CURRICULUM

FOR

B. TECH DEGREE PROGRAMME

IN

COMPUTER SCIENCE AND ENGINEERING

2020 SCHEME (Revised in 2022)

(AUTONOMOUS)

MAR BASELIOS COLLEGE OF ENGINEERING AND TECHNOLOGY
(Approved by AICTE, Autonomous Institution Affiliated to APJ Abdul Kalam Technological University)
MAR IVANIOS VIDYANAGAR, NALANCHIRA, THIRUVANANTHAPURAM – 695015, KERALA.
Phone: 0471 2545866
Fax: 0471 2545869
Web: www.mbcet.ac.in
email: hodcs@mbcet.ac.in

CURRICULUM AND DETAILED SYLLABI

FOR

B. TECH DEGREE PROGRAMME

IN

COMPUTER SCIENCE AND ENGINEERING

2020 SCHEME (Revised in 2022)

(AUTONOMOUS)

MAR BASELIOS COLLEGE OF ENGINEERING AND TECHNOLOGY
(Approved by AICTE, Autonomous Institution Affiliated to APJ Abdul Kalam Technological University)
MAR IVANIOS VIDYANAGAR, NALANCHIRA, THIRUVANANTHAPURAM – 695015, KERALA.
Phone: 0471 2545866
Fax: 0471 2545869
Web: www.mbcet.ac.in
email: hodcs@mbcet.ac.in

MAR BASELIOS COLLEGE OF ENGINEERING AND TECHNOLOGY

Vision and Mission of the Institution

Vision:
To be an Institution moulding globally competent professionals as epitomes of Noble Values.

Mission:
To transform the Youth as technically competent, ethically sound and socially committed professionals, by providing a vibrant learning ambience for the welfare of humanity.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Vision and Mission of the Department

Vision:
To be a Centre of Excellence in Computer Science and Engineering providing quality education and research for the betterment of the society.

Mission:
To impart sound knowledge in theoretical and applied foundations of Computer Science and Engineering, and to train the students to solve real life issues to effectively define and shape life.

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)

1. Graduates will be successful professionals in Industries of core or interdisciplinary nature or entrepreneurs, demonstrating effective leadership and excellent team work.
2. Graduates will expand the horizon of knowledge through higher education or research, leading to self-directed professional development.
3. Graduates will demonstrate professional attitude and ethics while providing solutions in societal and environmental contexts.

PROGRAMME OUTCOMES (POs)

Engineering graduates will be able to:

1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAMME SPECIFIC OUTCOMES (PSOs)

Engineering Graduates will have the ability to:

1. Apply Algorithmic Principles, Programming Skills and Software Engineering Principles to design, develop and evaluate Software Systems of varying complexities.
2. Apply knowledge of System Integration to design and implement computer-based systems.
3. Solve real world and socially relevant problems with the knowledge in recent and advanced Computing Technologies.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

For the students admitted from 2022-23

Scheduling of Courses

i) Knowledge Segments and Credits

Every course of B. Tech Programme is placed in one of the nine categories as listed in the following table.No semester shall have more than six lecture-based courses and two laboratory courses, and/ordrawing/seminar/project courses in the curriculum.

Sl. No.	Category	Category Code	Total credits
1	Humanities and Social Sciences including Management Courses	HSC	5
2	Basic Science Courses	BSC	26
3	Engineering Science Courses	ESC	22
4	Programme Core Courses, Comprehensive Course Work and Viva Voce	PCC	79
5	Programme Elective Courses	PEC	15
6	Open Elective Courses	OEC	3
7	Project Work and Seminar	PWS	10
8	Mandatory Non-credit Courses (P/F) with Grade	MNC	---
9	Mandatory Student Activities (P/F)	MSA	2
Total Mandatory Credits			**162**
	Value Added Courses (Optional) – Honours/Minor	VAC	20

ii) Semester-wise Credit Distribution

Semester	**I**	**II**	**III**	**IV**	**V**	**VI**	**VII**	**VIII**	***Total Credits***
Credits for Courses	17	21	22	22	23	23	15	17	**160**
Activity Points (Min.)	40				60				**100**
Credits for Activities	2								**2**
Total Credits									**162**
Value Added Courses (Optional) – Honours / Minor									**20**
Total Credits									**182**

SEMESTER V						
Slot	**Category Code**	**Course Number**	**Courses**	**L-T-P**	**Hours**	**Credit**
A	PCC	CS1U30J	Algorithm Analysis and Design	3-1-0	4	4
B	PCC	CS1U30C	System Software	3-1-0	4	4
C	PCC	CS1U30K	Operating Systems	3-1-0	4	4
D	PCC	CS1U30D	Microprocessors and Microcontrollers	3-1-0	4	4
E	PCC	CS1U30E	Management of Software Systems	3-0-0	3	3
F	MNC	NC0U30A	Disaster Management	2-0-0	2	---
S	PCC	CS1U38A	System Software and Microprocessors Lab	0-0-4	4	2
T	PCC	CS1U38D	Operating Systems Lab	0-0-4	4	2
R/M/H	VAC		Remedial/Minor/Honours Course	3-1-0	4	4
TOTAL					**29/33**	**23/27**

SEMESTER VI						
Slot	**Category Code**	**Course Number**	**Courses**	**L-T-P**	**Hours**	**Credit**
A	PCC	CS1U30F	Compiler Design	3-1-0	4	4
B	PCC	CS1U30G	Computer Graphics and Image Processing	3-1-0	4	4
C	PCC	CS1U30B	Computer Networks	3-1-0	4	4
D	PEC	CS1UXXX	Programme Elective I	2-1-0	3	3
E	HSC	HS0U30A	Industrial Economics and Foreign Trade	3-0-0	3	3
F	PCC	CS1U30I	Comprehensive Course Work	1-0-0	1	1
S	PCC	CS1U38C	Networking Lab	0-0-3	3	2
T	PWS	CS1U39A	Mini Project	0-0-3	3	2
R/M/H	VAC		Remedial/Minor/Honours Course	3-1-0	4	4
TOTAL					**25/29**	**23/27**

