**	8
2.1	Review of basic data analytic methods using R: Introduction to R	1
2.2	R graphical user interface-data import and export-attribute and data type	1
2.3	Exploratory data analysis-Visualization	1
2.4	Dirty data, single and multiple variables, data exploration vs presentation.	1
2.5	Statistical methods for evaluation	1
2.6	Hypothesis testing, Difference of means.	1
2.7	Wilcoxon rank sum test, type I and II errors	1
2.8	Power and sample size.	1
3	**Module 3: Hadoop & HDFS**	**8**
3.1	History of Hadoop, Apache Hadoop	1
3.2	Analysing Data with Unix tools, Analysing Data with Hadoop	1
3.3	Hadoop Streaming	1
3.4	Hadoop Echo System, IBM Big Data Strategy	1
3.5	Introduction to Infosphere Big Insights and Big Sheets.	1
3.6	The Design of HDFS, HDFS Concepts, Command Line Interface	1

3.7	Hadoop file system interfaces, Data flow, Data Ingest with Flume and Scoop and Hadoop archives	1
3.8	Hadoop I/O: Compression, Serialization, Avro and File-Based Data structures	1
4	**Module 4: Map reduce & Hadoop Ecosystem**	**8**
4.1	Anatomy of a Map Reduce Job Run	1
4.2	Failures, Job Scheduling.	1
4.3	Shuffle and Sort, Task Execution	1
4.4	Map Reduce Types and Formats, Map Reduce Features.	1
4.5	Pig: Introduction to PIG, Execution Modes of Pig	1
4.6	Comparison of Pig with Databases, Grunt, Pig Latin	1
4.7	User Defined Functions, Data Processing operators.	1
4.8	Hive: Hive Shell, Hive Services, Hive Metastore, Comparison with Traditional Databases, HiveQL.	1
5	**Module 5: Advanced Analytical Theory, Methods and Text Analysis**	**8**
5.1	Advanced analytical theory and methods: Time Series Analysis	1
5.2	Overview of Time Series Analysis.	1
5.2	Box-Jenkins Methodology ARIMA Model	1
5.3	Autocorrelation Function (ACF)	1
5.4	Autoregressive Models	1
5.5	Moving Average Models	1
5.6	ARMA and ARIMA Models Building	1
5.7	Evaluating an ARIMA Model	1
5.8	Reasons to Choose and Cautions	1

References

1. Christian Heumann and Michael Schomaker, “Introduction to Statistics and DataAnalysis”, Springer, 2016
2. David Dietrich, Barry Heller, Biebie Yang, Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data, EMC Education Services, John Wiley &Sons, Inc
3. Jaiwei Han, MichelineKamber, “Data Mining Concepts and Techniques", Elsevier, 2006.
4. Tom White “ Hadoop: The Definitive Guide” Third Edit on, O’reily Media, 2012.
5. Seema Acharya, Subhasini Chellappan, "Big Data Analytics" Wiley 2015

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
R25CST2X4	WIRELESS SENSOR NETWORKS	PROGRAM ELECTIVE 3	3	0	0	3

Preamble: This course gives an introduction to wireless sensor networks, which has helped to develop a vast array of applications in diverse areas like agriculture, military, health and civil infrastructures. The course deals with hardware aspects like sensors with their built in transceivers and associated electronics as well as the softwares needed to run them. The MAC layer with its myriad protocols, as well as routing layer protocols, are discussed in the course. Localization of sensor nodes, deployment and coverage, as well as security of wireless sensor networks also form part of the course. This course enables the learners to design and develop wireless sensor protocols and applications.

Course Outcomes: After the completion of the course the student will be able to

CO 1	List the applications, hardware and software components of wireless sensor networks, and the challenges faced in design of sensor networks. (**Cognitive knowledge: Analyze**)
CO 2	Apply design principles and formulate necessary service interfaces while designing a wireless sensor network **(Cognitive knowledge: Apply)**
CO 3	Design MAC protocols for wireless sensor networks taking into account the specific requirements of the network. (**Cognitive knowledge: Create**)
CO 4	Explain localization techniques, coverage problem and security issues in wireless sensor networks (**Cognitive knowledge: Evaluate**)
CO 5	Design energy efficient routing protocols for wireless sensor networks (**Cognitive knowledge**: **Create**)

Program Outcomes (PO)

Outcomes are the attributes that are to be demonstrated by a graduate after completing the course.

PO1: An ability to independently carry out research/investigation and development work in engineering and allied streams

PO2: An ability to communicate effectively, write and present technical reports on complex engineering activities by interacting with the engineering fraternity and with society at large.

PO3: An ability to demonstrate a degree of mastery over the area as per the specialization of the program. The mastery should be at a level higher than the requirements in the appropriate bachelor program

PO4: An ability to apply stream knowledge to design or develop solutions for real world problems by following the standards

PO5: An ability to identify, select and apply appropriate techniques, resources and stat e - of the-art tools to model, analyse and solve practical engineering problems.

PO6: An ability to engage in life-long learning for the design and development related to the stream related problems taking into consideration sustainability, societal, ethical and environmental aspects

PO7: An ability to develop cognitive load management skills related to project management and finance which focus on Entrepreneurship and Industry relevance.

Mapping of course outcomes with program outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7
CO 1	2		3	2		1	
CO 2	2		3	3	2	1	
CO 3	3		3	3	3	2	
CO 4	2		3	2	2	2	
CO 5	3	2	3	3	3	2	2

Assessment Pattern

Bloom's Category	End Semester Examination
Apply	70%-80%
Analyze	30%-40%
Evaluate	
Create	

Mark distribution

Total Marks	CIE	ESE	ESE Duration
100	40	60	2.5 hours

Continuous Internal Evaluation Pattern:

Evaluation shall only be based on application, analysis or design based questions (for both internal and end semester examinations).

Continuous Internal Evaluation: 40 marks

i. Preparing a review article based on peer reviewed original publications (minimum 10 publications shall be referred) : 15 marks

ii. Course based task / Seminar/ Data collection and interpretation : 15 marks
iii. Test paper (1 number) : 10 marks

Test paper shall include minimum 80% of the syllabus.

Course based task/test paper questions shall be useful in the testing of knowledge, skills, comprehension, application, analysis, synthesis, evaluation and understanding of the students.

End Semester Examination Pattern:

The end semester examination will be conducted by the respective College.

There will be two parts; Part A and Part B.

Part A will contain 5 numerical/short answer questions with 1 question from each module, having 5 marks for each question. Students should answer all questions. Part B will contain 7 questions (such questions shall be useful in the testing of overall achievement and maturity of the students in a course, through long answer questions relating to theoretical/practical knowledge, derivations, problem solving and quantitative evaluation), with minimum one question from each module of which student should answer any five. Each question can carry 7 marks Total duration of the examination will be 150 minutes.

Note: The marks obtained for the ESE for an elective course shall not exceed 20% over the average ESE mark % for the core courses. ESE marks awarded to a student for each elective course shall be normalized accordingly.
For example if the average end semester mark % for a core course is 40, then the maximum eligible mark % for an elective course is 40+20 = 60 %.

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Enumerate the challenges faced while designing a wireless sensor network.
2. List the important hardware elements of a wireless sensor and explain the function of each.
3. What are the challenges faced in sensor network programming?

Course Outcome 2 (CO2):

1. Examine the design principles that should be followed while designing a wireless sensor network.
2. Evaluate the performance of IEEE 802.11 DCF using Markov chain.
3. Illustrate the functionalities that a service interface should provide when interfacing an application to a protocol stack.

Course Outcome 3(CO3):

1. Analyze any three techniques which MAC protocols use to reduce energy consumption.
2. Differentiate between contention free and schedule based MAC protocols.
3. Design a schedule based MAC protocol.

Course Outcome 4 (CO4):

1. Propose a method to find the best coverage path between two nodes in a wireless sensor network.
2. List the types of security attacks that can occur in WSNs.

Course Outcome 5 (CO5):

1. What is data-centric routing? Why is data-centric routing more feasible compared to routing based on addresses?
2. Under what circumstances GPSR enters the perimeter routing mode ?
3. Design a routing protocol for a wireless sensor network that supports an application with.

Syllabus

Module 1: Introduction to Wireless Sensor Networks

Motivation, Challenges and Constraints, Applications.
Node Architecture – Hardware elements, Sensors and Actuators, Power supply, Energy Consumption of sensor nodes.
Challenges in sensor network programming, Operating systems and execution environments- embedded OS, issues, Programming models.

Module 2: Sensor Network Architecture

Sensor network scenarios, Optimization goals and figures of merit. Design principles for WSNs, Service interfaces of WSNs. Gateway concepts, WSN- Internet Communication. Discrete time Markov Chain, Performance analysis of IEEE 802.11 DCF using Markov Chain

Module 3: MAC Protocols for WSNs

Characteristics and design goals, low duty cycle protocols and wakeup concepts. Contention based protocols: PAMAS, STEM, T-Meschede based protocols: SMACS, TRAMA, Hybrid MAC protocols: Z-MAC

Case Studies: S-MAC, 802.15.4, 802.15.6

Module 4: Routing Protocols for WSNs

Introduction, Routing challenges and design issues, Routing Strategies, Routing Techniques: Flooding and Gossiping, SPIN, LEACH, PEGASIS, Directed Diffusion, Rumour Routing, Geographic Routing- Forwarding Strategies.

Module 5: Localization, Coverage and Security in WSNs

Localization: approaches-proximity-trilateration and triangulation- scene analysis Coverage and deployment: sensingmodels, coverage measures, uniform random deployments, coverage determination. Security: Security challenges in WSNs, Security attacks in WSNs

Course Plan

No	Topic	No. of Lectures (40)
1	**Module 1: Introduction to Wireless Sensor Networks**	**9**
1.1	Motivation, Challenges and Constraints	1
1.2	Applications	1
1.3	Node Architecture	1
1.4	Hardware elements	1
1.5	Sensors and Actuators, Power supply	1
1.6	Energy Consumption of sensor nodes	1
1.7	Challenges in sensor network programming	1
1.8	Operating systems and execution environments- embedded OS, issues	1
1.9	Programming models	1
2	**Module 2: Sensor Network Architecture**	**7**
2.1	Sensor network scenarios	1
2.2	Optimization goals and figures of merit	1
2.3	Design principles for WSNs	1
2.4	Service interfaces of WSNs	1
2.5	Gateway concepts, WSN- Internet Communication	1
2.6	Discrete time Markov Chain	1
2.7	Performance analysis of IEEE 802.11 DCF using Markov Chain	1
3	**Module 3: MAC Protocols for WSNs**	**8**
3.1	Characteristics and design goals	1
3.2	Low duty cycle protocols and wakeup concepts	1
3.3	Contention based protocols: PAMAS	1
3.4	STEM, T-MAC	1
3.5	Schedule based protocols: SMACS TRAMA	1
3.6	Hybrid MAC protocols: Z-MAC	1
3.7	Case Studies: S-MAC	1
3.8	802.15.4, 802.15.6	1
4	**Module 4: Routing Protocols for WSNs**	**8**
4.1	Introduction, Routing challenges and design issues	1
4.2	Routing Strategies	1
4.3	Routing Techniques: Flooding and Gossiping, SPIN	1
4.4	LEACH, PEGASIS	1
4.5	Directed Diffusion	1

4.6	Rumour Routing	1
4.7	Geographic Routing	1
4.8	Forwarding Strategies	1
5	**Module 5: Localization, Coverage and Security in WSNs**	**8**
5.1	Localization: approaches-proximity-trilateration	1
5.2	Triangulation, scene analysis	1
5.3	Coverage and deployment: sensing models	1
5.4	Coverage measures	1
5.5	Uniform random deployments	1
5.6	Coverage determination	1
5.7	Security: Security challenges in wireless sensor networks	1
5.8	Security attacks in wireless sensor networks	1

References

1. W. Dargie and C. Poellabauer, Fundamentals of Wireless Sensor Networks, Theory and Practice,Wiley, 2010
2. Holger Karl and Andreas Willig, Protocols and Architectures for Wireless Sensor Networks,Wiley, 2005
3. K. Sohraby, D. Minoli and T. Znati, Wireless Sensor Networks, Technology, Protocols, and Applications,Wiley-Interscience, 2007
4. C. Siva Ram Murthy and B. S Manoj, Adhoc Wireless Networks Architectures and Protocols,Prentice Hall, 2004
5. Feng Zhao and Leonidas Guibas, Wireless Sensor Networks An Information Processing Approach, Morgan Kaufman, 2005

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
R25CST2X4	DEEP LEARNING	PROGRAM ELECTIVE 3	3	0	0	3

Preamble: Study of this course provides the learners an overview of the concepts and algorithms involved in deep learning. The course covers the basic concepts in deep learning, optimization techniques, regularization techniques, convolutional neural networks, recurrent neural networks, graphical models, deep generative models. This course helps the students to implement deep learning algorithms to solve real-world problems.

Prerequisite : Nil

Course Outcomes: After the completion of the course the student will be able to

CO 1	Use the standard regularization and optimization techniques for the effective training of deep neural networks. **(Cognitive Knowledge Level: Apply)**
CO 2	Build convolutional Neural Network (CNN) models for different use cases. **(Cognitive Knowledge Level: Apply)**
CO 3	Apply the concepts of Recurrent Neural Network (RNN), Long Short Term Memory(LSTM), Gated Recurrent Unit (GRU) for solving problems. **(Cognitive Knowledge Level:Apply)**
CO 4	Construct Bayesian networks, Markov networks and apply computational techniques to draw inferences. **(Cognitive Knowledge Level: Apply)**
CO 5	Illustrate the concepts of auto encoder, sampling algorithms, deep generative models and transfer learning. **(Cognitive Knowledge Level: Apply)**
CO 6	Design, develop, implement and present innovative ideas on deep learning concepts and techniques to solve real-world problems. **(Cognitive Knowledge Level: Create)**

Program Outcomes (PO)

Outcomes are the attributes that are to be demonstrated by a graduate after completing the course.

PO1: An ability to independently carry out research/investigation and development work in engineering and allied streams

PO2: An ability to communicate effectively, write and present technical reports on complex engineering activities by interacting with the engineering fraternity and with society at large.

PO3: An ability to demonstrate a degree of mastery over the area as per the specialization of the program. The mastery should be at a level higher than the requirements in the appropriate bachelor program

PO4: An ability to apply stream knowledge to design or develop solutions for real world problems by following the standards

PO5: An ability to identify, select and apply appropriate techniques, resources and state-of-the-art tool to model, analyse and solve practical engineering problems.

PO6: An ability to engage in life-long learning for the design and development related to the stream related problems taking into consideration sustainability, societal, ethical and environmental aspects

PO7: An ability to develop cognitive load management skills related to project management and finance which focus on Entrepreneurship and Industry relevance.

Mapping of course outcomes with program outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7
CO 1	3		2	3	2	2	
CO 2	3		2	3	2	2	
CO 3	3		2	3	2	2	
CO 4	3		2	3	2	2	
CO 5	3		2	2	2	2	
CO 6	3	2	2	3	2	2	2

Assessment Pattern

Bloom's Category	End Semester Examination
Apply	70%-80%
Analyze	30%-40%
Evaluate	
Create	

Mark distribution

Total Marks	CIE	ESE	ESE Duration
100	40	60	2.5 hours

Continuous Internal Evaluation Pattern:

Evaluation shall only be based on application, analysis or design-based questions (for both internal and end semester examinations).

Continuous Internal Evaluation: 40 marks

i. Preparing a review article based on peer reviewed original publications (minimum 10 publications shall be referred) : 15 marks

ii. Course based task / Seminar/ Data collection and interpretation : 15 marks

iii. Test paper (1 number) : 10 marks

Test paper shall include minimum 80% of the syllabus.

Course based task/test paper questions shall be useful in the testing of knowledge, skills, comprehension, application, analysis, synthesis, evaluation and understanding of the students.

End Semester Examination Pattern:

The end semester examination will be conducted by the respective College.
There will be two parts; Part A and Part B.
Part A will contain 5 numerical/short answer questions with 1 question from each module, having 5 marks for each question. Students should answer all questions. Part B will contain 7 questions (such questions shall be useful in the testing of overall achievement and maturity of the students in a course, through long answer questions relating to theoretical/practical knowledge, derivations, problem solving and quantitative evaluation), with minimum one question from each module of which student should answer any five. Each question can carry 7 marks
Total duration of the examination will be 150 minutes.
Note: The marks obtained for the ESE for an elective course shall not exceed 20% over the average ESE mark % for the core courses. ESE marks awarded to a student for each elective course shall be normalized accordingly.
For example, if the average end semester mark % for a core course is 40, then the maximum eligible mark % for an elective course is 40+20 = 60 %.

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Write an algorithm for backpropagation which uses stochastic gradient descent method. Comment on the effect of adding momentum to the network.
2. You are doing full batch gradient descent using the entire training set (not stochastic gradient descent). Is it necessary to shuffle the training data? Explain your answer.
3. You would like to train a dog/cat image classifier using mini-batch gradient descent. You have already split your dataset into train, validation and test sets. The classes are balanced. You realize that within the training set, the images are ordered in such a way that all the dog images come first and all the cat images come after. A friend tells you: "you absolutely need to shuffle your training set before the training procedure." Is your friend right? Explain.

4. Explain how L1 regularization method leads to weight sparsity.
5. Data augmentation is often used to increase the amount of data you have. Should you apply data augmentation to the test set? Explain why.

Course Outcome 2 (CO2):

1. You are given a dataset of 10 x 10 grayscale images. Your goal is to build a 5-class classifier. Explain which one of the following two options you would choose and why ? a) the input is flattened into a 100-dimensional vector, followed by a fully-connected layer with 5 neurons, b) the input is directly given to a convolutional layer with five 10 x 10 filters.

2. Weight sharing allows CNNs to deal with image data without using too many parameters. Does weight sharing increase the bias or the variance of a model? Explain

3. A convolutional neural network has 4 consecutive layers as follows: 3 x 3 conv (stride 2) - 2 x 2 Pool - 3 x 3 conv (stride 2) - 2 x 2 Pool. Determine how large is the set of image pixels which activate a neuron in the 4th non-image layer of this network?

4. Consider the convolutional neural network defined by the layers in the left column below. Determine the shape of the output volume and the number of parameters at each layer. You can write the activation shapes in the format (H, W, C), where H, W, C are the height, width and channel dimensions, respectively. Unless specified, assume padding 1, stride 1 where appropriate.

   Notation:

- CONVx-N denotes a convolutional layer with N filters with height and width equal to x.
- POOL-n denotes a n×n max-pooling layer with stride of n and 0 padding.
- FLATTEN flattens its inputs.
- FC-N denotes a fully-connected layer with N neurons.

Layer	Activation Volume Dimensions	Number of parameters
Input	32 × 32 × 3	0
CONV3-8		
Leaky ReLU		
POOL-2		
BATCHNORM		
CONV3-16		

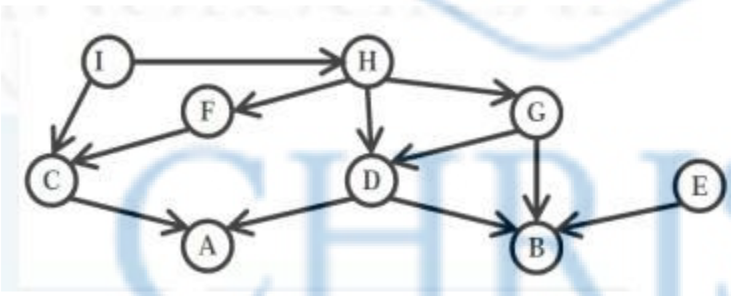
Leaky ReLU		
POOL-2		
FLATTEN		
FC-10		

Course Outcome 3(CO3):

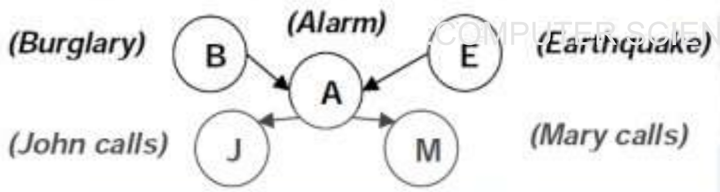
1. Illustrate the workings of the RNN with an example of a single sequence defined on a vocabulary of four words.
2. List the differences between LSTM and GRU
3. Explain your understanding of unfolding a recursive or recurrent computation into a computational graph.
4. The vanishing gradient problem is more pronounced in RNN than in traditional neural networks. Give reason. Discuss a solution for the problem.
5. Show the steps involved in an LSTM to predict stock prices. Give one advantage of using an RNN rather than a convolutional network.

Course Outcome 4 (CO4):

1. Construct the Bayesian Network that corresponds to this conditional probability: P(A | B,C,E) P(B | D,E) P(C | F,H) P(D | G) P(E| G,H) P(F | H) P(G) P(H)
2. Write down the factored conditional probability expression that corresponds to the graphical Bayesian Network shown below.



3. Shown below is the Bayesian network corresponding to the Burglar Alarm problem, P(J | A) P(M | A) P(A | B, E) P(B) P(E). The probability tables show the probability that variable is True, e.g., P(M) means P(M = t). Calculate P(J = t ∧ M = f ∧ A = f ∧ B = f ∧ E = t).



P(E)
.002

A	P(M)
t	.70
f	.01

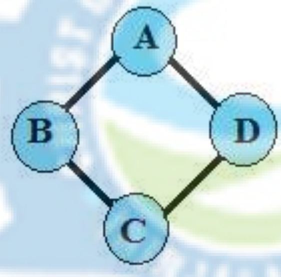
B	E	P(A)
t	t	.95
t	f	.94
f	t	.29
f	f	.001

P(B)
.001

A	P(J)
t	.90
f	.05

4. Consider the simple Markov network given below. Let A,B,C and D be binary random variables representing four people's beliefs as to whether the earth is round (1 for believes, 0 for does not believe). Determine the probability of only person A and D believes that the earth is round.

A	B	ϕ_{AB}
0	0	50
0	1	5
1	0	5
1	1	50



A	D	ϕ_{AD}
0	0	5
0	1	50
1	0	50
1	1	5

B	C	ϕ_{BC}
0	0	1
0	1	5
1	0	45
1	1	50

C	D	ϕ_{CD}
0	0	1
0	1	15
1	0	40
1	1	50

Course Outcome 5 (CO5):

1. Define effective sample size (ESS). Large ESS necessary but not sufficient for good MCMC mixing. Justify
2. Is an autoencoder for supervised learning or for unsupervised learning? Explain briefly.
3. Explain how does the variational auto-encoder(VAE) architecture allow it to generate new data points, compared to auto-encoder, which cannot generate new data points?
4. Generative Adversarial Networks(GANs) include a generator and a discriminator. Sketch a basic GAN using those elements, a source of real images, and a source of randomness.
5. Write down the formula for the energy function (E) of a Restricted Boltzmann Machine (RBM).
6. List the difference between Boltzmann Machine and Deep Belief Network.

7. One of your friends has trained a cat vs. non-cat classifier. It performs very well and you want to use transfer learning to build your own model. Explain what additional hyperparameters (due to the transfer learning) you will need to tune.

Course Outcome 6 (CO6):

1. Implement image classification using CNN
2. Implement ECG classification using LSTM

Syllabus

Module 1: Introduction to Deep learning

Introduction to deep learning, Deep feed forward network, Training deep models - introduction, setup and initialization issues, Vanishing and exploding gradient problems, Optimization techniques - Gradient Descent (GD), Stochastic GD, GD with momentum, GD with Nesterov momentum, AdaGrad, RMSProp, Adam. Regularization Techniques - L1 and L2 regularization, Early stopping, Dataset augmentation, Parameter tying and sharing, Ensemble methods, Dropout.

Module 2: Convolutional Neural Networks

Convolutional Neural Networks –Architecture, Convolution and Pooling operation, Motivation, Variants of convolution functions, Structured outputs, Data types, Efficient convolution algorithms, Training a Convolutional Network, Applications of Convolutional Networks, Case study of Convolutional Architectures – AlexNet

Module 3: Recurrent neural networks

Recurrent neural networks – Computational graphs, RNN design, encoder – decoder sequence to sequence architectures, deep recurrent networks, recursive neural networks, challenges of training Recurrent Networks, gated RNNs LSTM and GRU, Applications of RNNs.

Module 4: Graphical Models and Sampling

Graphical models - Bayesian network, Markov networks, Inference on chains and factor graphs. Monte Carlo Methods – Basics of Monte Carlo Sampling, Importance sampling, Markov chain Monte Carlo methods(MCMC), Gibbs sampling.

Module 5: Advanced Deep learning Topics

Autoencoders, Variational AutoEncoder , Deep generative models - Boltzmann machines, Restricted Boltzmann Machines, Deep Belief Networks, Deep Boltzmann Machines, ,Generative Adversarial Networks, Auto-Regressive Networks. Transfer Learning and Domain Adaptation.

Course Plan

No	Topic	No. of Lectures (40 Hours)
1	**Module 1: Introduction to deep learning**	**9**
1.1	Introduction to deep learning, Deep feed forward network	1
1.2	Training deep models - Introduction, setup and initialization issues	1
1.3	Vanishing and exploding gradient problems	1
1.4	Concepts of optimization, Gradient Descent (GD)	1
1.5	Stochastic GD, GD with momentum, GD with Nesterov momentum	1
1.6	AdaGrad, RMSProp, Adam	1
1.7	Concepts of Regularization, L1 and L2 regularization	1
1.8	Early stopping, Dataset augmentation	1
1.9	Parameter tying and sharing, Ensemble methods, Dropout	1
2	**Module 2 : Convolutional Neural Network**	**9**
2.1	Convolutional Neural Networks, Architecture	1
2.2	Convolution and Pooling operation with example	1
2.3	Motivation	1
2.4	Variants of convolution functions	1

2.5	Structured outputs, Data types	1
2.6	Efficient convolution algorithms	1
2.7	Training a Convolutional Network	1
2.8	Applications of Convolutional Networks	1
2.9	Case study of Convolutional Architectures – AlexNet	1
3	**Module 3 : Recurrent Neural Network**	**7**
3.1	Recurrent neural networks – Computational graphs	1
3.2	RNN design, Encoder – decoder sequence to sequence architectures	1
3.3	Deep recurrent networks, Recursive neural networks	1
3.4	Challenges of training Recurrent Networks	1
3.5	LSTM	1
3.6	GRU	1
3.7	Applications of RNN	1
4	**Module 4 : Graphical Models and Sampling**	**6**
4.1	Graphical models - Bayesian network	1
4.2	Markov network	1
4.3	Inference on chains and factor graphs	1
4.4	Monte Carlo Methods – Basics of Monte Carlo Sampling	1
4.5	Importance sampling	1
4.6	Markov chain Monte Carlo methods(MCMC), Gibbs sampling	1
5	**Module 5 : Advanced Deep learning Topics**	**9**
5.1	Autoencoders	1
5.2	Variational Autoencoder	1
5.3	Deep generative models - Boltzmann machines	1
5.4	Restricted Boltzmann Machines	1
5.5	Deep Belief Networks	1
5.6	Deep Boltzmann Machines	1

5.7	Generative Adversarial Networks	1
5.8	Auto-Regressive Networks	1
5.9	Transfer Learning and Domain Adaptation.	1

References

1. Goodfellow, I., Bengio,Y., and Courville, A., Deep Learning, MIT Press, 2016.
2. Neural Networks and Deep Learning, Aggarwal, Charu C., c Springer International Publishing AG, part of Springer Nature 2018.
3. Christopher M. Bishop. Pattern recognition and machine learning. Springer 2006.
4. David Foster. Generative Deep Learning - Teaching Machines to Paint, Write, Compose, and Play. O'Reilly Media, Inc., June 2019.

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
R25CST2X4	COMPUTER VISION	PROGRAM ELECTIVE 3	3	0	0	3

Preamble: Study of this course provides the leaners to explain the fundamental concepts of Computer Vision and Image Processing, and major approaches that address them. This course provides concepts in computer vision including image acquisition and image formation models. This course helps the learners to model fuzzy image processing, probabilistic model of image formation and different image compression techniques.

Prerequisite : Nil

Course Outcomes: After the completion of the course the student will be able to

CO 1	Apply domain knowledge in vision system, radiometry, shadows, shading, views and Stereopsis for image formation. **(Cognitive Knowledge Level: Apply)**
CO 2	Apply affine structures and Geometry for reconstructing images. **(Cognitive Knowledge Level: Apply)**
CO 3	Apply Bayesian decision theory for pattern classification. **(Cognitive Knowledge Level: Apply)**
CO 4	Analyze the contents of the image using image processing, probabilistic modelling and fuzzy image processing. **(Cognitive Knowledge Level: Analyze)**
CO 5	Analyze various techniques used for image restoration, compression and segmentation. **(Cognitive Knowledge Level: Analyze)**
CO 6	Design, develop, implement and present innovative ideas on computer vision concepts and techniques. **(Cognitive Knowledge Level: Create)**

Program Outcomes (PO)

Outcomes are the attributes that are to be demonstrated by a graduate after completing the course.

PO1: An ability to independently carry out research/investigation and development work in engineering and allied streams

PO2: An ability to communicate effectively, write and present technical reports on complex engineering activities by interacting with the engineering fraternity and with society at large.

PO3: An ability to demonstrate a degree of mastery over the area as per the specialization of the program. The mastery should be at a level higher than the requirements in the appropriate bachelor program

PO4: An ability to apply stream knowledge to design or develop solutions for real world problems by following the standards

PO5: An ability to identify, select and apply appropriate techniques, resources and state-of-the-art tool to model, analyse and solve practical engineering problems.

PO6: An ability to engage in life-long learning for the design and development related to the stream related problems taking into consideration sustainability, societal, ethical and environmental aspects

PO7: An ability to develop cognitive load management skills related to project management and finance which focus on Entrepreneurship and Industry relevance.

Mapping of course outcomes with program outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7
CO 1	2		3	2	2	1	
CO 2	2		3	3	2	1	
CO 3	2		3	3	2	1	
CO 4	3		3	2	3	2	
CO 5	3		3	2	3	2	
CO 6	3	2	3	3	3	2	2

Assessment Pattern

Bloom's Category	End Semester Examination
Apply	70%-80%
Analyse	30%-40%
Evaluate	
Create	

Mark distribution

Total Marks	CIE	ESE	ESE Duration
100	40	60	2.5 hours

Continuous Internal Evaluation Pattern:

Evaluation shall only be based on application, analysis or design based questions (for both internal and end semester examinations).

Continuous Internal Evaluation: 40 marks

i. Preparing a review article based on peer reviewed original publications (minimum 10 publications shall be referred) : 15 marks

ii. Course based task / Seminar/ Data collection and interpretation : 15 marks

iii. Test paper (1 number) : 10 marks

Test paper shall include minimum 80% of the syllabus.

Course based task/test paper questions shall be useful in the testing of knowledge, skills, comprehension, application, analysis, synthesis, evaluation and understanding of the students.

End Semester Examination Pattern:

The end semester examination will be conducted by the respective College.
There will be two parts; Part A and Part B.
Part A will contain 5 numerical/short answer questions with 1 question from each module, having 5 marks for each question. Students should answer all questions. Part B will contain 7 questions (such questions shall be useful in the testing of overall achievement and maturity of the students in a course, through long answer questions relating to theoretical/practical knowledge, derivations, problem solving and quantitative evaluation), with minimum one question from each module of which student should answer any five. Each question can carry 7 marks
Total duration of the examination will be 150 minutes.
Note: The marks obtained for the ESE for an elective course shall not exceed 20% over the average ESE mark % for the core courses. ESE marks awarded to a student for each elective course shall be normalized accordingly.
For example if the average end semester mark % for a core course is 40, then the maximum eligible mark % for an elective course is 40+20 = 60 %.

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. What is BRDF? How are area sources different from line sources?
2. What shapes can the shadow of a sphere take, if it is cast on a plane, and the source is a point source?
3. What are the different criteria for evaluating the performance of the computer vision algorithms?

Course Outcome 2 (CO2) :

1. How can a 3D structure captured using a sequence of video frames be recovered?
2. Discuss the methods that are used to segment the data points into independently-moving objects

Course Outcome 3(CO3):

1. Explain Minimum Squared Error Method (MSE) for Classification.
2. Explain how Bayesian Decision Theory can be used to determine whether a given email is "spam" or "non-spam".

Course Outcome 4 (CO4):

1. Explain about the basic relationships and distance measures between pixels in a digital image?
2. Explain Object recognition as probabilistic modelling.
3. How to define an image as a Fuzzy Set?

Course Outcome 5 (CO5):

1. Explain how an image is restored using an inverse filter. What are its drawbacks?
2. Given an image with the intensity distribution as

Intensity Value	Percentage(%)
0	20
32	10
128	25
144	5
152	5
160	10
168	5
249	5
250	10
251	5

Use the Huffman code to compress the image. Draw the Huffman tree.

Course Outcome 6 (CO6):

1. Implement any one applications- Object Recognition with Intelligent Cameras/ Fast 3-D Full Body Scanning for Humans and Other Objects/ Motion Tracking.

Syllabus

Module 1: Image formation and modelling

Components of a vision system, Imaging systems, Signal processing for computer vision, Pattern recognition for computer vision, Performance evaluation of algorithms.
Image formation and Image model- Camera model and camera calibration- Radiometry- Light in space- Light in surface - Sources, shadows and shading, Multiple images-The Geometry of multiple views- Stereopsis.

Module 2: Affine structures

Affine structure from motion- Elements of Affine Geometry, Affine structure and motion from two images- Affine structure and motion from multiple images- From Affine to Euclidean images.

High level vision- Geometric methods- Model based vision- Obtaining hypothesis by pose consistency, pose clustering and using Invariants, Verification.

Module 3: Bayesian Decision Theory

Bayesian Decision Theory- Minimum error rate classification Classifiers, discriminant functions, decision surfaces- The normal density and discriminant-functions for the Normal density.

Module 4: Introduction to Digital Image Processing

Introduction to Digital Image Processing- fundamental steps in Digital Image Processing, relationship between pixels, intensity transformations and spatial filtering: basic intensity transformation functions, histogram processing, spatial filtering, smoothing and sharpening filters

Probabilistic Modelling and Fuzzy Image Processing: Introduction of Probabilistic Modelling in Computer Vision, why probabilistic models, Object recognition as probabilistic modelling, Introduction, Fuzzy image understanding, Fuzzy image processing systems, Theoretical components of fuzzy image processing.

Module 5: Processing on Images

Image restoration: noise models, restoration in the presence of noise only, periodic noise reduction.
Image compression: fundamentals, compression models and standards, basic compression methods: Huffman coding, arithmetic coding, LZW coding, run-length coding.
Image segmentation: point, line and edge detection, thresholding region based segmentation

Case Study: Any two applications-Object Recognition with Intelligent Cameras/ Fast 3-D Full Body Scanning for Humans and Other Objects/ Motion Tracking/ Multicolour Classification of Astronomical Objects.

Course Plan

No	Topic	No. of Lectures (40 Hours)
1	**Module 1: Image formation and modelling**	**7**
1.1	Components of a vision system	1
1.2	Application for computer vision and performance evaluation.	1
1.3	Cameras- camera model and camera calibration	1
1.4	Radiometry- Light in space- Light in surface	1
1.5	Sources, shadows and shading	1
1.6	Multiple images-The Geometry of multiple views	1
1.7	Stereopsis	1
2	**Module 2: Affine structures**	**8**
2.1	Elements of Affine Geometry	1
2.2	Affine structure	1
2.3	Motion from two images	1
2.4	Affine structure and motion from multiple images	1
2.5	From Affine to Euclidean images.	1
2.6	High level vision- Geometric methods-	1
2.7	Model based vision- Obtaining hypothesis by pose consistency,	1
2.8	Pose clustering and using Invariants, Verification.	1
3	**Module 3: Bayesian Decision Theory**	**6**
3.1	Bayesian Decision Theory	1
3.2	Minimum error rate classification Classifiers	1
3.3	Discriminant functions	1
3.4	Decision surfaces.	1
3.5	The normal density and discriminant	1
3.6	Functions for the Normal density.	1
4	**Module 4: Introduction to Digital Image Processing**	**10**
4.1	Fundamental steps in Digital Image Processing	1
4.2	Relationship between pixels	1
4.3	Intensity transformations	1
4.4	Spatial filtering	1
4.5	basic intensity transformation functions	1
4.6	histogram processing	1
4.7	Spatial filtering	1
4.8	Smoothing and sharpening filters.	1
4.9	Probabilistic Modelling in Computer Vision	1
4.10	Fuzzy Image processing	1
5	**Module 5: Processing on Images**	**9**

5.1	Image restoration: noise models	1
5.2	Restoration in the presence of noise only	1
5.3	Periodic noise reduction.	1
5.4	Image compression: fundamentals	1
5.5	Compression models and standards	1
5.6	Basic compression methods: Huffman coding, arithmetic coding,	1
5.7	LZW coding, run-length coding,	1
5.8	Image segmentation: point, line and edge detection,	1
5.9	Thresholding , region based segmentation	1

References

1. C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

2. R. O. Duda, P. E. Hart and D. G. Stork, Pattern Classification, John Wiley, 2001.

3. Richard Hartley and Andrew Zisserman, Multiple View Geometry in Computer Vision, Second Edition, Cambridge University Press, 2004.