PROGRAMME ELECTIVE I

Slot	Category Code	Course Number	Courses	L-T-P	Hours	Credit
D	PEC	CS1U31A	Foundations of Machine Learning	2-1-0	3	3
		CS1U31B	Data Analytics	2-1-0	3	3
		CS1U31C	Foundations of Security in Computing	2-1-0	3	3
		CS1U31D	Automated Verification	2-1-0	3	3
		CS1U31E	Programming in Python	2-1-0	3	3
		CS1U31F	Advanced Data Communication	2-1-0	3	3
		CS1U31G	Applied Data Science with Python	2-0-2	4	3

MINOR

Semester	BASKET I Specialization: SOFTWARE ENGINEERING				BASKET II Specialization: MACHINE LEARNING				BASKET III Specialization: NETWORKING			
	Course Number	Course	L-T-P	Credit	Course Number	Course	L-T-P	Credit	Course Number	Course	L-T-P	Credit
S3	CS0M 20A	Object Oriented Programming	3-1-0	4	CS0M 20B	Python for Machine Learning	3-1-0	4	CS0M 20C	Data Communication	3-1-0	4
S4	CS0M 20D	Programming Methodologies	3-1-0	4	CS0M 20E	Mathematics for Machine Learning	3-1-0	4	CS0M 20F	Introduction to Computer Networks	3-1-0	4
S5	CS0M 30A	Concepts in Software Engineering	3-1-0	4	CS0M 30B	Concepts in Machine Learning	3-1-0	4	CS0M 30C	Client Server Systems	3-1-0	4
S6	CS0M 30D	Introduction to Software Testing	3-1-0	4	CS0M 30E	Concepts in Deep Learning	3-1-0	4	CS0M 30F	Wireless Networks and IoT Applications	3-1-0	4
S7	CS0M 49A	Mini Project	0-1-6	4	CS0M 49A	Mini Project	0-1-6	4	CS0M 49A	Mini Project	0-1-6	4
S8	CS0M 49B	Mini Project	0-1-6	4	CS0M 49B	Mini Project	0-1-6	4	CS0M 49B	Mini Project	0-1-6	4

HONOURS

Semester	BASKET I Specialization: SECURITY IN COMPUTING				BASKET II Specialization: MACHINE LEARNING				BASKET III Specialization: FORMAL METHODS			
	Course Number	Course	L-T-P	Credit	Course Number	Course	L-T-P	Credit	Course Number	Course	L-T-P	Credit
S4	CS1H 20A	Number Theory	3-1-0	4	CS1H 20B	Computational Fundamentals of Machine Learning	3-1-0	4	CS1H 20C	Principles of Program Analysis and Verification	3-1-0	4
S5	CS1H 30A	Cryptographic Algorithms	3-1-0	4	CS1H 30B	Neural Networks and Deep Learning	3-1-0	4	CS1H 30C	Principles of Model Checking	3-1-0	4
S6	CS1H 30D	Network Security	3-1-0	4	CS1H 30E	Advanced Topics in Machine Learning	3-1-0	4	CS1H 30F	Theory of Computability and Complexity	3-1-0	4
S7	CS1H 40A	Cyber Forensics	3-1-0	4	CS1H 40B	Advanced Topics in Artificial Intelligence	3-1-0	4	CS1H 40C	Logic for Computer Science	3-1-0	4
S8	CS1H 49A	Mini Project	0-1-6	4	CS1H 49A	Mini Project	0-1-6	4	CS1H 49A	Mini Project	0-1-6	4

SEMESTER V

CS1U30J	ALGORITHM ANALYSIS AND DESIGN	Category	L	T	P	Credit	Revised in 2022
		PCC	3	1	0	4	

PRE-REQUISITE: CS1U20A Data Structures

COURSE OVERVIEW: The course introduces students to the design of computer algorithms, as well as analysis of algorithms. Algorithm design and analysis provide the theoretical backbone of computer science and are a must in the daily work of the successful programmer. The goal of this course is to provide a solid background in the design and analysis of the major classes of algorithms. At the end of the course students will be able 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:

CO1	Analyze any given algorithm and express its time and space complexities in asymptotic notations.	Analyze
CO2	Derive recurrence equations and solve it using Iteration, Recurrence Tree, Substitution and Master's Methods to compute time complexity of algorithms.	Apply
CO3	Apply Graph algorithms and Advanced Data structures like AVL trees and Disjoint set operations in real world scenario.	Apply
CO4	Demonstrate Divide-and-conquer, Greedy Strategy, Dynamic programming, Branch-and Bound and Backtracking algorithm design techniques.	Understand
CO5	Classify a problem as computationally tractable or intractable, and discuss strategies to address intractability .	Understand
CO6	Identify the suitable design strategy to solve a given problem.	Apply

SYLLABUS

Introduction to algorithm analysis, Advanced data structures and graph algorithms, Various algorithm design techniques, Introduction to complexity theory and approximation algorithms.

TEXT BOOKS

1) T.H.Cormen, C.E.Leiserson, R.L.Rivest, C. Stein, Introduction to Algorithms, 2nd Edition, Prentice-Hall India (2001)

2) Ellis Horowitz, SartajSahni, Sanguthevar Rajasekaran, "Fundamentals of Computer Algorithms", 2nd Edition, Orient Longman Universities Press (2008)

3) Sara Baase and Allen Van Gelder —Computer Algorithms, Introduction to Design and Analysis, 3rd Edition, Pearson Education (2009)

REFERENCES

1) Jon Kleinberg, Eva Tardos, " Algorithm Design", First Edition, Pearson(2005)
2) Robert Sedgewick, Kevin Wayne, "Algorithms", 4th Edition Pearson(2011)
3) Gilles Brassard, Paul Brately, " Fundamentals of Algorithmics", Pearson(1996)
4) Steven S Sjiena, " The Algorithm Design Manual", 2nd Edition, Springer(2008)