4. S. Theodoridis and K. Koutroumbas, Pattern Recognition, 4th Ed., Academic Press, 2009.

5. Gonzalez R. C. & Woods R. E., Digital Image Processing, 3rd ed, PHI Learning, 2008

6. Jain A K, Fundamentals of Digital Image Processing, Prentice-Hall India, 2007.

7. Bernd Jahne, Horst Haubecker, "Computer Vision and Applications", Academic Press.

8. David A. Forsyth, Jean Ponce, "Computer Vision: A Modern Approach", 2nd Ed., 2011.

9. Richard Szeliski, "Computer Vision: Algorithms and Applications", Springer, 1st Ed., 2010.

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
R25CST2X4	SEMANTIC WEB ARCHITECTURE	PROGRAM ELECTIVE 3	3	0	0	3

Preamble: This course helps the learners to understand the advanced concepts in emerging trends in Web architecture. The learner will be able to grasp the basics of metadata, OWL, RDF and ontologies. This course enables the learners to build an ontology for a given domain.

Course Outcomes: After the completion of the course the student will be able to

CO 1	Distinguish between Traditional web Environment and Semantic web Environment and in the behavior of a search engine. **(Cognitive Knowledge Level: Understand)**
CO 2	Recognize and Apply the concept of RDF and RDFS and the benefits of Ontology and Taxonomy **(Cognitive Knowledge Level: Apply)**
CO 3	Identify how to use OWL to rewrite the ontology and new features in OWL. **(Cognitive Knowledge Level: Analyze)**
CO 4	Analyze the Web services like UDDI and Real World Examples like Swoogle and FOAF. **(Cognitive Knowledge Level: Analyze)**
CO 5	Apply concepts OWL-S for Web service annotation and mapping OWL-S to UDDI **(Cognitive Knowledge Level: Apply)**
CO 6	Design, develop, implement or present innovative ideas on Semantic Web Architecture and techniques. **(Cognitive Knowledge Level: Create)**

Program Outcomes (PO)

Outcomes are the attributes that are to be demonstrated by a graduate after completing the course.

PO1: An ability to independently carry out research/investigation and development work in engineering and allied streams

PO2: An ability to communicate effectively, write and present technical reports on complex engineering activities by interacting with the engineering fraternity and with society at large.

PO3: An ability to demonstrate a degree of mastery over the area as per the specialization of the program. The mastery should be at a level higher than the requirements in the appropriate bachelor program

PO4: An ability to apply stream knowledge to design or develop solutions for real world problems by following the standards

PO5: An ability to identify, select and apply appropriate techniques, resources and state-of-the-art tool to model, analyse and solve practical engineering problems.

PO6: An ability to engage in life-long learning for the design and development related to the stream related problems taking into consideration sustainability, societal, ethical and environmental aspects

PO7: An ability to develop cognitive load management skills related to project management and finance which focus on Entrepreneurship and Industry relevance.

Mapping of course outcomes with program outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7
CO 1	1		3	1	1	2	
CO 2	2		3	2	2	1	
CO 3	2		3		3	1	
CO 4	2		3	3	3	2	
CO 5	2		3		3	1	
CO 6	3	2	3	3	3	2	2

Assessment Pattern

Bloom's Category	End Semester Examination
Apply	70%-80%
Analyze	30%-40%
Evaluate	
Create	

Mark distribution

Total Marks	CIE	ESE	ESE Duration
100	40	60	2.5 hours

Continuous Internal Evaluation Pattern:

Evaluation shall only be based on application, analysis or design based questions (for both internal and end semester examinations).

Continuous Internal Evaluation: 40 marks

i. Preparing a review article based on peer reviewed original publications (minimum 10 publications shall be referred) : 15 marks

ii. Course based task / Seminar/ Data collection and interpretation : 15 marks

iii. Test paper (1 number) : 10 marks

Test paper shall include minimum 80% of the syllabus.

Course based task/test paper questions shall be useful in the testing of knowledge, skills, comprehension, application, analysis, synthesis, evaluation and understanding of the students.

End Semester Examination Pattern:

The end semester examination will be conducted by the respective College.
There will be two parts; Part A and Part B.
Part A will contain 5 numerical/short answer questions with 1 question from each module, having 5 marks for each question. Students should answer all questions. Part B will contain 7 questions (such questions shall be useful in the testing of overall achievement and maturity of the students in a course, through long answer questions relating to theoretical/practical knowledge, derivations, problem solving and quantitative evaluation), with minimum one question from each module of which student should answer any five. Each question can carry 7 marks
Total duration of the examination will be 150 minutes.
Note: The marks obtained for the ESE for an elective course shall not exceed 20% over the average ESE mark % for the core courses. ESE marks awarded to a student for each elective course shall be normalized accordingly.
For example if the average end semester mark % for a core course is 40, then the maximum eligible mark % for an elective course is 40+20 = 60 %.

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Give the specific instances to illustrate the issue in the markup of the webpage?
2. "The Semantic Web is an extension of the current Web". Justify the statement.
3. Determine how metadata can be embedded in to existing web pages

Course Outcome 2 (CO2):

1. Differentiate RDF and RDFS

2. A Resource in the world of RDF schema has the same semantics as "class". Explain the statement with suitable example.

3. Illustrate the main differences in the concept of Taxonomy and Ontology.

Course Outcome 3(CO3):

1. Compare and contrast OWL and RDF?

2. OWL has created a new class called "owl:Class" to define classes in OWL documents; it is a subclass of "rdfs:Class". The relationship between all these top classes is summarized in Figure. Define all the classes in our camera ontology

Course Outcome 4 (CO4):

1. Determine in what manner ranking of documents using metadata done in Swoogle.

2. Illustrate why the Person class is the core of the FOAF vocabulary. Explain with an Example

3. Swoogle is expected to be used more often by the researchers and developers in the Semantic Web community. Why?

Course Outcome 5 (CO5):

1. Determine in general what is necessary for the concept of degree of matching? Explain the four degree of match between two concepts.

2. "OWL-S is not about learning a new language; instead, it is about understanding and using three new ontologies." Justify the statement.

3. Discuss the issues of mapping OWL-S profile information into UDDI registry

Course Outcome 6 (CO6):

1. Make a Project on Semantic Web that creates a platform to store data on the Web, build vocabularies, and write rules for handling data. Linked data are empowered by technologies such as RDF & OWL.

2. Take a Seminar on the Topic " Latest technologies in the Semantic Web" (for example Web 3.0).

Syllabus

Module 1 : Introduction to semantic web technology:
Traditional web to semantic web-WWW- First Look at the Semantic Web – meta data-Search Engine for the Traditional Web- Search Engine for the Semantic Web

Module 2: Resource Description Framework
Elements - Resource – Property- Statement -rules of RDF – tools- RDFS core elements--Syntax and Examples - More about Properties - XML Schema and RDF Schema –Taxonomy and ontology concepts.

Module 3: Web ontology language: OWL:
Define classes- set operators –enumerations- defining properties- Symmetric Properties-Transitive Properties- Functional Properties- Inverse Property- Inverse Functional Property-Validating OWL ontology- Related Development Tools- Validate OWL Ontology by Using Web Utilities- Using Programming APIs to Understand OWL Ontology.

Module 4: Web services and Real world examples: Web services – web services standards – web services to semantic web services- UDDI. Swoogle- architecture and usage of meta data; FOAF – vocabulary – creating documents – overview of semantic markup – semantic web search engines- Implementation Details.

Module 5: Concept of OWL-S – building blocks of OWL-S- OWL-S Profile Ontology-OWL-S Process Ontology- OWL-S Grounding Ontology - mapping OWL-S to UDDI - WSDL

Course Plan

No	Topic	No. Of Lectures (40 Hours)
1	**Module 1: Introduction to semantic web technology**	**6**
1.1	Traditional web to semantic web	1
1.2	World Wide Web	1
1.3	First Look at the Semantic Web	1
1.4	Meta data	1
1.5	Search Engine for the Traditional Web	1
1.6	Search Engine for the Semantic Web	1
2	**Module 2: Resource Description Framework**	**10**
2.1	Elements- Resource	1
2.2	Property- Statement	1
2.3	Rules of RDF	1
2.4	Tools	1
2.5	RDFS core elements	1
2.6	Syntax and Examples	1
2.7	Syntax and Examples	1
2.8	More about Properties	1
2.9	XML Schema and RDF Schema	1
2.10	Taxonomy and ontology concepts .	1
3	**Module 3: Web ontology language: OWL**	**9**
3.1	OWL: define classes	1
3.2	Set operators, enumerations	1
3.3	Defining properties- Symmetric Properties	1
3.4	Transitive Properties- Functional Properties	1
3.5	Inverse Property- Inverse Functional Property	1
3.6	Validating OWL ontology.	
3.7	Related Development Tools	1
3.8	Validate OWL Ontology by Using Web Utilities	1
3.9	Using Programming APIs to Understand OWL Ontology	1
4	**Module 4: Web services and Real world examples**	**10**
4.1	Web services	1
4.2	Web services standards	1
4.3	Web services to semantic web services	1
4.4	UDDI	1
4.5	Swoogle	1
4.6	Architecture and usage of meta data	1
4.7	FOAF – vocabulary – creating documents	1
4.8	Overview of semantic markup	1
4.9	Semantic web search engines.	1

4.10	Implementation Details	1
5	**Module 5: Concept of OWL-S**	**5**
5.1	Concept of OWL-S	1
5.2	Building blocks of OWL-S- OWL-S Profile Ontology	1
5.3	OWL-S Process Ontology- OWL-S Grounding Ontology	1
5.4	Mapping OWL-S to UDDI	1
5.5	WSDL	1

References

1. Liyang Yu, Introduction to the Semantic Web and Semantic web services. Chapman & Hall/CRC, Taylor & Francis group, 2007.

2. Johan Hjelm. Creating the Semantic Web with RDF. Wiley, 2001

3. Grigoris Antoniou and Frank van Harmelen. A Semantic Web Primer. MIT Press

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
R25CST2X4	PROGRAM ANALYSIS	PROGRAM ELECTIVE 3	3	0	0	3

Preamble: The course enables the learners to perform static analysis of programs. It explores methods such as abstract interpretation, intraprocedutral and pointer analysis, program slicing and type checking. The learner will be able to do program analysis in several applications such as compilers, tools that help programmers understand and modify programs, and tools that help programmers verify that programs satisfy certain properties of interest.

Course Outcomes: After the completion of the course the student will be able to

CO 1	Examine the various mathematical concepts needed for program analysis, including lattice, chains and Galois connections. (**Cognitive Knowledge Level: Analyze**)
CO 2	Investigate the role of abstract interpretation in static analysis of programs. (**Cognitive Knowledge Level: Analyze**)
CO 3	Use various methods such as interprocedural analysis, pointer analysis and program slicing for program analysis. (**Cognitive Knowledge Level: Analyze**)
CO 4	Use simply typed lambda calculus for developing type checking systems. (**Cognitive Knowledge Level: Analyze**)
CO 5	Design, develop and implement solutions based on the concepts of program analysis. (**Cognitive Knowledge Level: Create**)

Program Outcomes (PO)

Outcomes are the attributes that are to be demonstrated by a graduate after completing the course.

PO1: An ability to independently carry out research/investigation and development work in engineering and allied streams

PO2: An ability to communicate effectively, write and present technical reports on complex engineering activities by interacting with the engineering fraternity and with society at large.

PO3: An ability to demonstrate a degree of mastery over the area as per the specialization of the program. The mastery should be at a level higher than the requirements in the appropriate bachelor program

PO4: An ability to apply stream knowledge to design or develop solutions for real world problems by following the standards

PO5: An ability to identify, select and apply appropriate techniques, resources and state-of-the-art tools to model, analyse and solve practical engineering problems.

PO6: An ability to engage in life-long learning for the design and development related to the stream related problems taking into consideration sustainability, societal, ethical and environmental aspects

PO7: An ability to develop cognitive load management skills related to project management and finance which focus on Entrepreneurship and Industry relevance.

Mapping of course outcomes with program outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7
CO 1	2		3	1	2	2	
CO 2	3		3	2	3	2	
CO 3	2		3	2	3	1	
CO 4	2		3	2	2	1	
CO 5	3	2	3	3	3	2	2

Assessment Pattern

Bloom's Category	End Semester Examination
Apply	70%-80%
Analyze	30%-40%
Evaluate	
Create	

Mark distribution

Total Marks	CIE	ESE	ESE Duration
100	40	60	2.5 hours

Continuous Internal Evaluation Pattern:

Evaluation shall only be based on application, analysis or design-based questions (for both internal and end semester examinations).

Continuous Internal Evaluation: 40 marks

i. Preparing a review article based on peer reviewed original publications (minimum 10 publications shall be referred) : 15 marks

ii. Course based task / Seminar/ Data collection and interpretation : 15 marks

iii. Test paper (1 number) : 10 marks

Test paper shall include minimum 80% of the syllabus.

Course based task/test paper questions shall be useful in the testing of knowledge, skills, comprehension, application, analysis, synthesis, evaluation and understanding of the students.

End Semester Examination Pattern:

The end semester examination will be conducted by the respective College.
There will be two parts; Part A and Part B.
Part A will contain 5 numerical/short answer questions with 1 question from each module, having 5 marks for each question. Students should answer all questions. Part B will contain 7 questions (such questions shall be useful in the testing of overall achievement and maturity of the students in a course, through long answer questions relating to theoretical/practical knowledge, derivations, problem solving and quantitative evaluation), with minimum one question from each module of which student should answer any five. Each question can carry 7 marks
Total duration of the examination will be 150 minutes.

Note: The marks obtained for the ESE for an elective course shall not exceed 20% over the average ESE mark % for the core courses. ESE marks awarded to a student for each elective course shall be normalized accordingly.
For example, if the average end semester mark % for a core course is 40, then the maximum eligible mark % for an elective course is 40+20 = 60 %.

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Show that the inverse of a partial order is a partial order.

Course Outcome 2 (CO2):

1. For a statement n: x := y+5, considering the Constant Propagation (CP)transfer function f_n, show an element d_1 belonging to the CP abstract lattice such that $\gamma(f_n(d_1)) = nstate'(\gamma(d_1))$.

Course Outcome 3(CO3):

1. Using program slicing, show that the following program segments are equivalent.

c = initialState(c)
i = 0;
j = 0;
while (i < 100) {
  i = i+2;
  j = j-2;
  c = c+i+j;
}
finalUse(c);
	c = initialState(c)
	i = 0;
	j = 0;
	while (i < 100) {
	  j = j-2;
	  i = i+2;
	  c = c+i+j;
	}
	finalUse(c);

Course Outcome 4 (CO4):

1. Prove that the type of a variable is preserved under substitution.

Course Outcome 5 (CO5):

1. Given a Java program, implement a tool that performs interval analysis.

Syllabus

Module 1 - Mathematical Foundations for Program Analysis

–Revision of Partially Ordered Sets, Lattice, Chains, Fixed Points, Knaster-Tarski Fixed Point Theorem, Galois Connections and their properties, Introduction to Program Analysis.

Module 2- Data Flow Analysis / Abstract Interpretation

Collecting Semantics, Abstract Interpretation, Join Over all Paths, Abstract Interpretation for Constant propagation, Correctness of Abstract Interpretation, Kildall's algorithm.

Module 3- Interprocedural Analysis

Call Strings approach, Join Over Interprocedurally Valid Paths, Sharir and Pneuli's approaches to Interprocedural Analysis, Functional Approach.

Module 4 - Pointer Analysis, PDGs and Slicing

Pointer Analysis, Program Dependence Graph, Computing a Program Slice.

Module 5 - Simply Typed Lambda Calculus

Introduction to Lambda Calculus, Type Systems, Algorithms for Type Checking.

Course Plan

No	Topic	No. of Lectures (40 hrs)
1	**Module 1: Mathematical Foundations for Program Analysis**	**7**
1.1	Revision of Partially Ordered Sets, Complete Lattice	1
1.2	Chains, Fixed Points	1
1.3	Knaster-Tarski Fixed Point Theorem	1
1.4	Galois Connections	1
1.5	Properties of Galois Connections	1
1.6	Introduction to Program Analysis (Lecture 1)	1
1.7	Introduction to Program Analysis (Lecture 2)	1
2	**Module 2: Data Flow Analysis / Abstract Interpretation**	**8**
2.1	Collecting Semantics	1
2.2	Abstract Interpretation, Collecting abstract values	1
2.3	Comparison of abstract Join Over all Paths (JOP) states and collecting states	1
2.4	Abstract Interpretation for Constant Propagation	1
2.5	Correctness of Abstract Interpretation using Galois Connections	1
2.6	Kildall's algorithm for over-approximate JOP (Lecture 1)	1
2.7	Kildall's algorithm for over-approximate JOP (Lecture 2)	1
2.8	Kildall's algorithm for over-approximate JOP (Lecture 2)	1
3	**Module 3: Interprocedural Analysis**	**8**

3.1	Sharir-Pnueli's Call-Strings approach – Handling programs with procedure calls, Problem with JOP	1
3.2	Interprocedurally valid paths and their call-strings, Join Over Interprocedurally-valid Paths (JVP)	1
3.3	Sharir and Pnueli's approaches to Interprocedural Analysis (Lecture 1)	1
3.4	Sharir and Pnueli's approaches to Interprocedural Analysis (Lecture 2)	1
3.5	Sharir and Pnueli's approaches to interprocedural Analysis (Lecture 3)	1
3.6	Sharir-Pneuli's Functional Approach – Equation Solving Approach (Lecture 1)	1
3.7	Equation Solving Approach (Lecture 2)	1
3.8	Iterative / Tabular Approach	1
4	**Module 4: Pointer Analysis, PDGs and Slicing**	**9**
4.1	Points-To Analysis	1
4.2	May-Point-To Analysis	1
4.3	Andersen's Analysis	1
4.4	Program Dependence Graph (PDG)	1
4.5	Def-order Dependencies	1
4.6	Sequence of values at a node, Adequacy of PDGs, PDG Isomorphism	1
4.7	Computing a Program Slice (Lecture 1)	1
4.8	Computing a Program Slice (Lecture 2)	1
4.9	Computing a Program Slice (Lecture 3)	1
5	**Module 5: Simply Typed Lambda Calculus**	**7**
5.1	Introduction, Syntax, Semantics	1
5.2	Examples	1
5.3	Type Systems	1
5.4	Typing Rules, Using a derivation tree to prove that a term is well-typed	1
5.5	Properties – Progress	1
5.6	Properties - Preservation	1
5.7	Algorithm for type checking	1
5.8	Soundness of the algorithm	1

References

1. Nielson, Nielson, and Hankin, Principles of Program Analysis, Springer-Verlag.
2. Alfred Tarski, A lattice-theoretic fixpoint theorem and its applications, Pacific J. Mathematics, 5, pages 285--309, 1955.
3. Gary A. Kildall, A unified approach to global program optimization. In POPL '73: Proceedings of the 1st annual ACM SIGACT-SIGPLAN symposium on Principles of programming languages, pages 194--206, New York, NY, USA, 1973. ACM Press.