COURSE PLAN

Module	Contents	No. of hours
I	Characteristics of Algorithms, Criteria for Analysing Algorithms, Time and Space Complexity - Best, Worst and Average Case Complexities, Asymptotic Notations - Big-Oh (O), Big- Omega (Ω), Big-Theta (Θ), Little-oh (o) and Little- Omega (ω) and their properties. Classifying functions by their asymptotic growth rate, Time and Space Complexity Calculation of simple algorithms. Analysis of Recursive Algorithms: Recurrence Equations, Solving Recurrence Equations – Iteration Method, Recursion Tree Method, Substitution method and Master's Theorem (Proof not required).	13
II	Self Balancing Tree - AVL Trees (Insertion and deletion operations with all rotations in detail, algorithms not expected); Disjoint Sets- Disjoint set operations, Union and find algorithms. DFS and BFS traversals - Analysis, Strongly Connected Components of a Directed graph, Topological Sorting.	12
III	The Control Abstraction of Divide and Conquer- 2-way Merge sort, Strassen's Algorithm for Matrix Multiplication-Analysis. The Control Abstraction of Greedy Strategy- Fractional Knapsack Problem, Minimum Cost Spanning Tree Computation- Kruskal's Algorithms - Analysis, Single Source Shortest Path Algorithm - Dijkstra's Algorithm-Analysis.	12
IV	The Control Abstraction- The Optimality Principle- Matrix Chain Multiplication-Analysis, All Pairs Shortest Path Algorithm - Floyd-Warshall Algorithm-Analysis. The Control Abstraction of Back Tracking – The N Queen's Problem. Branch and Bound Algorithm for Travelling Salesman Problem.	11

V	Tractable and Intractable Problems, Complexity Classes – P, NP, NP-Hard and NP-Complete Classes- NP Completeness proof of Clique Problem and Vertex Cover Problem- Approximation algorithms- Bin Packing, Graph Coloring. Randomized Algorithms (Definitions of Monte Carlo and Las Vegas algorithms), Randomized version of Quick Sort algorithm with analysis.	12

MARK DISTRIBUTION

Total Marks	CIE	ESE	ESE Duration
150	50	100	3 hours

CONTINOUS INTERNAL EVALUATION PATTERN

Attendance : 10 marks

Continuous Assessment Tests : 25 marks

Continuous Assessment Assignment : 15 marks

CS1U30C	SYSTEM SOFTWARE	Category	L	T	P	Credit	Year of Introduction
		PCC	3	1	0	4	2020

PRE-REQUISITE: CS1U20A Data Structures, CS1U20D Computer Organization & Architecture

COURSE OVERVIEW

The purpose of this course is to create awareness about the low-level codes which are very close to the hardware and about the environment where programs can be developed and executed. This course helps the learner to understand the machine dependent and machine independent system software features and to design/implement system software like assembler, loader, linker, macro processor and device drivers. Study of system software develops the ability to design interfaces between software applications and computer hardware.

COURSE OUTCOMES: After the completion of the course the student will be able to

CO1	Distinguish softwares into system and application software categories.	Understand
CO2	Identify standard and extended architectural features of machines.	Apply
CO3	Identify machine dependent features of system software.	Apply
CO4	Identify machine independent features of system software.	Understand
CO5	Design algorithms for system softwares and analyze the effect of data structures.	Apply
CO6	Explain the features of device drivers and editing & debugging tools	Understand

SYLLABUS

Different types of System Software, Architecture, Instruction set, Assembler Directives and addressing modes of SIC and SIC/XE. Basic functions, Algorithms and Data Structures of Assembler. Machine dependent assembler features, Hand Assembly of SIC/XE Programs, Machine independent Assembler Features, Assembler Design Options, Implementation Example-MASM. Basic Loader Functions, Machine Dependent Loader Features, Machine Independent Loader Features, Loader Design Options. One pass Macro processor Algorithm and data structures, Machine Independent Macro Processor Features, Macro processor design options. Device drivers, Text Editors and Debuggers.

TEXT BOOKS

1. Leland L. Beck, System Software: An Introduction to Systems Programming, 3/E, Pearson Education Asia

REFERENCES

1. D.M. Dhamdhere, Systems Programming and Operating Systems, Second Revised Edition, Tata McGraw Hill.
2. John J. Donovan, Systems Programming, Tata McGraw Hill Edition 1991.
3. George Pajari, Writing UNIX Device Drivers, Addison Wesley Publications (Ebook : http://tocs.ulb.tu-darmstadt.de/197262074.pdf).
4. Peter Abel, IBM PC Assembly Language and Programming, Third Edition, Prentice Hall of India.
5. Jonathan Corbet, Alessandro Rubini, Greg Kroah-Hartman, Linux Device Drivers, Third Edition, O.Reilly Books
6. M. Beck, H. Bohme, M. Dziadzka, et al., Linux Kernel Internals, Second Edition, Addison Wesley Publications,
7. J Nithyashri, System Software, Second Edition, Tata McGraw Hill.
8. The C Preprocessorhttp://gcc.gnu.org/onlinedocs/gcc-2.95.3/cpp_1.html

COURSE PLAN

Module	Contents	No. of hours
I	System Software vs Application Software, Different System Software– Assembler, Linker, Loader, Macro Processor, Text Editor, Debugger, Device Driver, Compiler, Interpreter, Operating System (Basic Concepts only). SIC & SIC/XE Architecture, Addressing modes, SIC & SIC/XE Instruction set, Assembler Directives.	11
II	SIC/XE Programming, Basic Functions of Assembler, Assembler Output Format – Header, Text and End Records. Assembler Data Structures, Two Pass Assembler Algorithm, Hand Assembly of SIC/XE Programs.	12
III	Machine Dependent Assembler Features-Instruction Format and Addressing Modes, Program Relocation. Machine Independent Assembler	12