4. Patrick Cousot and RadhiaCousot, Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints, In POPL '77: Proceedings of the 4th ACM SIGACT-SIGPLAN symposium on Principles of programming languages, pages 238--252, New York, NY, USA, 1977. ACM Press.
5. Anders Moller and Michael I. Schwartzbach, Static Program Analysis.
6. MichaSharir and Amir Pnueli: Two approaches to interprocedural data-flow analysis, in Program Flow Analysis, Ed. Muchnik and Jones (1981).
7. Tom Reps, Susan Horwitz and MoolySagiv, Precise interprocedural dataflow analysis via graph reachability, In POPL '95, pages 49-61.
8. Manuvir Das, Sorin Lerner and Mark Seigle, ESP: path-sensitive program verification in polynomial time, ACM SIGPLAN Notices, Volume 37, Issue 5, May 2002, pages 57–68
9. Thomas Reps and Wuu Yang, The Sematics of Program Slicing, University of Wisconsin – Madison.
10. S. Horwitz, T. Reps, and D. Binkley. Interprocedural slicing using dependence graphs, Wisconsin – Madison.
11. Jeanne Ferrante, Karl J Ottenstein, and Joe D Warren, The program dependence graph and its use in optimization, ACM Transactions on Programming Languages and Systems, Vol. 9, No. 3, July 1987, Pages 319 – 349.
12. Karl J Ottenstein and Linda M Ottenstein, The program dependence graph in a software development environment.
13. Benjamin C. Pierce, "Types and Programming Languages". Relevant chapters: Chapters 1-4 (preliminaries), 5 and 6 (basics of lambda calculus), and 8-9 (simple type systems, including simply typed lambda calculus).
14. L. Damas and R. Milner, Principal type-schemes for functional programs, in POPL '82, pages 207--212.

SEMESTER II

PROGRAM ELECTIVE IV

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
R25CST2X5	BLOCKCHAIN TECHNOLOGY AND IOT	PROGRAM ELECTIVE 4	3	0	0	3

Preamble: The objective of this course is to enable to the student to build an IoT application. It also provides conceptual understanding of how blockchain technology can be used to innovate and improve business processes. This course covers the interconnection and integration of the physical world and the cyber space. This course will provide the students with the advanced competitive skills required to contribute to the development of the IoT.

Course Outcomes: After the completion of the course the student will be able to

CO 1	Analyze various protocols for IoT (**Cognitive Knowledge Level: Analyze**)
CO 2	Design IoT based applications using Arduino or Raspberry PI boards. (**Cognitive Knowledge Level: Apply**)
CO 3	Identify the need of blockchains to find the solution to the real-world problems. (**Cognitive Knowledge Level: Analyze**)
CO 4	Recognize the underlying technology of transactions, blocks, proof-of-work, and consensus building. (**Cognitive Knowledge Level: Analyze**)
CO 5	Design and implement new ways of using blockchain for real time applications. (**Cognitive Knowledge Level: Create**)

Program Outcomes (PO)

Outcomes are the attributes that are to be demonstrated by a graduate after completing the course.

PO1: An ability to independently carry out research/investigation and development work in engineering and allied streams

PO2: An ability to communicate effectively, write and present technical reports on complex engineering activities by interacting with the engineering fraternity and with society at large.

PO3: An ability to demonstrate a degree of mastery over the area as per the specialization of the program. The mastery should be at a level higher than the requirements in the appropriate bachelor program

PO4: An ability to apply stream knowledge to design or develop solutions for real world problems by following the standards

PO5: An ability to identify, select and apply appropriate techniques, resources and state-of-the-art tool to model, analyse and solve practical engineering problems.

PO6: An ability to engage in life-long learning for the design and development related to the stream related problems taking into consideration sustainability, societal, ethical and environmental aspects

PO7: An ability to develop cognitive load management skills related to project management and finance which focus on Entrepreneurship and Industry relevance.

Mapping of course outcomes with program outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7
CO 1	2		3	2			
CO 2	2		3	3	3	1	
CO 3	2		3	2		2	
CO 4	2		3	2		2	
CO 5	3	2	3	3	3	2	2

Assessment Pattern

Bloom's Category	End Semester Examination
Apply	70%-80%
Analyze	30%-40%
Evaluate	
Create	

Mark distribution

Total Marks	CIE	ESE	ESE Duration
100	40	60	2.5 hours

Continuous Internal Evaluation Pattern:

Evaluation shall only be based on application, analysis or design based questions (for both internal and end semester examinations).

Continuous Internal Evaluation: 40 marks

i. Preparing a review article based on peer reviewed original publications (minimum 10 publications shall be referred) : 15 marks

ii. Course based task / Seminar/ Data collection and interpretation : 15 marks

iii. Test paper (1 number) : 10 marks

Test paper shall include minimum 80% of the syllabus.

Course based task/test paper questions shall be useful in the testing of knowledge, skills, comprehension, application, analysis, synthesis, evaluation and understanding of the students.

End Semester Examination Pattern:

The end semester examination will be conducted by the respective College.
There will be two parts; Part A and Part B.
Part A will contain 5 numerical/short answer questions with 1 question from each module, having 5 marks for each question. Students should answer all questions. Part B will contain 7 questions (such questions shall be useful in the testing of overall achievement and maturity of the students in a course, through long answer questions relating to theoretical/practical knowledge, derivations, problem solving and quantitative evaluation), with minimum one question from each module of which student should answer any five. Each question can carry 7 marks
Total duration of the examination will be 150 minutes.
Note: The marks obtained for the ESE for an elective course shall not exceed 20% over the average ESE mark % for the core courses. ESE marks awarded to a student for each elective course shall be normalized accordingly.
For example if the average end semester mark % for a core course is 40, then the maximum eligible mark % for an elective course is 40+20 = 60 %.

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Describe how the environment can be more protected with the help of IoT technology in the following categories: (i) Air pollution monitoring (ii) Noise pollution monitoring (iii) Forest fire detection (iv) River flood detection.

Course Outcome 2 (CO2):

1. Design an automatic refrigerator light system with LED, switch & raspberry pi and write a python program to support the working of that design.

Course Outcome 3(CO3):

1. Identify the steps that are involved in the Blockchain project implementation.

Course Outcome 4 (CO4):

1. Write a crowd-sale smart contract code in Solidity programming language.

Course Outcome 5 (CO5):

1. Design and implement private blockchain for any given enterprise use-cases.

Syllabus

Module 1: Introduction to IoT

Sensor basics, sensing and actuation, basics of networking - wired, wireless, MANET, PAN, wireless and wired protocols, Communication protocols - IEEE standards, 5G era, sensor communications, connectivity challenges, fading and attenuations.

Module 2: IoT Architecture and Programming

basic architectures, Data processing mechanisms, scalability issues and visualization issues, analytic basics, utility of cloud computing, fog computing, edge computing, Raspberry Pi and Arduino programming, Applications- IoT for industrial automation (Industry 4.0).

Module 3: Introduction to Blockchain

Blockchain concepts, evolution, structure and characteristics, benefits and challenges, Blockchain as public ledgers - Transactions, Elements of Cryptography - Cryptographic Hash functions, Merkle Tree.

Module 4: Blockchain architecture and Use Cases

Design methodology for Blockchain applications, Blockchain application templates, Blockchain application development, Ethereum, Solidity,Bitcoin, Sample use cases from Industries,Business problems.

Module 5: Smart Contracts and Decentralized Applications (DApps)

Smart contract smart contract examples, structure of a contract, smart contract examples, smart contract patterns, implementing Dapps, Ethereum Dapps, case studies related to Dapps.

Course Plan

No	Topic	No. of Lecture (40 hrs)
1	**Module 1: Introduction to IoT**	**10**
1.1	Sensor basics	1
1.2	sensing and actuation	1
1.3	basics of networking - wired, wireless	1
1.4	MANET	1
1.5	PAN	1

1.6	Wireless and wired protocols	1
1.7	Communication protocols - IEEE standards	1
1.8	5G era, sensor communications	1
1.9	Connectivity challenges	1
1.10	Fading and attenuations	1
2	**Module 2: IoT architectures and programming**	**10**
2.1	Basic architectures	1
2.2	Data processing mechanisms	1
2.3	Scalability issues and visualization issues	1
2.4	Analytic basics	1
2.5	Utility of cloud computing	1
2.6	Fog computing	1
2.7	Edge computing	1
2.8	Raspberry Pi and Arduino programming	1
2.9	Applications	1
2.10	IoT for industrial automation (Industry 4.0)	1
3	**Module 3: Introduction to Blockchain**	**6**
3.1	Blockchain concepts	1
3.2	Evolution, structure and characteristics	1
3.3	Benefits and challenges	1
3.4	Blockchain as public ledgers - Transactions	1
3.5	Elements of Cryptography - Cryptographic Hash functions	1
3.6	Merkle Tree	1
4	**Module 4: Blockchain Architecture and Use cases**	**7**
4.1	Design methodology for Blockchain applications	1
4.2	Blockchain application templates	1
4.3	Blockchain application development	1
4.4	Ethereum, Solidity	1
4.5	Bitcoin	1
4.6	Sample use cases from Industries	1
4.7	Business problems	1

5	**Module 5: Smart contracts and Decentralized applications (Dapps)**	7
5.1	Smart contract smart contract examples	1
5.2	Structure of a contract	1
5.3	Smart contract examples	1
5.4	Smart contract patterns	1
5.5	Implementing Dapps	1
5.6	Ethereum Dapps	1
5.7	Case studies related to Dapps	1

References

1. The Internet of Things: Enabling Technologies, Platforms, and Use Cases, Pethuru Raj and Anupama C. Raman, CRC Press, First edition, 2017.

2. The Internet of Things in the Cloud: A Middleware Perspective, Honbu Zhou, CRC press, First edition, 2012.

3. Internet of Things: A Hands-on Approach, Arshdeep Bahga and Vijay Madisetti, Universities Press, First edition, 2014.

4. Fog for 5G and IoT (Information and Communication Technology Series, Mung Chiang, Bharath Balasubramanian, Flavio Bonomi, Wiley series, First edition, 2017.

5. Blockchain applications: a hands-on approach, Bahga A., Madisetti V., VPT, 2017.

6. Beginning Blockchain, A Beginner's Guide to Building Blockchain Solutions, Bikramaditya Singhal, Gautam Dhameja, Priyansu Sekhar Panda, Apress, 2018.

7. Blockchain A Practical Guide to Developing Business, Law, and Technology Solutions, Joseph J. Bambara and Paul R. Allen, McGraw Hill, 2018.

8. Blockchain enabled Applications Vikram Dhillon, David Metcalf and Max Hooper, Apress, 2017.

9. The Business Blockchain: Promise, Practice, and Application of the Next Internet Technology, William Mougayar, Wiley, 2016.

10. Blockchain Science: Distributed Ledger Technology, Roger Wattenhofer, Inverted Forest Publishing; 3rd edition, 2019.

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
R25CST2X5	SOCIAL NETWORK ANALYSIS	PROGRAM ELECTIVE 4	3	0	0	3

Preamble: This course provides an exposure to the concepts and techniques in Social Network Analysis. This course covers various types of modelling, visualization and mining techniques used in social networks. This course helps the learners to analyse social media data using appropriate data/web mining techniques.

Course Outcomes: After the completion of the course the student will be able to

CO 1	Understand the concepts and properties of social networks. **(Cognitive Knowledge Level: Understand)**
CO 2	Analyze the concepts of evolution and privacy in social networks. **(Cognitive Knowledge Level: Analyze)**
CO 3	Model and visualize social networks. **(Cognitive Knowledge Level: Apply)**
CO 4	Mine the behaviour of users in the Social Networks. **(Cognitive Knowledge Level: Analyze)**
CO 5	Use Multimedia Information Networks in Social Media. **(Cognitive Knowledge Level: Analyze)**
CO 6	Design, Develop, Implement and Present innovative ideas on Social network analysis concepts and techniques. **(Cognitive Knowledge Level: Create)**

Program Outcomes (PO)

Outcomes are the attributes that are to be demonstrated by a graduate after completing the course.

PO1: An ability to independently carry out research/investigation and development work in engineering and allied streams

PO2: An ability to communicate effectively, write and present technical reports on complex engineering activities by interacting with the engineering fraternity and with society at large.

PO3: An ability to demonstrate a degree of mastery over the area as per the specialization of the program. The mastery should be at a level higher than the requirements in the appropriate bachelor program

PO4: An ability to apply stream knowledge to design or develop solutions for real world problems by following the standards

PO5: An ability to identify, select and apply appropriate techniques, resources and state-of-the-art tools to model, analyse and solve practical engineering problems.

PO6: An ability to engage in life-long learning for the design and development related to the stream related problems taking into consideration sustainability, societal, ethical and environmental aspects.

PO7: An ability to develop cognitive load management skills related to project management and finance which focus on Entrepreneurship and Industry relevance.

Mapping of course outcomes with program outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7
CO 1	1					2	
CO 2	2		3		2	2	
CO 3	2		3	3	3	1	
CO 4	3		3	3	3	2	
CO 5	2		3	2	3	2	1
CO 6	3	2	3	3	3	2	2

Assessment Pattern

Bloom's Category	End Semester Examination
Apply	70%-80%
Analyze	30%-40%
Evaluate	
Create	

Mark distribution

Total Marks	CIE	ESE	ESE Duration
100	40	60	2.5 hours

Continuous Internal Evaluation Pattern:

Evaluation shall only be based on application, analysis or design-based questions (for both internal and end semester examinations).

Continuous Internal Evaluation: 40 marks

i. Preparing a review article based on peer reviewed original publications (minimum 10 publications shall be referred) : 15 marks

ii. Course based task/ Seminar/ Data collection and interpretation : 15 marks

iii. Test paper (1 number) : 10 marks

Test paper shall include minimum 80% of the syllabus.

Course based task/test paper questions shall be useful in the testing of knowledge, skills, comprehension, application, analysis, synthesis, evaluation and understanding of the students.

End Semester Examination Pattern:

The end semester examination will be conducted by the respective College.
There will be two parts; Part A and Part B.
Part A will contain 5 numerical/short answer questions with 1 question from each module, having 5 marks for each question. Students should answer all questions. Part B will contain 7 questions (such questions shall be useful in the testing of overall achievement and maturity of the students in a course, through long answer questions relating to theoretical/practical knowledge, derivations, problem solving and quantitative evaluation), with minimum one question from each module of which student should answer any five. Each question can carry 7 marks
Total duration of the examination will be 150 minutes.
Note: The marks obtained for the ESE for an elective course shall not exceed 20% over the average ESE mark % for the core courses. ESE marks awarded to a student for each elective course shall be normalized accordingly.
For example, if the average end semester mark % for a core course is 40, then the maximum eligible mark % for an elective course is 40+20 = 60 %.

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Explain the properties of dynamic weighted and unweighted graphs.

Course Outcome 2 (CO2):

1. Describe privacy breaches and privacy preserving mechanisms in social networks.

Course Outcome 3(CO3):

1. Classify the visualization of social networks.

Course Outcome 4 (CO4):

1. Given a social network with labels on some nodes, how to provide a high quality labelling for every node?

2. What are the types of changes in the evolution of the web community?

Course Outcome 5 (CO5):

1. Describe the different kinds of tags.
2. Explain the applications of tags.

Syllabus

Module 1: Introduction

Introduction to Social Network Data Analytics: Introduction, Online Social Networks: Research Issues, Research Topics in Social Networks. Statistical Properties of Social Networks: Preliminaries, Static Properties, Dynamic Properties. Random Walks in Social Networks and their Applications: Random Walks on Graphs: Background, Application in Computer Vision, Text Analysis, Collaborative Filtering.

Module 2: Evolution in Social Networks

Evolution in Social Networks: Framework, Challenges of Social Network Streams, Incremental Mining for Community Tracing, Tracing Smoothly Evolving Communities. Models and Algorithms for Social Influence Analysis: Influence Related Statistics, Social Similarity and Influence. Privacy in Social Networks: Privacy breaches in social networks, Privacy-preserving mechanisms.

Module 3: Visualizing Social Networks

Visualizing Social Networks: A Taxonomy of Visualizing Social Networks: A Taxonomy of Visualizations. Data Mining in Social Media: Methods for Social Media, Ethnography and Netnography, Event Maps. Text Mining in Social Networks: Keyword Search, Classification and Clustering Algorithms, Transfer Learning in Heterogeneous Networks.

Module 4: Mining Communities

Aggregating and reasoning with social network data, Advanced Representations - Extracting evolution of Web Community from a Series of Web Archive -Detecting Communities in Social Networks - Evaluating Communities – Core Methods for Community Detection & Mining - Applications of Community Mining Algorithms - Node Classification in Social Networks

Module 5: Multimedia Information Networks in Social Media

Multimedia Information Networks in Social Media: Links from Semantics, Links from Community Media. Network of Personal Photo Albums, Geographical Information, Inference Methods. Social Tagging and Applications: Tags: Why What, Tagging System Design, Tag analysis, Visualization of Tags, Applications of Tags.

Course Plan

No	Topic	No. of Lectures (40 Hours)
1	**Module 1: Introduction**	**8**
1.1	Introduction to Social Network Data Analytics	1
1.2	Online Social Networks: Research Issues	1
1.3	Research Topics in Social Networks	1
1.4	Statistical Properties of Social Networks: Preliminaries, Static Properties, Dynamic Properties	1
1.5	Random Walks in Social Networks and their Applications	1
1.6	Random Walks on Graphs: Background	1
1.7	Application in Computer Vision	1
1.8	Text Analysis, Collaborative Filtering	1
2	**Module 2: Evolution in Social Networks**	**8**
2.1	Framework	1
2.2	Challenges of Social Network Streams	1
2.3	Incremental Mining for Community Tracing	1
2.4	Tracing Smoothly Evolving Communities	1
2.5	Models and Algorithms for Social Influence Analysis: Influence Related Statistics	1
2.6	Social Similarity and Influence	1
2.7	Privacy in Social Networks: Privacy breaches in social networks	1
2.8	Privacy-preserving mechanisms	1
3	**Module 3: Visualizing Social Networks**	**8**
3.1	A Taxonomy of Visualizing Social Networks	1
3.2	A Taxonomy of Visualizations	1
3.3	Data Mining in Social Media: Methods for Social Media	1
3.4	Ethnography and Netnography, Event Maps	1
3.5	Text Mining in Social Networks: Keyword Search	1
3.6	Text Mining in Social Networks: Keyword Search	1
3.7	Classification and Clustering Algorithms	1
3.8	Transfer Learning in Heterogeneous Networks	1
4	**Module 4: Mining Communities**	**8**
4.1	Aggregating and reasoning with social network data	1
4.2	Advanced Representations - Extracting evolution of Web Community from a Series of Web Archive	1
4.3	Detecting Communities in Social Networks	1
4.4	Evaluating Communities	1
4.5	Core Methods for Community Detection & Mining	1
4.6	Core Methods for Community Detection & Mining	1
4.7	Applications of Community Mining Algorithms	1
4.8	Node Classification in Social Networks	1

5	**Module 5: Multimedia Information Networks in Social Media**	**8**
5.1	Links from Semantics, Links from Community Media	1
5.2	Network of Personal Photo Albums	1
5.3	Geographical Information	1
5.4	Inference Methods	1
5.5	Social Tagging and Applications: Tags: Why What	1
5.6	Tagging System Design	1
5.7	Tag analysis	1
5.8	Visualization of Tags, Applications of Tags	1

References

1. Charu C. Aggarwal, "Social Network Data Analytics", Springer.
2. Peter Mika, "Social Networks and the Semantic Web", Springer, 1St edition 2007.
3. BorkoFurht, "Handbook of Social Network Technologies and Applications", Springer, 1st edition, 2010.
4. GuandongXu, Yanchun Zhang and Lin Li, "Web Mining and Social Networking Techniques and applications", Springer, 1St edition, 2011.