	Features –Literals, Symbol Defining Statements, Expressions, Program Blocks, Control Sections and Program Linking. Assembler Design Options- One Pass Assembler, Multi Pass Assembler. Implementation Example- MASM Assembler.	
IV	Basic Loader Functions - Design of Absolute Loader, Simple Bootstrap Loader. Machine Dependent Loader Features- Relocation, Program Linking, Algorithm and Data Structures of Two Pass Linking Loader. Machine Independent Loader Features -Automatic Library Search, Loader Options. Loader Design Options.	12
V	Macro Preprocessor - Macro Instruction Definition and Expansion, One pass Macro processor Algorithm and data structures, Machine Independent Macro Processor Features, Macro processor design options. Device drivers - Anatomy of a device driver, Character and block device drivers, General design of device drivers. Text Editors- Overview of Editing, User Interface, Editor structure. Debuggers- Debugging functions and capabilities, Relationship with other parts of the system, Debugging methods- By Induction, Deduction and Backtracking.	13

MARK DISTRIBUTION

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

CONTINOUS INTERNAL EVALUATION PATTERN

Attendance : 10 marks

Continuous Assessment Tests : 25 marks

Continuous Assessment Assignment : 15 marks

Course Code	Course Name	Category	L	T	P	Credit	Revised in 2022
CS1U30K	Operating Systems	PCC	3	1	0	4	

PRE-REQUISITE: CS1U20A Data Structures

COURSE OVERVIEW: Study of operating system is an essential to understand the overall working of computer system, trade-offs between performance and functionality and the division of jobs between hardware and software. This course introduces the concepts of memory management, device management, process management, file management and security & protection mechanisms available in an operating system. The course helps the learner to understand the fundamentals about any operating system design so that they can extend the features of operating system to detect and solve many problems occurring in operating system and to manage the computer resources appropriately.

COURSE OUTCOMES

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

CO 1	Explain the relevance, structure and functions of Operating Systems in computing devices.	Understand
CO 2	Apply the concepts of process management and process scheduling mechanisms employed in Operating Systems.	Apply
CO 3	Explain process synchronization in Operating Systems and illustrate process synchronization mechanisms using Mutex Locks, Semaphores and Monitors.	Understand
CO 4	Explain different methods for detection, prevention, avoidance and recovery for managing deadlocks in Operating Systems.	Understand
CO 5	Develop problem solving skills through memory management techniques.	Apply
CO 6	Explain the security aspects and algorithms for file and storage management in Operating Systems.	Understand

SYLLABUS

Introduction: Functions of an operating system-overview. Kernel Data Structures-Operating System Interfaces and implementation - User Interfaces, System Calls. Operating System implementation and Structure, System Boot process.

Process Management, Threads, Scheduling Queues- Context Switching-Process Creation and Termination-CPU Scheduling -Inter Process Communication.

Process Synchronization, Critical Section-Synchronization – Locks, Semaphores, Monitors, Classical Problems and its implementation- Deadlocks- Handling – Prevention – Avoidance-Detection Recovery.

Memory Management-Swapping - Segmentation – Paging – Page replacement algorithms. Storage Management-RAID- Disk structure, Disk scheduling and implementation

File System Interface- Protection in Operating Systems-Directory implementation – allocation methods-Protection-Access Matrix.

TEXT BOOKS

1. Abraham Silberschatz, Peter B Galvin, Greg Gagne, Operating System Concepts, 9/e, Wiley India, 2015.

REFERENCES

1. Andrew S Tanenbaum, “Modern Operating Systems”, 4th Edition, Prentice Hall, 2015.
2. William Stallings, “Operating systems”, 6th Edition, Pearson, Global Edition, 2015.
3. Garry Nutt, Nabendu Chaki, Sarmistha Neogy, “Operating Systems”, 3rd Edition, Pearson Education,2016.
4. D.M. Dhamdhere, “Operating Systems”, 2nd Edition, Tata McGraw Hill, 2011.
5. Sibsankar Haldar, Alex A Aravind, “Operating Systems”, Pearson Education

COURSE PLAN

Module	Contents	No. of hours
I	Introduction: Functions of an operating system. Single processor, multiprocessor and clustered systems – overview. Kernel Data Structures Operating System Interfaces and implementation - User Interfaces, System Calls – examples. Operating System implementation - approaches. Operating System Structure – Monolithic, Layered, Micro-kernel, Modular. System Boot process.	11

II	Process Management: Process Concept – Processes-States – Process Control Block – Threads. Scheduling – Queues – Schedulers – Context Switching. Process Creation and Termination. CPU Scheduling – Scheduling Criteria – Scheduling Algorithms & implementation (P). Inter Process Communication: Shared Memory, Message Passing, Pipes	12
III	Process Synchronization: Critical Section - Peterson's solution. Synchronization – Locks, Semaphores, Monitors, Classical Problems and its implementation – Producer Consumer, Dining Philosophers and Readers-Writers Problems (P). Deadlocks – Conditions, Modeling using graphs. Handling – Prevention – Avoidance – Banker's Algorithm and its implementation (P) – Detection-Recovery.	13
IV	Memory Management: Main Memory – Swapping - fixed partitions - variable partitions - – Contiguous Memory allocation – Segmentation – Paging – Demand Paging-Page replacement algorithms. Storage Management: Overview of mass storage structure- disks and tapes. Disk structure – accessing disks. Disk scheduling and implementation (P).	12
V	File System Interface: File Concepts – Attributes – operations – types – structure – access methods. Protection. File system implementation. Directory implementation – allocation methods. Free space Management. Protection– Goals, Principles, Domain. Access Matrix.	12
	Total Hours	60

MARK DISTRIBUTION

Total Marks	CIE	ESE	ESE Duration
150	50	100	3 hours

CONTINOUS INTERNAL EVALUATION PATTERN

Attendance	:	10 marks
Continuous Assessment Tests	:	25 marks
Continuous Assessment Assignment	:	15 marks

CS1U30D	MICROPROCESSORS AND MICROCONTROLLERS	Category	L	T	P	Credit	Year of Introduction
		PCC	3	1	0	4	2020

PRE-REQUISITE: CS1U20D Computer organization & Architecture

COURSE OVERVIEW: The course enables the learners capable of understanding the fundamental architecture of microprocessors and micro controllers. This course focuses on the architecture, assembly language programming, interrupts, interfacing of microprocessors with peripheral devices and microcontrollers and its programming. It helps the learners to extend the study of latest processors and develop hardware based solutions.

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

CO1	Illustrate the architecture, modes of operation and addressing modes of microprocessors and microcontrollers	Understand
CO2	Develop 8086 and 8051 assembly language programs.	Apply
CO3	Demonstrate interrupts, its handling and memory interfacing with 8086.	Understand
CO4	Illustrate how different peripherals (8255,8254,8257) are interfaced with microprocessors.	Understand

SYLLABUS

8085 microprocessor- Architecture, 8086 microprocessor- architecture, memory organization, addressing modes, instruction set, assembly language programming.

8086 - Stack structure, interrupt handling, types of interrupts, 8259 Programmable Interrupt Controller interfacing with 8086.

8255 Programmable Peripheral Input/output port Architecture and modes of operation, 8254 and 8257 architecture.

8051 microcontroller architecture, memory organization, interrupts and stack, addressing modes, instruction set, sample programs.

TEXT BOOKS

1. Bhurchandi and Ray, Advanced Microprocessors and Peripherals, Third Edition McGraw Hill.
2. Raj Kamal, Microcontrollers: Architecture, Programming, Interfacing and System Design, Pearson Education.
3. Ramesh Gaonkar, Microprocessor Architecture, Programming, and Applications with the 8085, Penram International Publishing Pvt. Ltd.

REFERENCES

1. Barry B. Brey, The Intel Microprocessors – Architecture, Programming and Interfacing, Eighth Edition, Pearson Education.
2. A. NagoorKani, Microprocessors and Microcontrollers, Second Edition, Tata McGraw Hill
3. Douglas V. Hall, SSSP Rao, Microprocessors and Interfacing, Third Edition, McGrawHill Education.

COURSE PLAN

Module	Contents	No. of hours
I	8085 microprocessor (-Basic Architecture only). 8086 microprocessor – Architecture and signals, Physical Memory organization, Minimum and maximum mode of 8086 system and timings. Comparison of 8086 and 8088.Machine language Instruction format.	10
II	Addressing Modes of 8086. Instruction set – data copy /transfer instructions, arithmetic instructions, logical instructions, string manipulation instructions, branch instructions, unconditional and conditional branch instruction, flag manipulation and processor control instructions. Assembler Directives and operators. Assembly Language Programming with 8086.	13
III	Stack structure of 8086, programming using stack- Interrupts - Types of Interrupts and Interrupt Service Routine- Handling Interrupts in 8086- Interrupt programming. - Programmable Interrupt Controller - 8259, Architecture (Just mention the control word, no need to memorize the control word)- Interfacing Memory with 8086	11
IV	Programmable Peripheral Input/output port 8255 - Architecture and modes of operation- Programmable interval timer 8254-Architecture and modes of operation- DMA controller 8257 Architecture (Just mention the control word, no need to memorize the control word of 8254 and 8257)	13
V	8051 Architecture- Register Organization- Memory and I/O addressing- Interrupts and Stack- 8051 Addressing Modes- Instruction Set- data transfer instructions, arithmetic instructions, logical instructions, Boolean instructions, control transfer instructions- Simple programs	13

	Total Hours	60

MARK DISTRIBUTION

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

CONTINOUS INTERNAL EVALUATION PATTERN

Attendance : 10 marks

Continuous Assessment Tests : 25 marks

Continuous Assessment Assignment : 15 marks

CS1U30E	MANAGEMENT OF SOFTWARE SYSTEMS	Category	L	T	P	Credit	Year of Introduction
		PCC	3	0	0	3	2020

PRE-REQUISITE: CS1U20C Object Oriented Programming using Java

COURSE OVERVIEW: 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.	Understand
CO2	Prepare Software Requirement Specification and Software Design for a given problem.	Apply
CO3	Justify the significance of design patterns and licensing terms in software development, prepare testing, maintenance and DevOps strategies for a project.	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.	Apply
CO5	Explain SQA practices, Process Improvement techniques and Technology advancements in cloud based software models and containers & micro services.	Understand

SYLLABUS

Introduction to Software Engineering-Software process models- Process activities - Agile software development-case studies: An insulin pump control system. Mentcare - a patient information system for mental health care. Functional and non-functional requirements-Requirements engineering processes. Software Requirements Specification- Template - Design Concepts-Architectural Design - What is a component? - Designing Class-Based Components, Conducting Component level design, Component level design for web-apps-Object-oriented design using the UML-Review Techniques - Software testing strategies-Test automation-Overview of DevOps and Code Management-Software Evolution - Evolution processes, Software maintenance. Software Project Management - Risk

management-Project Planning - Project scheduling - Configuration management - Software Quality- Software Process Improvement(SPI) - Cloud-based Software - Software as service.

TEXT BOOKS

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

REFERENCES

1) IEEE Std 830-1998 - IEEE Recommended Practice for Software Requirements Specifications
2) IEEE Std 1016-2009 IEEE Standard for Information Technology—Systems Design— Software Design Descriptions
3) David J. Anderson, Kanban, Blue Hole Press 2010
4) David J. Anderson, Agile Management for Software Engineering, Pearson, 2003
5) Walker Royce, Software Project Management : A unified framework, Pearson Education, 1998
6) Steve. Denning, The age of agile, how smart companies are transforming the way work gets done. New York, Amacom, 2018.
7) Satya Nadella, Hit Refresh: The Quest to Rediscover Microsoft's Soul and Imagine a Better Future for Everyone, Harper Business, 2017
8) HenricoDolfing, Project Failure Case Studies: Lessons learned from other people's mistakes, Kindle edition
9) Mary Poppendieck, Implementing Lean Software Development: From Concept to Cash, Addison-Wesley Signature Series, 2006
10) StarUML documentation - https://docs.staruml.io/
11) OpenProject documentation - https://docs.openproject.org/
12) BugZilla documentation - https://www.bugzilla.org/docs/
13) GitHub documentation - https://guides.github.com/
14) Jira documentation - https://www.atlassian.com/software/jira

COURSE PLAN

Module	Contents	No. of hours
I	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	9

	- 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.	
II	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.	10
III	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.	11
IV	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.	8
V	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,	7

	Everything as a service (IaaS, PaaS), Software as a service. Microservices Architecture -Microservices, Microservices architecture, Microservice deployment.	
	Total Hours	45

MARK DISTRIBUTION

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

CONTINOUS INTERNAL EVALUATION PATTERN

Attendance : 10 marks

Continuous Assessment Tests : 25 marks

Continuous Assessment Assignment : 15 marks

(Each student shall identify a software development problem and prepare Requirements Specification, Design Document, Project Plan and Test case documents for the identified problem as the assignment.)