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
R25CST2X5	MODERN DATABASE MANAGEMENT	PROGRAM ELECTIVE 4	3	0	0	3

Preamble: This course provides an exposure to the concepts and techniques in modern data management. Different types of NoSQL databases, their architecture and use cases are discussed in this course. A better understanding of data management is provided through MongoDB, Cassandra and Neo4J databases. This course helps the learners to develop applications that manage data efficiently with the help of suitable data models and techniques.

Course Outcomes: After the completion of the course the student will be able to:

CO 1	Analyze issues and solutions in modern databases. **(Cognitive Knowledge Level: Understand)**
CO 2	Employ and operate on document databases and techniques. **(Cognitive Knowledge Level: Apply)**
CO 3	Apply spatial database concepts to efficiently organise and retrieve spatial data. **(Cognitive Knowledge Level: Apply)**
CO 4	Research, analyze and use emerging technologies in column store. **(Cognitive Knowledge Level: Analyze)**
CO 5	Practice techniques in Graph database. **(Cognitive Knowledge Level: Analyze)**
CO 6	Design, Develop, Implement and Present innovative ideas on modern database concepts and techniques. **(Cognitive Knowledge Level: Create)**

Program Outcomes (PO)

Outcomes are the attributes that are to be demonstrated by a graduate after completing the course.

PO1: An ability to independently carry out research/investigation and development work in engineering and allied streams

PO2: An ability to communicate effectively, write and present technical reports on complex engineering activities by interacting with the engineering fraternity and with society at large.

PO3: An ability to demonstrate a degree of mastery over the area as per the specialization of the program. The mastery should be at a level higher than the requirements in the appropriate bachelor program

PO4: An ability to apply stream knowledge to design or develop solutions for real world problems by following the standards

PO5: An ability to identify, select and apply appropriate techniques, resources and state-of-the-art tool to model, analyse and solve practical engineering problems.

PO6: An ability to engage in life-long learning for the design and development related to the stream related problems taking into consideration sustainability, societal, ethical and environmental aspects

PO7: An ability to develop cognitive load management skills related to project management and finance which focus on Entrepreneurship and Industry relevance.

Mapping of course outcomes with program outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7
CO 1	1		3	2	1	2	
CO 2	2		3	3	3	1	
CO 3	2		3	3	3	2	
CO 4	3		3	2	3	2	
CO 5	2		3	3	3	2	
CO 6	3	2	3	3	3	2	2

Assessment Pattern

Bloom's Category	End Semester Examination
Apply	70%-80%
Analyze	30%-40%
Evaluate	
Create	

Mark distribution

Total Marks	CIE	ESE	ESE Duration
100	40	60	2.5 hours

Continuous Internal Evaluation Pattern:

Evaluation shall only be based on application, analysis or design-based questions (for both internal and end semester examinations).

Continuous Internal Evaluation: 40 marks

i. Preparing a review article based on peer reviewed original publications (minimum 10 publications shall be referred) : 15 marks

ii. Course based task / Seminar/ Data collection and interpretation : 15 marks

iii. Test paper (1 number) : 10 marks

Test paper shall include minimum 80% of the syllabus.
Course based task/test paper questions shall be useful in the testing of knowledge, skills, comprehension, application, analysis, synthesis, evaluation and understanding of the students.

End Semester Examination Pattern:

The end semester examination will be conducted by the respective College.
There will be two parts; Part A and Part B.
Part A will contain 5 numerical/short answer questions with 1 question from each module, having 5 marks for each question. Students should answer all questions. Part B will contain 7 questions (such questions shall be useful in the testing of overall achievement and maturity of the students in a course, through long answer questions relating to theoretical/practical knowledge, derivations, problem solving and quantitative evaluation), with minimum one question from each module of which student should answer any five. Each question can carry 7 marks
Total duration of the examination will be 150 minutes.
Note: The marks obtained for the ESE for an elective course shall not exceed 20% over the average ESE mark % for the core courses. ESE marks awarded to a student for each elective course shall be normalized accordingly.
For example, if the average end semester mark % for a core course is 40, then the maximum eligible mark % for an elective course is 40+20 = 60 %.

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Give specific instances to illustrate vertical and horizontal fragmentation in distributed databases.
2. The figure shows version v1 of a system. Values ‘berlin’ and ‘142’ are assigned to city and number fields of the address object respectively.

    i. Explain how the system can maintain its consistency using object versioning after the change.

ii. Draw version 2 of the system.

Course Outcome 2 (CO2):

1. While creating Schema in MongoDB what are the points that need to be taken in consideration? Explain with an example.

Course Outcome 3(CO3):

1. Use R-Tree to store the following points in which only 3 points/node. (10,10), (15,10), (20,10), (30,20), (50,10), (50,80), (60,30), (60,60), (70,20), (75,10).

Course Outcome 4 (CO4):

1. How does Cassandra handle large volumes of data efficiently? Describe the use of Sorted String Tables and consistent hashing with virtual nodes in Cassandra.

Course Outcome 5 (CO5):

1. Write the Neo4j Cypher Query which returns all the players (nodes) that belong to the country India and have scored runs greater than 210. Assume that the below nodes are created using Neo4j.

CREATE (Dhawan: player {name: "Shikar Dhawan", YOB: 1985, runs:363, country: "India"})

CREATE (Jonathan: player {name: "Jonathan Trott", YOB:1981, runs:229, country: "South Africa"})

CREATE (Sangakkara: player {name: "KumarSangakkara", YOB:1977, runs:222, country: "Sri Lanka"})

CREATE (Virat: player {name:"ViratKohli", YOB: 1988, runs:176, country: "India"})

Course Outcome 6 (CO6):

1. Implement E-Commerce stream analysis using Cassandra.

Syllabus

Module 1: NoSQL Databases

Review of Distributed Databases- Fragmentation, Replication, Transparencies in design, CAP theorem- BASE transactions and eventual consistency, consistent hashing, object versioning & vector clocks, Consensus algorithms, Logging & Snapshots, Properties of NoSQL databases, Types of NoSQL Databases.

Module 2: Document based databases

MongoDB- Documents- JSON & BSON format, representing relationships, CRUD operations, Indexing, Aggregation, Sharding architecture and Replication strategies, consistency and locking.

Module 3: Spatial Databases

Types of Spatial Data and Queries- Point and Region Data- Queries, Spatial Indexing: Space Filling Curves- Z ordering, Quad Trees, R-Trees, Geospatial queries & geospatial indexes in MongoDB.

Module 4: Column databases

Column family, Cassandra Architecture-Gossiping, Snitches, Rings and Tokens, Virtual Nodes, Replication Strategies, Consistency, Hinted handoff, Lightweight Transactions, Bloom filter, Compaction; Fault tolerance, Caching, SSTable & MemTable, Cassandra Query Language

Module 5: Graph Databases

Neo4j- Introduction, Example graphs, Data Modeling, Traversal, Indexing, Features, operations, Cypher Queries- Create, Match clause.

Course Plan

No	Topic	No. of Lectures (40 Hours)
1	**Module 1: NoSQL Databases**	**8**
1.1	Review of Distributed Databases	1
1.2	CAP theorem- BASE transactions and eventual consistency	1
1.3	Consistent hashing	1
1.4	Object versioning	1
1.5	Vector clocks	1
1.6	Consensus algorithms	1

1.7	Logging & Snapshots	1
1.8	Properties of NoSQL databases, Types of NoSQL Databases.	1
2	**Module 2: Document based databases**	**10**
2.1	Introduction to Document Database & MongoDB	1
2.2	MongoDB Documents- JSON & BSON format	1
2.3	Representing relationships	1
2.4	CRUD operations	1
2.5	Indexing	1
2.6	Aggregation	1
2.7	Sharding architecture	1
2.8	Replication strategies	1
2.9	Consistency	1
2.10	Locking	1
3	**Module 3: Spatial Database**	**8**
3.1	Types of Spatial Data	1
3.2	Types of Spatial Queries	1
3.3	Spatial Indexing	1
3.4	Space Filling Curves- Z ordering	1
3.5	Quad Trees	1
3.6	R-Trees	1
3.7	Geospatial queries in MongoDB	1
3.8	Geospatial indexes in MongoDB	1
4	**Module 4: Column databases**	**8**
4.1	Introduction to Column family & Cassandra, Cassandra Architecture-Gossiping	1
4.2	Snitches, Rings and Tokens	1
4.3	Virtual Nodes & Replication Strategies	1
4.4	Consistency	1
4.5	Hinted handoff & Lightweight Transactions	1
4.6	Bloom filter, Compaction	1
4.7	Fault tolerance, Caching, SSTable & MemTable	1
4.8	Cassandra Query Language	1
5	**Module 5: Graph Databases**	**6**
5.1	Introduction to graph databases	1
5.2	Neo4j, Example graphs	1
5.3	Data Modeling	1
5.4	Traversal, Indexing, Features and operations	1
5.5	Cypher Queries- CREATE	1
5.6	Cypher Queries- MATCH	1

References

1. Guy Harrison, “Next Generation Databases: NoSQL, NewSQL, and Big Data”, Apress, 2016
2. Thomas Cannolly and Carolyn Begg, “Database Systems, A Practical Approach to Design, Implementation and Management”, 6/e, Pearson Education, 2015.
3. Henry F Korth, Abraham Silberschatz, S. Sudharshan, “Database System Concepts”, 7/e, Tata McGraw Hill, 2019.
4. Pramod Sadalage, Martin Fowler,” NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence”, Addison-Wesley, 2012
5. Shannon Bradshaw, Eoin Brazil, Kristina Chodorow, “MongoDB: The Definitive Guide”, 3/e, O’Reilly, 2019
6. Ian Robinson, Jim Webber, Emil Eifrem, “Graph databases”, O’Reilly, 2013.
7. Jeff Carpenter, Eben Hewitt, “Cassandra: The Definitive Guide”, 3/e, O’Reilly, 2020
8. Dan Sullivan, “NoSQL for Mere Mortals”, Addison-Wesley, 2015
9. Alex Petrov, “Database Internals: A Deep Dive into How Distributed Data Systems Work”, O’Reilly, 2019
10. Luc Perkins, Eric Redmond, Jim Wilson, “Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement”, 2/e, O’Reilly, 2018
11. Jérôme Baton, Rik Van Bruggen, “Learning Neo4j 3.x”, 2/e, Packt, 2017.
12. Aleksa Vukotic, Nicki Watt, “Neo4j in Action”, Manning Publications, 2015.
13. Shashi Shekhar, Sanjay Chawla, Spatial databases A Tour, Pearson Education, Indian Edition, First Impression 2009.
14. Web Resource: https://neo4j.com/docs/operations-manual/current/

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
R25CST2X5	DISTRIBUTED ALGORITHMS	PROGRAM ELECTIVE 4	3	0	0	3

Preamble: The purpose of this course is to introduce the distributed algorithms with an emphasis on principles and theory. This course provides the concepts and techniques of distributed algorithms in synchronous and asynchronous distributed computing systems. This course helps the leaners to gain a good idea regarding various system models and their capabilities, which will help to design new algorithms.

Course Outcomes: After the completion of the course the student will be able to

CO 1	Understand the working of problem domain in synchronous distributed computing system. **(Cognitive Knowledge Level: Understand)**
CO 2	Develop algorithms for synchronous distributed computing system**(Cognitive Knowledge Level: Apply)**
CO 3	Examine the consensus problem in a distributed computing system**(Cognitive Knowledge Level: Apply)**
CO 4	Illustrate the use of different communication models in asynchronous distributed computing systems. **(Cognitive Knowledge Level: Apply)**
CO 5	Develop algorithms for various resource allocation problems in Asynchronous distributed computing systems. **(Cognitive Knowledge Level: Apply)**
CO 6	Design, Develop, Implement and Present innovative ideas on distributed algorithms and techniques. **(Cognitive Knowledge Level: Create)**

Program Outcomes (PO)

Outcomes are the attributes that are to be demonstrated by a graduate after completing the course.

PO1: An ability to independently carry out research/investigation and development work in engineering and allied streams

PO2: An ability to communicate effectively, write and present technical reports on complex engineering activities by interacting with the engineering fraternity and with society at large.

PO3: An ability to demonstrate a degree of mastery over the area as per the specialization of the program. The mastery should be at a level higher than the requirements in the appropriate bachelor program

PO4: An ability to apply stream knowledge to design or develop solutions for real world problems by following the standards

PO5: An ability to identify, select and apply appropriate techniques, resources and state-of-the-art tool to model, analyse and solve practical engineering problems.

PO6: An ability to engage in life-long learning for the design and development related to the stream related problems taking into consideration sustainability, societal, ethical and environmental aspects

PO7: An ability to develop cognitive load management skills related to project management and finance which focus on Entrepreneurship and Industry relevance.

Mapping of course outcomes with program outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7
CO 1	1					2	
CO 2	2		3	3	2	1	
CO 3	2		3		3	2	
CO 4	2		3		2	2	
CO 5	3		3	3	3	2	
CO 6	3	2	3	3	3	2	2

Assessment Pattern

Bloom's Category	End Semester Examination
Apply	70%-80%
Analyze	30%-40%
Evaluate	
Create	

Mark distribution

Total Marks	CIE	ESE	ESE Duration
100	40	60	2.5 hours

Continuous Internal Evaluation Pattern:

Evaluation shall only be based on application, analysis or design based questions (for both internal and end semester examinations).

Continuous Internal Evaluation: 40 marks

Preparing a review article based on peer reviewed original publications (minimum 10 publications shall be referred) : 15 marks

Course based task / Seminar/ Data collection and interpretation : 15 marks
Test paper (1 number) : 10 marks

Test paper shall include minimum 80% of the syllabus.
Course based task/test paper questions shall be useful in the testing of knowledge, skills, comprehension, application, analysis, synthesis, evaluation and understanding of the students.

End Semester Examination Pattern:

The end semester examination will be conducted by the respective College.
There will be two parts; Part A and Part B.
Part A will contain 5 numerical/short answer questions with 1 question from each module, having 5 marks for each question. Students should answer all questions. Part B will contain 7 questions (such questions shall be useful in the testing of overall achievement and maturity of the students in a course, through long answer questions relating to theoretical/practical knowledge, derivations, problem solving and quantitative evaluation), with minimum one question from each module of which student should answer any five. Each question can carry 7 marks
Total duration of the examination will be 150 minutes.
Note: The marks obtained for the ESE for an elective course shall not exceed 20% over the average ESE mark % for the core courses. ESE marks awarded to a student for each elective course shall be normalized accordingly.
For example if the average end semester mark % for a core course is 40, then the maximum eligible mark % for an elective course is 40+20 = 60 %.

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Explain the complexity measures of synchronous distributed computing system.
2. Compare invariant assertion and simulations.
3. Illustrate the execution of a system.

Course Outcome 2 (CO2):

1. For the LCR algorithm,
   a. Give a UID assignment for which $\Omega(n^2)$ messages are sent.
   b. Give a UID assignment for which only $O(n)$ messages are sent.
2. Design a unidirectional leader-election algorithm that works with unknown ring size. Your algorithm should manipulate the UIDs using comparisons only.
3. Suppose that LubyMIS is executed in a ring of size n. Estimate the probability that any particular edge is removed from the graph in one iteration of the algorithm.

Course Outcome 3(CO3):

1. Explain Consensus in a distributed computing system.
2. Compare Byzantine failure and stopping failure.
3. Describe the required conditions for the K agreement problem.

Course Outcome 4 (CO4):

1. Describe a specific I/O automaton having no input actions, whose output actions are {0, 1, 2,... }, and whose fair traces are exactly the sequences in set S, defined as follows. S consists of all the sequences of length 1 over the output set, that is, all the sequences consisting of exactly one nonnegative integer.
2. Write a pre condition effect code for Reliable reordering channel.

Course Outcome 5 (CO5):

1. Describe a fair execution of the DijkstraME algorithm in which a particular process is locked out.
2. Illustrate Right-Left Dining Philosophers Algorithm.

Course Outcome 6 (CO6):

1. Implement FloodMax algorithm in a synchronous distributed computing system.
2. Implement Peterson Leader-Election Algorithm in an asynchronous distributed computing system.

Syllabus

Module 1: Synchronous distributed computing system

Synchronous distributed computing system, Leader Election in a Synchronous Ring, LCR algorithm, HS algorithm, Time Slice Algorithm, Variable Speeds Algorithm, Lower Bound for Comparison-Based Algorithms.

Module 2: Algorithms in General Synchronous Networks

Algorithms in General Synchronous Networks. Leader election in a General Network - Simple Flooding Algorithm, Basic Breadth- First Search Algorithm, Bellman-Ford algorithm, Minimum Spanning Tree, Maximal Independent Set, LubyMIS algorithm.

Module 3: Distributed Consensus

Distributed Consensus, Distributed Consensus with Link Failures – The coordinated Attack Problem- Deterministic version, Randomized version. Distributed Consensus with Process Failures and Process Failures. Algorithms for stopping Failures, Algorithms for Byzantine Failure. Byzantine agreement in general graphs. Weak Byzantine agreement. Consensus Problems-K Agreement, Approximate Agreement, Commit Problem.

Module 4: Asynchronous distributed computing system

Asynchronous distributed computing system, Asynchronous Network Model - Send/Receive systems, Broadcast systems, Multicast systems. Asynchronous Network algorithms- Peterson Leader-Election Algorithm, Local Synchronizer, Safe Synchronizer.

Module 5: Asynchronous Shared Memory Systems

Asynchronous Shared Memory Systems, Environment Model, Shared Variable Types. Mutual Exclusion - Asynchronous Shared Memory Model, Dijkstra's Mutual Exclusion Algorithm. Resource Allocation - Nonexistence of Symmetric Dining Philosophers Algorithms, Right-Left Dining Philosophers Algorithm, Mutual exclusion and Consensus, Relationship between shared memory and network models.