NC0U30A	DISASTER MANAGEMENT	Category	L	T	P	Credit	Year of Introduction
		MNC	2	0	0	-	2020

PRE-REQUISITE: Nil

COURSE OVERVIEW:

The goal of this course is to expose the students to the fundamental concepts of hazards and disaster management. The course details the various phases of disaster risk management and the measures to reduce disaster risks.

COURSE OUTCOMES

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

CO 1	Explain the fundamental concepts and terminology related to disaster management cycle	Understand
CO 2	Explain hazard and vulnerability types and disaster risk assessment	Understand
CO 3	Describe the process of risk assessment and appropriate methodologies to assess risk	Understand
CO 4	Explain the core elements and phases of disaster risk management and measures to reduce disaster risks across sector and community	Understand
CO 5	Discuss the factors that determine the nature of disaster response and the various disaster response actions	Understand
CO 6	Explain the legislations and best practices for disaster management and risk reduction at national and international level	Understand

SYLLABUS

Introduction- Systems of Earth, Key concepts and terminology in disaster risk reduction and management

Hazard types, Vulnerability types and their assessment, Disaster risk assessment

Disaster risk management- Phases of disaster risk management, Measures for disaster risk reduction- prevention, mitigation, preparedness, Disaster response, Relief

Participatory stakeholder engagement, Disaster communication, Capacity building

Common disaster types in India, Legislations in India on Disaster Management, National Disaster Management Policy, Institutional arrangements for disaster management in India, The Sendai Framework for Disaster risk reduction.

TEXT BOOKS

1. Coppola, D.P., *Introduction to International Disaster Management*, Elsevier Science (B/H), London, 2020
2. Srivastava, H.N., Gupta, G.D., *Management of Natural Disasters in developing countries*, Daya Publishers, Delhi, 2007
3. Subramanian, R., *Disaster Management*, Vikas Publishing House, 2018
4. Sulphey, M.M., Disaster *Management*, PHI Learning, 2016

REFERENCES

1. NDMA, National Policy on Disaster Management, Ministry of Home Affairs, Government of India,2009.
2. National Disaster Management Division, Disaster Management in India - A Status Report, Ministry of Home Affairs, Government of India, New Delhi,2004.
3. National Disaster Management Plan, NDMA, Ministry of Home Affairs, Government of India,2019.
4. Disaster Management Training Manual, UNDP, 2016
5. United Nations Office for Disaster Risk Reduction, Sendai Framework for Disaster Risk Reduction 2015-2030, 2015

COURSE PLAN

Module	Contents	No. of hours
I	Introduction about various systems of earth, Lithosphere- composition, rocks, soils; Atmosphere- layers, ozone layer, greenhouse effect. Weather, cyclones, atmospheric circulations, Indian monsoon; Hydrosphere- oceans, inland water bodies; Biosphere Definition and meaning of key terms in Disaster risk reduction and Management – disaster, hazard, exposure, vulnerability, risk, risk assessment, risk mapping, capacity, resilience, disaster risk reduction, disaster risk management, early warning systems, disaster preparedness, disaster prevention, disaster mitigation, disaster response, damage assessment, crisis counselling, needs assessment	6
II	Various hazard types, hazard mapping; Different types of vulnerability types and their assessment- Physical, social, economic and environmental vulnerability. Core elements of disaster risk assessment Components of a comprehensive disaster preparedness strategy approaches, procedures Different disaster response actions	6
III	Introduction to disaster risk management, core elements of disaster risk management	7

	Phases of disaster risk management, Measures for disaster risk reduction Measures for disaster prevention, mitigation, and preparedness Disaster response- objectives, requirements. Disaster response planning; types of responses Disaster relief, International relief organisations	
IV	Participatory stakeholder engagement, Importance of disaster communication, Disaster communication- methods, barriers, Crisis counselling Introduction to capacity building, Concept- Structural measures, Non-structural measures Introduction to Capacity assessment, Capacity assessment- Strengthening, Capacity for reducing risk	5
V	Introduction- common disaster types in India Common disaster legislations in India on disaster management National disaster management policy, Institutional arrangements for disaster management in India. The Sendai Framework for Disaster risk reduction and targets- priorities for action, guiding principles	6
	Total hours	30

MARK DISTRIBUTION

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

CONTINOUS INTERNAL EVALUATION PATTERN

Attendance : 10 marks

Continuous Assessment Tests : 25 marks

Continuous Assessment Assignment : 15 marks

CS1U38A	SYSTEM SOFTWARE AND MICROPROCESSORS LAB	Category	L	T	P	Credit	Year of Introduction
		PCC	0	0	4	2	2020

PRE-REQUISITE: CS1U20D Computer Organization & Architecture

COURSE OVERVIEW: The aim of this course is to give hands-on experience in how microcontrollers, and microprocessors can be programmed. The course also aims to enable students to design and implement system software. The student should get familiar with assembly level programming of microprocessors and microcontrollers, interfacing of devices to microcontrollers, resource allocation algorithms in operating systems and design and implementation of system software.