Course Plan

No	Topic	No. of Lectures (40 hrs)
1	**Module 1: Synchronous distributed computing system**	**8**
1.1	Synchronous distributed computing system	1
1.2	Synchronous network model	1
1.3	Leader Election in a Synchronous Ring	1
1.4	LCR algorithm	1
1.5	HS algorithm	1
1.6	Time Slice Algorithm	1
1.7	Variable Speeds Algorithm	1
1.8	Lower Bound for Comparison-Based Algorithms	1
2	**Module 2: Algorithms in General Synchronous Networks**	**8**
2.1	Algorithms in General Synchronous Networks	1
2.2	Leader election in a General Network	1
2.3	Simple Flooding Algorithm	1
2.4	Basic Breadth- First Search Algorithm	1
2.5	Bellman-Ford algorithm	1
2.6	Minimum Spanning Tree	1
2.7	Maximal Independent Set	1
2.8	LubyMIS algorithm.	1
3	**Module 3: Distributed Consensus**	**8**
3.1	Distributed Consensus	1
3.2	The coordinated Attack Problem- Deterministic version, Randomized version	1
3.3	Distributed Consensus with Process Failures and Process Failures	1
3.4	Algorithms for stopping Failures	1
3.5	Algorithms for Byzantine Failure	1
3.6	Byzantine agreement in general graphs	1
3.7	Weak Byzantine agreement	1
3.8	Consensus Problems-K Agreement, Approximate Agreement, Commit Problem	1
4	**Module 4: Asynchronous distributed computing system**	**7**
4.1	Asynchronous distributed computing system	1
4.2	Asynchronous Network Model - Send/Receive systems, Broadcast systems	1
4.3	Multicast systems	1
4.4	Asynchronous Network algorithms	1
4.5	Peterson Leader-Election Algorithm	1
4.6	Local Synchronizer	1
4.7	Safe Synchronizer	1
5	**Module 5: Asynchronous Shared Memory Systems**	**9**

5.1	Asynchronous Shared Memory Systems	1
5.2	Environment Model	1
5.3	Shared Variable Types	1
5.4	Mutual Exclusion - Asynchronous Shared Memory Model	1
5.5	Dijkstra's Mutual Exclusion Algorithm	1
5.6	Resource Allocation - Nonexistence of Symmetric Dining Philosophers Algorithms	1
5.7	Right-Left Dining Philosophers Algorithm	1
5.8	Mutual exclusion and Consensus	1
5.9	Relationship between shared memory and network models	1

References

1. Nancy A. Lynch, Distributed Algorithms, Morgan Kaufmann Publishers, Inc,1996.
2. Wolfgang Reisig, W. Reisig, Elements Of Distributed Algorithms: Modeling And Analysis With Petri Nets, Springer-verlag, First Edition, 1998.
3. Tel Gerard , Introduction To Distributed Algorithms, Second Edition, Cambridge University Press,2000
4. Valmir C. Barbosa, An Introduction To Distributed Algorithms, Mit Press,1996
5. Randy Chow, Theodore Johnson, Distributed Operating Systems and Algorithm Analysis, Pearson Education,2009.

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
R25CST2X5	CYBER FORENSICS AND INFORMATION SECURITY	PROGRAM ELECTIVE 4	3	0	0	3

Preamble: This course provides an exposure to the concepts and techniques in Computer Forensics Technologies. Different types of Computer Forensics systems, evidence of data gathered, network attacks, evidence data analysis and forensics tools are discussed in this course. This course helps the learners to design, and develop innovative ideas on different digital forensic investigation models

Course Outcomes: After the completion of the course the student will be able to

CO 1	Apply forensics technologies, data recovery , evidence collection and handling different forensics issues. **(Cognitive Knowledge Level: Apply)**
CO 2	Solve the issues of Data Acquisition and Data Recovery. **(Cognitive Knowledge Level: Apply)**
CO 3	Investigate network intrusions and attacks. **(Cognitive Knowledge Level: Analyze)**
CO 4	Validating Forensics data and process crime, incident scene. **(Cognitive Knowledge Level: Analyze)**
CO 5	Exploring file structures and perform forensics investigation. **(Cognitive Knowledge Level: Analyze)**
CO6	Design, Develop, Implement and Present innovative ideas on different digital forensic investigation model. **(Cognitive Knowledge Level: Create)**

Program Outcomes (PO)

Outcomes are the attributes that are to be demonstrated by a graduate after completing the course.

PO1: An ability to independently carry out research/investigation and development work in engineering and allied streams

PO2: An ability to communicate effectively, write and present technical reports on complex engineering activities by interacting with the engineering fraternity and with society at large.

PO3: An ability to demonstrate a degree of mastery over the area as per the specialization of the program. The mastery should be at a level higher than the requirements in the appropriate bachelor program

PO4: An ability to apply stream knowledge to design or develop solutions for real world problems by following the standards

PO5: An ability to identify, select and apply appropriate techniques, resources and state-of-the-art tool to model, analyse and solve practical engineering problems.

PO6: An ability to engage in life-long learning for the design and development related to the stream related problems taking into consideration sustainability, societal, ethical and environmental aspects

PO7: An ability to develop cognitive load management skills related to project management and finance which focus on Entrepreneurship and Industry relevance.

Mapping of course outcomes with program outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7
CO 1	2			2	3	2	
CO 2	3			1	2		
CO 3	3			3	3	3	
CO 4	2		2	3	3		
CO 5	1		2	2	2	3	
CO6	2	3	3	3	3	3	2

Assessment Pattern

Bloom's Category	End Semester Examination
Apply	70%-80%
Analyze	30%-40%
Evaluate	
Create	

Mark distribution

Total Marks	CIE	ESE	ESE Duration
100	40	60	2.5 hours

Continuous Internal Evaluation Pattern:

Evaluation shall only be based on application, analysis or design based questions (for both internal and end semester examinations).

Continuous Internal Evaluation: 40 marks

i. Preparing a review article based on peer reviewed original publications (minimum 10 publications shall be referred) : 15 marks

ii. Course based task / Seminar/ Data collection and interpretation : 15 marks

iii. Test paper (1 number) : 10 marks

Test paper shall include minimum 80% of the syllabus.
Course based task/test paper questions shall be useful in the testing of knowledge, skills, comprehension, application, analysis, synthesis, evaluation and understanding of the students.

End Semester Examination Pattern:

The end semester examination will be conducted by the respective College.
There will be two parts; Part A and Part B.
Part A will contain 5 numerical/short answer questions with 1 question from each module, having 5 marks for each question. Students should answer all questions. Part B will contain 7 questions (such questions shall be useful in the testing of overall achievement and maturity of the students in a course, through long answer questions relating to theoretical/practical knowledge, derivations, problem solving and quantitative evaluation), with minimum one question from each module of which student should answer any five. Each question can carry 7 marks
Total duration of the examination will be 150 minutes.
Note: The marks obtained for the ESE for an elective course shall not exceed 20% over the average ESE mark % for the core courses. ESE marks awarded to a student for each elective course shall be normalized accordingly.
For example if the average end semester mark % for a core course is 40, then the maximum eligible mark % for an elective course is 40+20 = 60 %.

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. An email threat has been sent to the President at 4 am. The police succeeded in tracking the person down and they found that the IP address belonged to Bob. They reached his home around 8 am and seized the computer (computer was ON) and arrested Bob. Suppose you are the Cyber Forensic expert who reached the venue at 9.am and started the investigation.

   a. Identify the steps need to be taken and justify your answer by giving the reason for each step.
   b. List the methods used by the police to track Bob down?

2. How is data seizure relevant in computer forensics and evidence collection?

Course Outcome 2 (CO2):

1. After the Covid-19 virus lockdown, the American based laboratory found that someone had tampered with their central server and they think that one of the staff members did this using the company system. The Manager sends the computer to a CFS team for a forensic examination. How did the CFST go about conducting their examination?

Course Outcome 3(CO3):

1. A bank suspected an employee of downloading sensitive files from the bank's computer network using his bank laptop computer from home while on leave of absence. The bank sent the computer for a computer forensic examination. How can this issue be addressed by the professionals?
2. A cyber-attack happened in a Govt. Official server and the network has been breached. In this case how will the network forensics and investigation of the logs happen?

Course Outcome 4 (CO4):

1. Analyse the computer image verification and authentication.

2. When a network breach happens, what all are the obstacles in collecting digital evidence from the network?

Course Outcome 5 (CO5):

1. A new trainee joined a cyber-security firm. His trainer gave a talk about how to poison DNS. So what may be the techniques he has explained to him?

2. Explain the most commonly used cyber forensic tools for forensic investigation. Also write on the surveillance tools used for information warfare of the future.

Course Outcome 6 (CO6):

1. Develop a digital forensic investigation model with the idea of Venter that digital forensics investigation can be conducted by even non-technical persons

Syllabus

Module 1: Introduction to Computer Forensics

Computer Forensics Fundamentals, Types of Computer Forensics Technology – Types of Computer Forensics Systems-Data Recovery and Evidence Collection– Forensic duplication and preservation of DE, Understanding Computer Investigation.

Module 2: Evidence Data Gathering

Data Acquisition. - Data Recovery- Evidence Collection and Data Seizure - Duplication and Preservation of Digital Evidence.

Module 3: Investigations

Network Traffic , Web Attacks, Router Forensics, DoS Attacks and Internet Crime.

Module 4: Evidence Data Analysis

Discovery of Electronic Evidence - Identification of Data - Determining and Validating Forensics Data – Data Hiding Techniques – Performing Remote Acquisition– Cell Phone and Mobile Devices Forensics- Processing Crime and Incident Scenes.

Module 5: Forensics Tools and Case Studies

Working with Windows and DOS Systems. - Understanding File systems, Exploring Microsoft file structures, Examining NTFS disks, Understanding whole disk encryption, windows registry, Microsoft start-up tasks.

Current Computer Forensics Tools: Software/ Hardware Tools. Computer forensics investigation – A case study

Course Plan

No	Topic	No. of Lectures (40 Hours)
1	**Module 1: Introduction to Computer Forensics**	**8**
1.1	Computer Forensics Fundamentals	1
1.2	Types of Computer Forensics Technology	1
1.3	Types of Military Computer Forensic Technology	1
1.4	Types of Computer Forensics Systems	1
1.5	Intrusion Detection Systems	1

1.6	Data Recovery and Evidence Collection	1
1.7	Forensic duplication and preservation of Digital Evidence	1
1.8	Understanding Computer Investigation	1
2	**Module 2: Evidence Data Gathering**	**11**
2.1	Data Acquisition	1
2.2	Data Recovery	1
2.3	Evidence Collection	1
2.4	Why Collect Evidence, Collection Options Obstacles	1
2.5	Types of Evidence, The Rules of Evidence	1
2.6	Volatile Evidence, General Procedure	1
2.7	Collection and Archiving	1
2.8	Methods of Collection, Artifacts	1
2.9	Collection Steps	1
2.10	Data Seizure	1
2.11	Duplication of Digital Evidence	1
3	**Module 3: Investigations**	**6**
3.1	Network Traffic :Investigating Network Intrusions	1
3.2	Network Forensics and Investigating logs	1
3.3	Web Attacks	1
3.4	Router Forensics	1
3.5	DoS Attacks	1
3.6	Internet Crime	1
4	**Module 4: Evidence Data Analysis**	**7**
4.1	Discovery of Electronic Evidence	1
4.2	Identification of Data	1
4.3	Determining and Validating Forensics Data	1
4.4	Data Hiding Techniques	1
4.5	Performing Remote Acquisition	1
4.6	Cell Phone and Mobile Devices Forensics	1
4.7	Processing Crime and Incident Scenes	1
5	**Module 5: Forensics Tools and Case Studies**	**8**
5.1	Working with Windows and DOS Systems	1
5.2	Understanding File systems	1
5.3	Exploring Microsoft file structures	1
5.4	Examining NTFS disks	1
5.5	Understanding whole disk encryption, windows registry	1
5.6	Microsoft start-up tasks.	1
5.7	Current Computer Forensics Software/Hardware Tools	1
5.8	Case Study	1

References

1. Man Young Rhee, “Internet Security: Cryptographic Principles”, “Algorithms and protocols”, Wiley Publications, 2003.
2. Nelson, Phillips, Enfinger, Steuart, “Computer Forensics and Investigations”, Cengage Learning, India Edition, 2008.
3. John R.Vacca, “Computer Forensics”, Cengage Learning, 2005.
4. Richard E.Smith, “Internet Cryptography”, 3rd Edition Pearson Education, 2008.
5. Marjie T.Britz, “Computer Forensics and Cyber Crime”: An Introduction”, 3rd Edition, Prentice Hall, 2013.

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
R25CST2X5	SOFTWARE TESTING	PROGRAM ELECTIVE 4	3	0	0	3

Preamble: The course aims at introducing various concepts of software testing, including white box, black box, static and dynamic testing. Testing is an integral part of software development and the software has to be tested in all aspects to ensure trouble free execution in all environments. The learner will be able to use the various testing tools such as JUnit, Selenium, AFL and CBMC.

Course Outcomes: After the completion of the course the student will be able to

CO 1	Explain various concepts in software testing including black box testing and white box testing. (**Cognitive Knowledge Level: Understand**)
CO 2	Develop test cases in JUnit to verify the behaviour of independent components of a program.(**Cognitive Knowledge Level: Evaluate**)
CO 3	Use Selenium to automate tests across web browsers to verify that the application behaves as expected.(**Cognitive Knowledge Level: Evaluate**)
CO 4	Use American Fuzzy Lop (AFL) to efficiently increase the coverage of test cases and to automatically discover clean, interesting test cases that trigger new internal states in the targeted binary. (**Cognitive Knowledge Level: Analyze**)
CO5	Perform bounded model checking for C programs using the tool CBMC. (**Cognitive Knowledge Level: Evaluate**)
CO 6	Design, develop and implement solutions based on the concepts of software testing. (**Cognitive Knowledge Level: Create**)

Program Outcomes (PO)

Outcomes are the attributes that are to be demonstrated by a graduate after completing the course.

PO1: An ability to independently carry out research/investigation and development work in engineering and allied streams

PO2: An ability to communicate effectively, write and present technical reports on complex engineering activities by interacting with the engineering fraternity and with society at large.

PO3: An ability to demonstrate a degree of mastery over the area as per the specialization of the program. The mastery should be at a level higher than the requirements in the appropriate bachelor program

PO4: An ability to apply stream knowledge to design or develop solutions for real world problems by following the standards

PO5: An ability to identify, select and apply appropriate techniques, resources and state-of-the-art tool to model, analyse and solve practical engineering problems.

PO6: An ability to engage in life-long learning for the design and development related to the stream related problems taking into consideration sustainability, societal, ethical and environmental aspects

PO7: An ability to develop cognitive load management skills related to project management and finance which focus on Entrepreneurship and Industry relevance.

Mapping of course outcomes with program outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7
CO 1	1		3	2	1		
CO 2	2		3	3	3	1	
CO 3	2		3	3	3	1	
CO 4	3		3	2	3	2	
CO 5	3		3	2	3	2	
CO 6	3	2	3	3	3	2	2

Assessment Pattern

Bloom's Category	End Semester Examination
Apply	70%-80%
Analyze	30%-40%
Evaluate	
Create	

Mark distribution

Total Marks	CIE	ESE	ESE Duration
100	40	60	2.5 hours

Continuous Internal Evaluation Pattern:

Evaluation shall only be based on application, analysis or design-based questions (for both internal and end semester examinations).

Continuous Internal Evaluation: 40 marks

i. Preparing a review article based on peer reviewed original publications (minimum 10 publications shall be referred) : 15 marks

ii. Course based task / Seminar/ Data collection and interpretation : 15 marks

iii. Test paper (1 number) : 10 marks

Test paper shall include minimum 80% of the syllabus.
Course based task/test paper questions shall be useful in the testing of knowledge, skills, comprehension, application, analysis, synthesis, evaluation and understanding of the students.

End Semester Examination Pattern:

The end semester examination will be conducted by the respective College.
There will be two parts; Part A and Part B.
Part A will contain 5 numerical/short answer questions with 1 question from each module, having 5 marks for each question. Students should answer all questions. Part B will contain 7 questions (such questions shall be useful in the testing of overall achievement and maturity of the students in a course, through long answer questions relating to theoretical/practical knowledge, derivations, problem solving and quantitative evaluation), with minimum one question from each module of which student should answer any five. Each question can carry 7 marks
Total duration of the examination will be 150 minutes.
Note: The marks obtained for the ESE for an elective course shall not exceed 20% over the average ESE mark % for the core courses. ESE marks awarded to a student for each elective course shall be normalized accordingly.
For example, if the average end semester mark % for a core course is 40, then the maximum eligible mark % for an elective course is 40+20 = 60 %.

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Differentiate between black box testing and white box testing, static testing and dynamic testing.

Course Outcome 2(CO2):

1. Write a Data Driven Test Class for the following class in Java and explain?

```
publicclassCalculate {

    publicintsum(intvar1, intvar2) {

        System.out.println("Adding values: "+ var1 + " + "+ var2);

        returnvar1 + var2;
```

}

}

Course Outcome 3 (CO3):

1. Demonstrate the use of Selenium WebDriver for the following use case.

   Launch Chrome browser.
   Open KTU Login Page
   Click on the Login Button
   Close the web browser.

Course Outcome 4 (CO4):

1. This problem is a pen-and-paper problem, and is based on the AFL fuzzerpseudo-code. For simplicity assume that the program being tested has only one branch pair. Hence, the arrays **G** and **Shm** have only one entry (consisting of a single byte). Say at some point of the run of afl-fuzz the value in **G** is **0xEB**.
   a. Say a run of the program on a test input t1 occurs now, and the exact visit count of the branch-pair in this run (i.e., before rounding) is 16. Would this test input t1 be added to the queue Q or not? Justify your answer by showing the necessary calculations. If it would be added to Q, what would be the updated entry in G? (You can ignore the array R for this problem.)
   b. Consider the next test input t2. What is the smallest (exact) visitcount of the branch pair by t2 that would cause t2 to be not added to Q, and what is the smallest (exact) visit count of the branch pair by t2 that would cause t2 to be added to Q? Justify your answer.

Course Outcome 5 (CO5):

1. A buffer is a contiguously allocated chunk of memory, represented by an array or a pointer in C. Programs written in C do not provide automatic bounds checking on the buffer, which means a program can – accidentally or deliberately – write beyond a buffer. The following example is a perfectly valid C program (in the sense that a compiler compiles it without any errors):

```
intmain()
  {
        int buffer[10];
        buffer[20] = 10;
  }
```

   However, the write access to an address outside the allocated memory region can lead to unexpected behavior. In particular, such bugs can be exploited to overwrite the return address of a function, thus enabling the execution of arbitrary user-induced code.

   Detect the above problem using CBMC.

Course Outcome 6 (CO6):

1. Use Selenium to test the web portal of KTU and report errors and bugs.

Syllabus

Module 1: Testing Fundamentals

Introduction, black box testing, white box testing, static black box testing, dynamic black box testing, static white box testing, dynamic white box testing.

Module 2: Introduction to JUnit

Introduction to Unit Testing in Java, Test Class, Test Method, Assertions, JUnit Life Cycle API, Test Execution, JUnit Test Framework.

Module 3: Introduction to Selenium

Introduction, Selenium methods, Verification Point in Selenium, Shared UI Maps, Using functions, Using a configuration file, Data Driven Testing, UI Objects, Debugging, Exception Handling, Reporting, Batch Execution, Continuous Integration with Jenkins.

Module 4: Coverage Guided Fuzzing Using American Fuzzy Lop (AFL)

Basic test input generation, AFL as a grey box fuzzer, Characteristics of retained test cases, vulnerability detection, Measuring the code coverage, AFL algorithm.

Module 5: Bounded Model Checking with CBMC

Principles of BMC, The CBMC tool, Applications of BMC, CBMC Hands on.

Course Plan

No	Topic	No. of Lectures (37 hrs)
1	**Module 1: Testing Fundamentals**	**7**
1.1	Getting started, Black box and white box testing, static and dynamic testing	1
1.2	Static black box testing	1
1.3	Dynamic black box testing (Lecture 1)	1
1.4	Dynamic black box testing (Lecture 2)	1
1.5	Static white box testing	1
1.6	Dynamic white box testing (Lecture 1)	1
1.7	Dynamic white box testing (Lecture 1)	1
2	**Module 2: Introduction to JUnit**	**8**
2.1	Test Driven Development, Benefits of Unit Testing	1
2.2	JUnit Introduction, Java 8 Primer (Lambdas, Stream API, Optional <T>)	1
2.3	Project Setup, Writing the first test.	1
2.4	Test Class, Test Method	1
2.5	Assertions	1
2.6	JUnit Life Cycle API, Test Execution	1
2.7	Developing an application with JUnit 5 (lecture 1)	1

2.8	Developing an application with JUnit 5 (lecture 1)	1
3	**Module 3: Introduction to Selenium**	**8**
3.1	Introduction to Selenium, Installation, Using Selenium IDE, Managing User Interface Controls	1
3.2	Creating First Selenium Web Driver Script, Selenium Methods	1
3.3	Verification Point in Selenium, Shared UI Map	1
3.4	Using Functions, Using a Configuration File	1
3.5	Data Driven Testing – Parameterization, Synchronizing WebDriver Scripts	1
3.6	Handling Pop-up Dialogs and Multiple Windows, Working with Dynamic UI Objects	1
3.7	Debugging scripts, Exception Handling in WebDriver, Reporting in Selenium	1
3.8	Batch Execution, Continuous Integration with Jenkins	1
4	**Module 4: Coverage Guided Fuzzing Using American Fuzzy Lop (AFL)**	**7**
4.1	Basics of test input generation using fuzzing	1
4.2	AFL as grey box fuzzer - Detailed Demo of AFL tool	1
4.3	Characteristics of Retained test cases	1
4.4	Vulnerability detection using AFL	1
4.5	Measuring the code coverage	1
4.6	AFL Algorithm - Selection of a test input for fuzzing, Scoring the test input	1
4.7	AFL Algorithm - Fuzzing the test input, Retention/discarding the generated test input	1
5	**Module 5: Bounded Model Checking with CBMC**	**7**
5.1	Principles of BMC (Lecture 1)	1
5.2	Principles of BMC (Lecture 2)	1
5.3	The CBMC Tool	1
5.4	Applications of BMC – Program Debugging and Repair, Concurrency	1
5.5	Applications of BMC – Testing, Security	1
5.6	CBMC Hands on (Lecture 1)	1
5.7	CBMC Hands on (Lecture 2)	1

References

1. Ron Patton, Software Testing, Second Edition, Sams Publishing, Pearson Education, 2007. (For module 1)
2. Shekhar Gulati and Rahul Sharma, Java unit Testing with JUnit 5, Apress, https://doi.org/10.1007/978-1-4842-3015-2 (For module 2)
3. Navneesh Garg, Test Automation Using Selenium WebDriver with java, AdactIn Group Pty Ltd, 2014. (For module 3)
1. The AFL home page (for module 4), https://lcamtuf.coredump.cx/afl/
2. The CPROVER Manual (for module 5), http://www.cprover.org/cprover-manual/

3. A. Biere, A. Cimatti, E. Clarke, and Y. Zhu. Symbolic model checking without BDDs. In Proc. of the Workshop on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'99), LNCS. Springer-Verlag, 1999. (for module 5)
4. Biere, Armin; Cimatti, Alessandro; Clarke, Edmund M; Strichman, Ofer; Zhu, Yunshan (2018): Bounded Model Checking. Carnegie Mellon University. Journal contribution. https://doi.org/10.1184/R1/6603944.v1 (for module 5)
5. Edmund Clarke, Daniel Kroening & Flavio Lerda. A Tool for Checking ANSI-C Programs. TACAS 2004for module 5)
6. Rohan Bavishi, Awanish Pandey, Subhajit Roy. To be precise: regression aware debugging. OOPSLA 2016 for module 5)
7. Bernd Fischer, Omar Inverso, Gennaro Parlato. CSeq: A Concurrency Pre-processor for Sequential C Verification Tools. ASE 2013 for module 5)
8. Andreas Holzer, Christian Schallhart, Michael Tautschnig & Helmut Veith. Query-Driven Program Testing. VMCAI 2009 for module 5)
9. Vladimir Klebanov, Norbert Manthey & Christian Muise. SAT-Based Analysis and Quantification of Information Flow in Programs. QEST 2013 for module 5)

SEMESTER II

INTERDISCIPLINARY ELECTIVE

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
R25CST2X6	INTRODUCTION TO MACHINE LEARNING	INTERDISCIPLINARY ELECTIVE	3	0	0	3

Preamble: This course helps the learners to understand the concepts in Machine Learning. Students will be able to understand the basics of regression, classification and clustering. After completing this course students will be able to develop machine learning based solution for real world problems in multidisciplinary environments.

Course Outcomes: After the completion of the course the student will be able to

CO 1	Illustrate the concept, purpose, scope, steps, and applications of ML techniques. **(Knowledge level : Apply)**
CO 2	Understand the concepts of supervised, unsupervised and reinforcement learning to apply in real world problems. **(Knowledge level : Apply)**
CO 3	Illustrate the working of classifiers and clustering techniques for typical machine learning applications. **(Knowledge level : Apply)**
CO 4	Acquire skills to improve the performance of Machine Learning models using ensemble techniques. **(Knowledge level : Apply)**
CO5	Design and Implement solution for a real world problem using Machine Learning algorithms (Cognitive Knowledge Level: **Create**)

Program Outcomes (PO)

Outcomes are the attributes that are to be demonstrated by a graduate after completing the course.

PO1: An ability to independently carry out research/investigation and development work in engineering and allied streams

PO2: An ability to communicate effectively, write and present technical reports on complex engineering activities by interacting with the engineering fraternity and with society at large.

PO3: An ability to demonstrate a degree of mastery over the area as per the specialization of the program. The mastery should be at a level higher than the requirements in the appropriate bachelor program

PO4: An ability to apply stream knowledge to design or develop solutions for real world problems by following the standards

PO5: An ability to identify, select and apply appropriate techniques, resources and state-of-the-art tool to model, analyse and solve practical engineering problems.

PO6: An ability to engage in life-long learning for the design and development related to the stream related problems taking into consideration sustainability, societal, ethical and environmental aspects

PO7: An ability to develop cognitive load management skills related to project management and finance which focus on Entrepreneurship and Industry relevance.

Mapping of course outcomes with program outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7
CO 1	2		3	2	2		
CO 2	2		3	3	2		
CO 3	2		3	3	3		
CO 4	2		3	3	3		
CO5	3	2	3	3	3	2	2

Assessment Pattern

Bloom's Category	End Semester Examination
Apply	80%
Analyse	20%
Evaluate	
Create	

Mark distribution

Total Marks	CIE	ESE	ESE Duration
100	40	60	2.5 hours

Continuous Internal Evaluation Pattern:

Continuous Internal Evaluation : 40 marks

Micro project/Course based project : 20 marks

Course based task/Seminar/Quiz : 10 marks

Test paper, 1 no. : 10 marks

The project shall be done individually. Group projects not permitted.

Test paper shall include minimum 80% of the syllabus.

Course based task/test paper questions shall be useful in the testing of knowledge, skills, comprehension, application, analysis, synthesis, evaluation and understanding of the students.

End Semester Examination Pattern:

Total : 60 marks

The end semester examination will be conducted by the respective College.

There will be two parts; Part A and Part B.

Part A will contain 5 numerical/short answer questions with 1 question from each module, having 5 marks for each question. Students should answer all questions. Part B will contain 7 questions (such questions shall be useful in the testing of overall achievement and maturity of the students in a course, through long answer questions relating to theoretical/practical knowledge, derivations, problem solving and quantitative evaluation), with minimum one question from each module of which student should answer any five. Each question can carry 7 marks

Total duration of the examination will be 150 minutes.

Note: The marks obtained for the ESE for an elective course shall not exceed 20% over the average ESE mark % for the core courses. ESE marks awarded to a student for each elective course shall be normalized accordingly.

For example if the average end semester mark % for a core course is 40, then the maximum eligible mark % for an elective course is 40+20 = 60 %.

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Suppose 10000 patients get tested for flu; out of them, 9000 are actually healthy and 1000 are actually sick. For the sick people, a test was positive for 620 and negative for 380. For the healthy people, the same test was positive for 180 and negative for 8820. Construct a confusion matrix for the data and compute the precision and recall for the data.

2. Distinguish between supervised learning and Reinforcement learning. Illustrate with an example.

3. Discuss any four examples of machine learning applications.

Course Outcome 2 (CO2)

1. State the mathematical formulation of the SVM problem. Give an outline of the method for solving the problem.

2. Show the final result of hierarchical clustering with complete link by drawing a dendrogram.

Course Outcome 3(CO3):

1. Identify the first splitting attribute for the decision tree by using the ID3 algorithm with the following dataset.

2. Consider the training data in the following table where Play is a class attribute. In the table, the Humidity attribute has values “L” (for low) or “H” (for high), Sunny has values “Y” (for yes) or “N” (for no), Wind has values “S” (for strong) or “W” (for weak), and Play has values “Yes” or “No”.

   What is the class label for the following day (Humidity=L, Sunny=N, Wind=W), according to naïve Bayesian classification?

3. Explain DBSCAN algorithm for density based clustering. List out its advantages compared to K-means.

4. Explain how Support Vector Machine can be used for classification of linearly separable data.

5. Define Hidden Markov Model. What is meant by the evaluation problem and how is this solved?

6. Use K Means clustering to cluster the following data into two groups. Assume cluster centroid are m1=2 and m2=4. The distance function used is Euclidean distance. { 2, 4, 10, 12, 3, 20, 30, 11, 25 }

Course Outcome 4 (CO4):

1. Explain how the *Random Forests* give output for *Classification*, and *Regression* problems?
2. Is *Random Forest* an *Ensemble Algorithm*

3. Why is the training efficiency of Random Forest better than Bagging?

	Syllabus	
Module	**Contents**	**Hours**
I	Overview of machine learning: supervised, semi-supervised, unsupervised learning, reinforcement learning. Types of ML problems: Classification, Clustering and Regression, Cost functions: Definition and Types, Data PreProcessing, Bias-Variance trade off, Cross validation techniques, Classifier performance measures, ROC Curves	6
II	Introduction to neural network :_Linear Regression, Least square Gradients, Logistic Regression, Sigmoid function & differentiation, Logistic Regression – Regularization, Neural Networks – Concept of perceptron and Artificial neuron, Weight initialization techniques, Feed Forward Neural Network, Back Propagation algorithm	8
III	Classification Methods : Support Vector Machine, Optimal Separating hyper plane, Kernel trick, Kernel functions, Gaussian class conditional distribution, Bayes Rule, Naïve Bayes Model, Decision Tree – ID3, Maximum Likelihood estimation techniques	8
IV	Clustering Methods: K-means clustering , Hierarchical clustering techniques, Density Based clustering, Feature Selection techniques: Entropy, Correlation Coefficient, Chi-square Test, Forward & Backward Selection, Dimensionality Reduction: PCA, LDA, t-SNE	7
V	Basics of graphical models - Bayesian networks, Hidden Markov model, Ensemble methods – Boosting, Bagging, Random forest, XGBoost (Case study)	6

Lesson Plan

1	**Introduction to machine learning (Hours: 6)**	
1.1	Overview of machine learning: supervised, semi-supervised, unsupervised learning, reinforcement learning	1
1.2	Types of ML problems: Classification, Clustering and Regression	1
1.3	Cost functions: Definition and Types	1
1.4	Data PreProcessing, Bias-Variance trade off	1
1.5	Cross validation techniques	1
1.6	Classifier performance measures, ROC Curves	1
2	**Introduction to neural network (Hours: 8)**	
2.1	Linear Regression, Least square Gradients	1
2.2	Logistic Regression	1
2.3	Sigmoid function & differentiation	1
2.4	Logistic Regression - Regularization	
2.5	Neural Networks – Concept of perceptron and Artificial neuron	1
2.6	Weight initialization techniques	1
2.7	Feed Forward Neural Network	1
2.8	Back Propagation algorithm	1
3	**Classification Methods (Hours: 8)**	
3.1	Support Vector Machine	1
3.2	Optimal Separating hyper plane	1
3.3	Kernel trick, Kernel functions	1
3.4	Gaussian class conditional distribution	1
3.5	Bayes Rule	1
3.6	Naïve Bayes Model	1
3.7	Decision Tree – ID3,	1
3.8	Maximum Likelihood estimation techniques	1
4	**Clustering Methods (Hours: 7)**	
4.1	K-means clustering	1
4.2	Hierarchical clustering techniques	1
4.3	BIRCH	1
4.4	Density Based clustering	1
4.5	Feature Selection techniques: Entropy, Correlation Coefficient, Chi-square Test	1
4.6	Forward & Backward Selection	1
4.5	Dimensionality Reduction: PCA	1
4.6	LDA	1
4.7	t-SNE	1
5	**Basics of graphical models (Hours: 6)**	
5.1	Basics of graphical models - Bayesian networks	1
5.2	Hidden Markov model	1
5.3	Ensemble methods - Boosting	1
5.4	Bagging	1
5.5	Random forest	1
5.6	XGBoost (Case study)	1

Reference Books

1. Ethem Alpaydın, “Introduction to Machine Learning (Adaptive Computation and Machine Learning)”, MIT Press, 2004.

2. Kevin Murphy, Machine Learning: A Probabilistic Perspective (MLAPP), MIT Press, 2012

3. Han, Jiawei, and Micheline Kamber. Data Mining: Concepts and Techniques. San Francisco: Morgan Kaufmann Publishers

4. Christopher M. Bishop, “Pattern Recognition and Machine Learning”, Springer, 2006

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
R25CST2X6	DATA STRUCTURES	INTERDISCIPLINARY ELECTIVE	3	0	0	3

Preamble: The purpose of the syllabus is to create awareness about Data Structures and their applications. After the completion of the course, the learners should be able to either use existing data structures or design their own data structures to solve real world problems.

Course Outcomes: After the completion of the course the student will be able to

CO1	Design algorithms for a task and calculate the time complexity of that algorithm (**Cognitive Knowledge Level: Apply**)
CO2	Use arrays and linked lists for problem solving (Cognitive Knowledge Level: **Apply**)
CO3	Represent data using trees, graphs and manipulate them to solve computational problems. (**Cognitive Knowledge Level:Apply**)
CO4	Make use of appropriate sorting algorithms to order data based on the situation. (**Cognitive Knowledge Level: Apply**)
CO5	Design and Implement appropriate Data Structures for solving a real world problem (Cognitive Knowledge Level: **Create**)

Program Outcomes (PO)

Outcomes are the attributes that are to be demonstrated by a graduate after completing the course.

PO1: An ability to independently carry out research/investigation and development work in engineering and allied streams

PO2: An ability to communicate effectively, write and present technical reports on complex engineering activities by interacting with the engineering fraternity and with society at large.

PO3: An ability to demonstrate a degree of mastery over the area as per the specialization of the program. The mastery should be at a level higher than the requirements in the appropriate bachelor program

PO4: An ability to apply stream knowledge to design or develop solutions for real world problems by following the standards

PO5: An ability to identify, select and apply appropriate techniques, resources and state-of-the-art tool to model, analyse and solve practical engineering problems.

PO6: An ability to engage in life-long learning for the design and development related to the stream related problems taking into consideration sustainability, societal, ethical and environmental aspects

PO7: An ability to develop cognitive load management skills related to project management and finance which focus on Entrepreneurship and Industry relevance.

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7
CO1	2		3	3		2	
CO2	2		3	2		1	
CO3	2		3	3		2	
CO4	2		3	3		1	
CO5	3	2	3	3		2	2

Assessment Pattern

Bloom's Category	End Semester Examination
Apply	70%
Analyse	30%
Evaluate	
Create	

Mark distribution

Total Marks	CIE	ESE	ESE Duration
100	40	60	2.5 hours

Continuous Internal Evaluation Pattern:

Evaluation shall only be based on application, analysis or design based questions (for both internal and end semester examinations).

Continuous Internal Evaluation: 40 marks

i. Preparing a review article based on peer reviewed original publications (minimum 10 publications shall be referred) : 15 marks

ii. Course based task / Seminar/ Data collection and interpretation : 15 marks

iii. Test paper (1 number) : 10 marks

Test paper shall include minimum 80% of the syllabus.

Course based task/test paper questions shall be useful in the testing of knowledge, skills, comprehension, application, analysis, synthesis, evaluation and understanding of the students.

End Semester Examination Pattern:
The end semester examination will be conducted by the respective College.

There will be two parts; Part A and Part B.

Part A will contain 5 numerical/short answer questions with 1 question from each module, having 5 marks for each question. Students should answer all questions. Part B will contain 7 questions (such questions shall be useful in the testing of overall achievement and maturity of the students in a course, through long answer questions relating to theoretical/practical knowledge, derivations, problem solving and quantitative evaluation), with minimum one question from each module of which student should answer any five. Each question can carry 7 marks

Total duration of the examination will be 150 minutes.

Note: The marks obtained for the ESE for an elective course shall not exceed 20% over the average ESE mark % for the core courses. ESE marks awarded to a student for each elective course shall be normalized accordingly.
For example if the average end semester mark % for a core course is 40, then the maximum eligible mark % for an elective course is 40+20 = 60 %.

Sample Course Level Assessment Questions

Course Outcome1(CO1):

1. Write an algorithm for matrix multiplication and calculate its time complexity.
2. Define Big-O notation. Derive the Big – O notation for $5n^3+2n^2+3^n$.
3. Check whether the following is true or not.
   2n+1 is O(2n). Give reason.

Course Outcome 2(CO2):

1. How a linked list can be used to represent the polynomial

$5x^4y^6+24x^3y^4-17x^2y^3+15xy^2+45$.
Write a procedure to add two Bivariate polynomials represented using linked lists.

2. Write an algorithm/pseudocode to convert a given infix expression to postfix expression. Trace the steps involved in converting the given infix expression ((A +B)^C)-((D*C)/F) to postfix expression.

3. Let L1 be a singly linked list in memory. Write an algorithm that
   i) Finds the number of non zero elements in L1
   ii) Adds a given value K to each element in L1

Course Outcome 3(CO3):

1. Create a Binary Tree with the following sequence 14, 15, 4, 18, 9, 16, 20, 17, 3, 7, 5, 2 and perform inorder, preorder and postorder traversals on the above tree and print the output of the traversals.
2. In a complete binary tree of depth d (complete including last level), give an expression to find the number of leaf nodes.