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

CO1	Develop 8086 and 8051 programs and execute it using a microprocessor and microcontroller kit respectively.	Apply
CO2	Develop 8086 programs and, debug and execute it using MASM assemblers.	Apply
CO3	Develop and execute programs to interface stepper motor, 8255, 8279 and digital to analog converters with 8086 trainer kit.	Apply
CO4	Implement and execute different paging and file management techniques in OS.	Apply
CO5	Design and implement assemblers, Loaders and macro processors.	Apply

SYLLABUS

MICROPROCESSOR LAB

1. Assembly Language Programming Exercises/Experiments using 8086 Trainer kit
2. Exercises/Experiments using MASM (PC required)
3. Interfacing Exercises/Experiments with 8086 trainer kit through Assembly Language programming
4. Exercises/Experiments using 8051 trainer kit.

SYSTEM SOFTWARE LAB:

1. Experiments related to the operating system.
2. Exercises/Experiments related to the assemblers, loaders and macro processors.

TEXT BOOKS

1) Bhurchandi and Ray, Advanced Microprocessors and Peripherals, Third Edition McGraw Hill.
2) Andrew S Tanenbaum, "Modern Operating Systems", 4th Edition, Prentice Hall, 2015.
3) Leland L. Beck, System Software: An Introduction to Systems Programming, 3/E, Pearson Education Asia, 1997.

REFERENCES

1) A. Nagoor Kani, Microprocessors and Microcontrollers, Second Edition, Tata McGraw Hill
2) Douglas V. Hall, SSSP Rao, Microprocessors and Interfacing, Third Edition, McGrawHill Education.
3) William Stallings, "Operating systems", 6th Edition, Pearson, Global Edition, 2015.
4) Garry Nutt, Nabendu Chaki, Sarmistha Neogy, "Operating Systems", 3rd Edition, Pearson Education.
5) D.M. Dhamdhere, Systems Programming and Operating Systems, Second Revised Edition, Tata McGraw Hill.

MICROPROCESSORS LAB : List of Exercises/ Experiments

(Minimum 10 Exercises (at least 2 questions from each part I, II, III & IV)) : 2 Hrs/week

I. Assembly Language Programming Exercises/Experiments using 8086 Trainer kit

1. Implementation of simple decimal arithmetic and bit manipulation operations.

2. Implementation of code conversion between BCD, Binary, Hexadecimal and ASCII.

3. Implementation of searching and sorting of 16-bit numbers.

II. Exercises/Experiments using MASM (PC Required)

4. Study of Assembler and Debugging commands.

5. Implementation of decimal arithmetic (16 and 32 bit) operations.

6. Implementation of String manipulations.

7. Implementation of searching and sorting of 16-bit numbers.

III. Interfacing Exercises/Experiments with 8086 trainer kit through Assembly Language Programming

8. Interfacing with stepper motor - Rotate through any given sequence.

9. Interfacing with 8255 (mode0 and mode1 only).

10. Interfacing with 8279 (Rolling message, 2 key lockout and N-key rollover implementation).

11. Interfacing with Digital-to-Analog Converter.

IV. Exercises/Experiments using 8051 trainer kit

12. Familiarization of 8051 trainer kit by executing simple Assembly Language programs such as decimal arithmetic and bit manipulation.

13. Implementation of Timer programming (in mode1).

SYSTEM SOFTWARE LAB: List of Exercises/ Experiments

(Minimum 8 Exercises (All the 3 and at least 5 questions from each part V and VI)): 2 Hrs/week

V. Exercises/Experiments from operating system

1. Simulate the following file allocation strategies.

a) Sequential b) Indexed c) Linked

2. Implement the different paging techniques of memory management.

3. Simulate the following file organization techniques

a) Single level directory b) Two level directory c) Hierarchical

VI. Exercises/Experiments from assemblers, loaders and macroprocessor

1. Implement pass one of a two pass assembler.

2. Implement pass two of a two pass assembler.

3. Implement a single pass assembler.

4. Implement a two pass macro processor

5. Implement a single pass macro processor.

6. Implement an absolute loader.

7. Implement a relocating loader

MARK DISTRIBUTION

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	75	75	3 hours

CONTINOUS INTERNAL EVALUATION PATTERN

Attendance	: 15 marks
Continuous Evaluation in Lab	: 30 marks
Continuous Assessment Test	: 15 marks
Viva-voce	: 15 marks

Course Code	Course Name	Category	L	T	P	Credit	Revised in 2022
CS1U38D	Operating Systems Lab	PCC	0	0	4	2	

PRE-REQUISITE: CS1U20A Data Structures

COURSE OVERVIEW: The course aims to offer students a hands-on experience on Operating System concepts using a constructivist approach and problem-oriented learning. Operating systems are the fundamental part of every computing device to run any type of software.

COURSE OUTCOMES

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

CO 1	Make use of systems calls in Linux Operating Systems.	Apply
CO 2	Implement Process Creation and Inter Process Communication in Operating Systems.	Apply
CO 3	Implement First Come First Served, Shortest Job First, Round Robin and Priority based CPU Scheduling Algorithms.	Apply
CO 4	Illustrate the performance of First In First Out, Least Recently Used and Least Frequently Used Page Replacement Algorithms.	Apply
CO 5	Implement modules for Deadlock Detection and Deadlock Avoidance in Operating Systems.	Apply
CO 6	Implement modules for Storage Management and Disk Scheduling in Operating Systems.	Apply

SYLLABUS

Linux basic commands: directory operations, directory structure, redirection, pipes, filters, job control, changing ownership/permissions of files/links/directory - Introduction to Shell Scripting - Inter Process Communication: PIPE, Message Queue, Shared Memory - CPU scheduling algorithms: FCFS, SJF, Round Robin, Priority - producer-consumer problem - dining philosopher's problem - First Readers-Writers Problem - Deadlock avoidance: banker's algorithm - page replacement algorithms: FIFO, LRU, LFU - Disk scheduling algorithms: FCFS, SSTF, SCAN, C-SCAN.

REFERENCES

1) Abraham Silberschatz, Peter B Galvin, Greg Gagne, Operating System Concepts, 9/e, Wiley India, 2015.