Course Outcome 4(CO4):

1. Write an algorithm/pseudocode to sort elements using Heap sort technique. Illustrate the working of Heap sort algorithm on the following input : 35,15,0,1,60

2. With the help of an algorithm/pseudocode and suitable example, explain how you would perform binary search on an array of n elements. Find the time complexity of binary search algorithm.

3. Suppose an array contains elements {10, 13, 21, 32, 35, 44, 55}. Give the steps to find an element "35" using i) linear search ii) binary search

Course Outcome 5(CO5):

Design a reservation system for railways that includes a waiting list. If the reservation is full, display "reservation full" and put them in the waiting list and give a waiting list number. If a passenger wishes to cancel his ticket, he may do it any time. Then the passenger at the front of the waiting list is allotted a berth automatically.

	Syllabus	
Module	**Contents**	**Hours**
I	**Basic Concepts of Data Structures** System Life Cycle, Algorithms, Performance Analysis, Space complexity, Time Complexity, Asymptotic Notation, Complexity Calculation of Simple Algorithms	5
II	**Arrays and Searching** Polynomial representation using Arrays, Sparse matrix, Stacks, Queues, Circular Queues, Priority Queues, Double Ended Queues, Evaluation of Expressions, Linear Search and Binary Search	9
III	**Linked List and Memory Management** Self Referential Structures, Dynamic Memory Allocation, Singly Linked List-Operations on Linked List, Doubly Linked List, Circular Linked List, Stacks and Queues using Linked List, Polynomial representation using Linked List, Memory allocation and de-allocation,	8
IV	Trees, Binary Trees-Tree Operations, Binary Tree Representation, Tree Traversals, Graphs- Representation of Graphs, Depth First Search and Breadth First Search on Graphs, Applications of Graphs	7
V	Searching Techniques – Linear search , Binary search Sorting Techniques – Selection Sort, Insertion Sort, Quick Sort, Merge Sort and Heap Sort	5

Course Plan

Module 1 :Basic Concepts of Data Structures		(5 hours)
1.1	System Life Cycle,	1 hour
1.2	Algorithms	1 hour
1.3	Performance Analysis, Space Complexity, Time Complexity,	1 hours
1.4	Asymptotic Notation	1hour
1.5	Complexity Calculation of Simple Algorithms	1hour
Module 2 :Arrays and Searching		(9 hours)
2.1	Polynomial representation using Arrays	1 hour
2.2	Sparse matrix	1 hours
2.3	Stacks	1 hour
2.4	Queues, Circular Queues	1 hour
2.5	Priority Queues,	1 hour
2.6	Double Ended Queue	1 hours
2.7	Evaluation of Expressions	1 hour
2.8	Linear Search	1 hour
2.9	Binary Search	1 hour
Module 3 : Linked List and Memory Management		(8 hours)
3.1	Self Referential Structures	1 hour
3.2	Dynamic Memory Allocation	1 hour
3.3	Single Linked List-Operations on Linked List,	1 hour
3.4	Double Linked List	1 hour
3.5	Circular Linked List	1 hour
3.6	Stacks and Queues using Linked List	1 hour
3.7	Polynomial representation using Linked List	1 hour
3.8	Memory de-allocation	1 hour
Module 4 :Trees and Graphs		(7 hours)
4.1	Trees, Binary Trees	1hour
4.2	Tree Operations, Binary Tree Representation,	1hour
4.3	Tree Traversals	1hour
4.4	Graphs	1hour
4.5	Representation of Graphs	1hour
4.6	Depth First Search and Breadth First Search on Graphs	1hour

4.7	Applications of Graphs	1hour
Module 5 : Sorting and Hashing		(5 hours)
5.1	Sorting Techniques – Selection Sort	1hour
5.2	Insertion Sort	1hour
5.3	Quick Sort	1hour
5.4	Merge Sort	1hour
5.5	Heap Sort	1hour

Text Book

1. Ellis Horowitz, Sartaj Sahni and Susan Anderson-Freed, Universities Press, Fundamentals of Data Structures in C

Reference Books

1. Samanta D., Classic Data Structures, Prentice Hall India, 2/e, 2009.
2. Richard F. Gilberg, Behrouz A. Forouzan, Data Structures: A Pseudocode Approach with C, 2/e, Cengage Learning, 2005
3. Aho A. V., J. E. Hopcroft and J. D. Ullman, Data Structures and Algorithms, Pearson Publication,1983.
4. Tremblay J. P. and P. G. Sorenson, Introduction to Data Structures with Applications, Tata McGraw Hill, 1995.
5. Peter Brass, Advanced Data Structures, Cambridge University Press, 2008
6. Lipschuts S., Theory and Problems of Data Structures, Schaum's Series, 1986.
7. Wirth N., Algorithms + Data Structures = Programs, Prentice Hall, 2004.
8. Hugges J. K. and J. I. Michtm, A Structured Approach to Programming, PHI, 1987.
9. Martin Barrett, Clifford Wagner, And Unix: Tools For Software Design, John Wiley, 2008 reprint

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
R25CST2X6	SOFTWARE PROJECT MANAGEMENT	INTERDISCIPLINARY ELECTIVE	3	0	0	3

Preamble: This course provides fundamental knowledge in the Software Development Process. It covers Software Development, Quality Assurance, Project Management concepts and technology trends. This course enables the learners to apply state of the art industry practices in Software development.

Course Outcomes: After the completion of the course the student will be able to

CO1	Demonstrate Traditional and Agile Software Development approaches **(Cognitive Knowledge Level: Apply)**
CO2	Prepare Software Requirement Specification and Software Design for a given problem. **(Cognitive Knowledge Level: Apply)**
CO3	Justify the significance of design patterns and licensing terms in software development, prepare testing, maintenance and DevOps strategies for a project. **(Cognitive Knowledge Level: Apply)**
CO4	Make use of software project management concepts while planning, estimation, scheduling, tracking and change management of a project, with a traditional/agile framework**. (Cognitive Knowledge Level: Apply)**
CO5	Utilize SQA practices, Process Improvement techniques and Technology advancements in cloud based software models and containers & microservices. **(Cognitive Knowledge Level: Apply)**

Program Outcomes (PO)

Outcomes are the attributes that are to be demonstrated by a graduate after completing the course.

PO1: An ability to independently carry out research/investigation and development work in engineering and allied streams

PO2: An ability to communicate effectively, write and present technical reports on complex engineering activities by interacting with the engineering fraternity and with society at large.

PO3: An ability to demonstrate a degree of mastery over the area as per the specialization of the program. The mastery should be at a level higher than the requirements in the appropriate bachelor program

PO4: An ability to apply stream knowledge to design or develop solutions for real world problems by following the standards

PO5: An ability to identify, select and apply appropriate techniques, resources and state-of-the-art tool to model, analyse and solve practical engineering problems.

PO6: An ability to engage in life-long learning for the design and development related to the stream related problems taking into consideration sustainability, societal, ethical and environmental aspects

PO7: An ability to develop cognitive load management skills related to project management and finance which focus on Entrepreneurship and Industry relevance.

Mapping of course outcomes with program outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7
CO 1	2		3	2		2	
CO 2	2		3	3		1	
CO 3	2		3	3		2	
CO 4	2		3	3		1	
CO 5	2		3	2		2	

Assessment Pattern

Bloom's Category	End Semester Examination
Apply	70%
Analyse	30%
Evaluate	
Create	

Mark distribution

Total Marks	CIE	ESE	ESE Duration
100	40	60	2.5 hours

Continuous Internal Evaluation Pattern:

Evaluation shall only be based on application, analysis or design based questions (for both internal and end semester examinations).

Continuous Internal Evaluation: 40 marks

i. Preparing a review article based on peer reviewed original publications (minimum 10 publications shall be referred) : 15 marks

ii. Course based task / Seminar/ Data collection and interpretation : 15 marks

iii. Test paper (1 number) : 10 marks

Test paper shall include minimum 80% of the syllabus.

Course based task/test paper questions shall be useful in the testing of knowledge, skills, comprehension, application, analysis, synthesis, evaluation and understanding of the students.

End Semester Examination Pattern:

The end semester examination will be conducted by the respective College.

There will be two parts; Part A and Part B.

Part A will contain 5 numerical/short answer questions with 1 question from each module, having 5 marks for each question. Students should answer all questions. Part B will contain 7 questions (such questions shall be useful in the testing of overall achievement and maturity of the students in a course, through long answer questions relating to theoretical/practical knowledge, derivations, problem solving and quantitative evaluation), with minimum one question from each module of which student should answer any five. Each question can carry 7 marks

Total duration of the examination will be 150 minutes.

Note: The marks obtained for the ESE for an elective course shall not exceed 20% over the average ESE mark % for the core courses. ESE marks awarded to a student for each elective course shall be normalized accordingly.

For example if the average end semester mark % for a core course is 40, then the maximum eligible mark % for an elective course is 40+20 = 60 %.

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. What are the advantages of an incremental development model over a waterfall model?
2. Illustrate how the process differs in agile software development and traditional software development with a socially relevant case study. (Assignment question)

Course Outcome 2 (CO2):

1. How to prepare a software requirement specification?
2. Differentiate between Architectural design and Component level design.
3. How does agile approaches help software developers to capture and define the user requirements effectively?
4. What is the relevance of the SRS specification in software development?
5. Prepare a use case diagram for a library management system.

Course Outcome 3 (CO3):

1. Differentiate between the different types of software testing strategies.
2. Justify the need for DevOps practices?
3. How do design patterns help software architects communicate the design of a complex system effectively?
4. What are the proactive approaches one can take to optimise efforts in the testing phase?

Course Outcome 4 (CO4):

1. Illustrate the activities involved in software project management for a socially relevant problem?
2. How do SCRUM, Kanban and Lean methodologies help software project management?
3. Is rolling level planning in software project management beneficial? Justify your answer.
4. How would you assess the risks in your software development project? Explain how you can manage identified risks?

Course Outcome 5 (CO5):

1. Justify the importance of Software Process improvement?
2. Explain the benefits of cloud based software development, containers and microservices.
3. Give the role of retrospectives in improving the software development process.
4. Illustrate the use of project history data as a prediction tool to plan future socially relevant projects.

Syllabus

Module 1 : Introduction to Software Engineering (7 hours)

Introduction to Software Engineering - Professional software development, Software engineering ethics. Software process models - The waterfall model, Incremental development. Process activities - Software specification, Software design and implementation, Software validation, Software evolution. Coping with change - Prototyping, Incremental delivery, Boehm's Spiral Model. Agile software development - Agile methods, agile manifesto - values and principles. Agile development techniques, Agile Project Management. Case studies : An insulin pump control system. Mentcare - a patient information system for mental health care.

Module 2 : Requirement Analysis and Design (8 hours)

Functional and non-functional requirements, Requirements engineering processes. Requirements elicitation, Requirements validation, Requirements change, Traceability Matrix. Developing use cases, Software Requirements Specification Template, Personas, Scenarios, User stories, Feature identification. Design concepts - Design within the context of software engineering, Design Process, Design concepts, Design Model. Architectural Design - Software Architecture, Architectural Styles, Architectural considerations, Architectural Design Component level design - What is a component?, Designing Class-Based Components, Conducting Component level design, Component level design for web-apps. Template of a Design Document as per "IEEE Std 1016-2009 IEEE Standard for Information Technology Systems Design Software Design Descriptions". Case study: The Ariane 5 launcher failure.

Module 3 : Implementation and Testing (9 hours)

Object-oriented design using the UML, Design patterns, Implementation issues, Open-source development - Open-source licensing - GPL, LGPL, BSD. Review Techniques - Cost impact of Software Defects, Code review and statistical analysis. Informal Review, Formal Technical Reviews, Post-mortem evaluations. Software testing strategies - Unit Testing, Integration Testing, Validation testing, System testing, Debugging, White box testing, Path testing, Control Structure testing, Black box testing, Testing Documentation and Help facilities. Test automation, Test-driven development, Security testing. Overview of DevOps and Code Management - Code management, DevOps automation, Continuous Integration, Delivery, and Deployment (CI/CD/CD). Software Evolution - Evolution processes, Software maintenance.

Module 4 : Software Project Management (6 hours)

Software Project Management - Risk management, Managing people, Teamwork. Project Planning, Software pricing, Plan-driven development, Project scheduling, Agile planning. Estimation techniques, COCOMO cost modeling. Configuration management, Version management, System building, Change management, Release management, Agile software management - SCRUM framework. Kanban methodology and lean approaches.

Module 5 : Software Quality, Process Improvement and Technology trends (6 hours)

Software Quality, Software Quality Dilemma, Achieving Software Quality Elements of Software Quality Assurance, SQA Tasks , Software measurement and metrics. Software Process Improvement(SPI), SPI Process CMMI process improvement framework, ISO 9001:2000 for Software. Cloud-based Software - Virtualisation and containers, Everything as a service(IaaS, PaaS), Software as a service. Microservices Architecture - Microservices, Microservices architecture, Microservice deployment.

Text Books

1. Book 1 - Ian Sommerville, Software Engineering, Pearson Education, Tenth edition, 2015.
2. Book 2 - Roger S. Pressman, Software Engineering : A practitioner's approach, McGraw Hill publication, Eighth edition, 2014
3. Book 3 - Ian Sommerville, Engineering Software Products: An Introduction to Modern Software Engineering, Pearson Education, First Edition, 2020.

References

2. IEEE Std 830-1998 - IEEE Recommended Practice for Software Requirements SpeciÞcations
3. IEEE Std 1016-2009 IEEE Standard for Information Technology—Systems Design— Software Design Descriptions
4. David J. Anderson, Kanban, Blue Hole Press 2010
5. David J. Anderson, Agile Management for Software Engineering, Pearson, 2003
6. Walker Royce, Software Project Management : A unified framework, Pearson Education, 1998
7. Steve. Denning, The age of agile, how smart companies are transforming the way work gets done. New York, Amacom, 2018.
8. Satya Nadella, Hit Refresh: The Quest to Rediscover Microsoft's Soul and Imagine a Better Future for Everyone, Harper Business, 2017
9. Henrico Dolfing, Project Failure Case Studies: Lessons learned from other people's mistakes, Kindle edition

10. Mary Poppendieck, Implementing Lean Software Development: From Concept to Cash, Addison-Wesley Signature Series, 2006
11. StarUML documentation - https://docs.staruml.io/
12. OpenProject documentation - https://docs.openproject.org/
13. BugZilla documentation - https://www.bugzilla.org/docs/
14. GitHub documentation - https://guides.github.com/
15. Jira documentation - https://www.atlassian.com/software/jira

Teaching Plan

No	Contents	No of Lecture Hrs
	Module 1 : Introduction to Software Engineering (7 hours)	
1.1	Introduction to Software Engineering.[Book 1, Chapter 1]	1 hour
1.2	Software process models [Book 1 - Chapter 2]	1 hour
1.3	Process activities [Book 1 - Chapter 2]	1 hour
1.4	Coping with change [Book 1 - Chapter 2, Book 2 - Chapter 4]	1 hour
1.5	Case studies : An insulin pump control system. Mentcare - a patient information system for mental health care. [Book 1 - Chapter 1]	1 hour
1.6	Agile software development [Book 1 - Chapter 3]	1 hour
1.7	Agile development techniques, Agile Project Management.[Book 1 - Chapter 3]	1 hour
	Module 2 : Requirement Analysis and Design (8 hours)	
2.1	Functional and non-functional requirements, Requirements engineering processes [Book 1 - Chapter 4]	1 hour

2.2	Requirements elicitation, Requirements validation, Requirements change, Traceability Matrix [Book 1 - Chapter 4]	1 hour
2.3	Developing use cases, Software Requirements Specification Template [Book 2 - Chapter 8]	1 hour
2.4	Personas, Scenarios, User stories, Feature identification [Book 3 - Chapter 3]	1 hour
2.5	Design concepts [Book 2 - Chapter 12]	1 hour
2.6	Architectural Design [Book 2 - Chapter 13]	1 hour
2.7	Component level design [Book 2 - Chapter 14]	1 hour
2.8	Design Document Template. Case study: The Ariane 5 launcher failure. [Ref - 2, Book 2 - Chapter 16]	1 hour
Module 3 : Implementation and Testing (9 hours)		
3.1	Object-oriented design using the UML, Design patterns [Book 1 - Chapter 7]	1 hour
3.2	Implementation issues, Open-source development - Open-source licensing - GPL, LGPL, BSD [Book 1 - Chapter 7]	1 hour
3.3	Review Techniques - Cost impact of Software Defects, Code review and statistical analysis. [Book 2 - Chapter 20]	1 hour
34	Informal Review, Formal Technical Reviews, Post-mortem evaluations. [Book 2 - Chapter 20]	1 hour
3.5	Software testing strategies - Unit Testing, Integration Testing, Validation testing, System testing and Debugging (basic concepts only). [Book 2 - Chapter 22]	1 hour
3.6	White box testing, Path testing, Control Structure testing, Black box testing. Test documentation [Book 2 - Chapter 23]	1 hour
3.7	Test automation, Test-driven development, Security testing. [Book 3 - Chapter 9]	1 hour
3.8	DevOps and Code Management - Code management, DevOps automation, CI/CD/CD. [Book 3 - Chapter 10]	1 hour
3.9	Software Evolution - Evolution processes, Software maintenance. [Book 1 - Chapter 9]	1 hour
Module 4 : Software Project Management (6 hours)		

4.1	Software Project Management - Risk management, Managing people, Teamwork [Book 1 - Chapter 22]	1 hour
4.2	Project Planning - Software pricing, Plan-driven development, Project scheduling, Agile planning [Book 1 - Chapter 23]	1 hour
4.3	Estimation techniques [Book 1 - Chapter 23]	1 hour
4.4	Configuration management [Book 1 - Chapter 25]	1 hour
4.5	Agile software management - SCRUM framework [Book 2 - Chapter 5]	1 hour
4.6	Kanban methodology and lean approaches.[Ref 9 - Chapter 2]	1 hour
Module 5 : Software Quality, Process Improvement and Technology trends (6 hours)		
5.1	Software Quality, Software Quality Dilemma, Achieving Software Quality. [Book 2 - Chapter 19]	1 hour
5.2	Elements of Software Quality Assurance, SQA Tasks , Software measurement and metrics. [Book 3 - Chapter 21]	1 hour
5.3	Software Process Improvement (SPI), SPI Process [Book 2 - Chapter 37]	1 hour
5.4	CMMI process improvement framework, ISO 9001:2000 for Software. [Book 2 - Chapter 37]	1 hour
5.5	Cloud-based Software - Virtualisation and containers, IaaS, PaaS, SaaS.[Book 3 - Chapter 5]	1 hour
5.6	Microservices Architecture - Microservices, Microservices architecture, Microservice deployment [Book 3 - Chapter 6]	1 hour