2) Andrew S Tanenbaum, "Modern Operating Systems", 4th Edition, Prentice Hall, 2015.

3) William Stallings, "Operating systems", 6th Edition, Pearson, Global Edition, 2015.

COURSE PLAN

SL No.	Topics	No. of hours
1	Getting started with Linux basic commands for directory operations, displaying directory structure in tree format, redirection, pipes, filters, job control, changing ownership/permissions of files/links/directory.	3
2	Introduction to Shell Scripting: Write a shell script to implement a menu driven calculator with following functions 1. Addition 2. Subtraction 3. Multiplication 4. Division 5. Modulus	6
3	Implement programs for Inter Process Communication using PIPE, Message Queue and Shared Memory.	6
4	Simulate the following non-preemptive CPU scheduling algorithms to find turnaround time and waiting time. a) FCFS b) SJF c) Round Robin (pre-emptive) d) Priority.	3
5	Implement the producer-consumer problem using semaphores.	3
6	Write a program to simulate the working of the dining philosopher's problem.	3
7	Implement the First Readers-Writers Problem.	3
8	Implement the banker's algorithm for deadlock avoidance.	3
9	Simulate the following page replacement algorithms a) FIFO b) LRU c) LFU	9
10	Simulate the following disk scheduling algorithms. a) FCFS b) SSTF c) SCAN d) C-SCAN	6
	Total	45

MARK DISTRIBUTION

Total Marks	CIE	ESE	ESE Duration
150	75	75	3 hours

CONTINOUS INTERNAL EVALUATION PATTERN

Attendance	:	15 marks
Continuous Evaluation in Lab	:	30 marks
Continuous Assessment Test	:	15 marks
Viva-voce	:	15 marks

SEMESTER V

MINOR

CS0M30A	CONCEPTS IN SOFTWARE ENGINEERING	Category	L	T	P	Credit	Year of Introduction
		VAC	3	1	0	4	2020

PRE-REQUISITE: CSOM20A Object Oriented Programming

COURSE OVERVIEW: This course provides fundamental knowledge in the Software Development Process. It covers Software Development, Quality Assurance and Project Management concepts. 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	Differentiate Traditional and Agile Software Development approaches	Understand
CO2	Prepare Software Requirement Specification and Software Design for a given problem.	Apply
CO3	Justify the significance of design patterns and licensing terms in software development, prepare testing, maintenance and DevOps strategies for a project.	Apply
CO4	Make use of software project management concepts while planning, estimation, scheduling, tracking and change management of a project, with proper application of SCRUM, Kanban and Lean frameworks.	Apply
CO5	Utilize SQA practices, Process Improvement techniques and Technology improvements namely cloud based software model and containers & microservices in a Software Development Process.	Apply

SYLLABUS

Introduction to Software Engineering - Professional software development -Software engineering ethics. Software process models- Agile software development -Functional and non-functional requirements, Requirements engineering processes. Design concepts - Design within the context of software engineering. Object-oriented design using the UML, Design patterns, Implementation issues, Open-source development -Review Techniques - Cost impact of Software Defects, Code review and statistical analysis. Software testing strategies - Unit Testing, Integration Testing, Validation testing, System testing, Debugging, Software Evolution - Evolution processes, Software

maintenance -Software Project Management - Risk management, Managing people, Teamwork. Project Planning, Configuration management, Version management, System building, Change management, Software Quality, Software Quality Dilemma, Achieving Software Quality Elements of Software Quality Assurance, Software Process Improvement(SPI), SPI Process CMMI process improvement framework, ISO 9001:2000 for Software.

TEXT BOOKS

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

REFERANCE MATERIALS

1. IEEE Std 830-1998 - IEEE Recommended Practice for Software Requirements Specifications
2. IEEE Std 1016-2009 IEEE Standard for Information Technology—Systems Design— Software Design Descriptions
3. David J. Anderson, Kanban, Blue Hole Press 2010
4. David J. Anderson, Agile Management for Software Engineering, Pearson, 2003
5. Walker Royce, Software Project Management : A unified framework, Pearson Education, 1998
6. Steve. Denning, The age of agile, how smart companies are transforming the way work gets done. New York, Amacom, 2018.
7. Satya Nadella, Hit Refresh: The Quest to Rediscover Microsoft's Soul and Imagine a Better Future for Everyone, Harper Business, 2017
8. Henrico Dolfing, Project Failure Case Studies: Lessons learned from other people's mistakes, Kindle edition
9. Mary Poppendieck, Implementing Lean Software Development: From Concept to Cash, Addison-Wesley Signature Series, 2006
10. StarUML documentation - https://docs.staruml.io/
11. OpenProject documentation - https://docs.openproject.org/
12. BugZilla documentation - https://www.bugzilla.org/docs/
13. GitHub documentation - https://guides.github.com/
14. Jira documentation - https://www.atlassian.com/software/jira

COURSE PLAN

Module	Contents	No. of hours
I	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	12

	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.	
II	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.	**12**
III	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, CI/CD/CD. Software Evolution - Evolution processes, Software maintenance.	**14**
IV	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.	**12**
V	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.	**10**

MARK DISTRIBUTION

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

CONTINOUS INTERNAL EVALUATION PATTERN

Attendance : 10 marks

Continuous Assessment Tests : 25 marks

Continuous Assessment Assignment : 15 marks

(Each student shall identify a software development problem and prepare Requirements Specification, Design Document, Project Plan and Test case documents for the identified problem as the assignment.)

CS0M30B	CONCEPTS IN MACHINE LEARNING	Category	L	T	P	Credit	Year of introduction
		VAC	3	1	0	4	2020

PRE-REQUISITE: CS0M20E Mathematics for Machine Learning

COURSE OVERVIEW: This course enables the learners to understand the fundamental concepts and algorithms in machine learning. The course covers the standard and most popular supervised learning algorithms such as linear regression, logistic regression, decision trees, Bayesian learning & the naive Bayes algorithm, support vector machines& kernels, basic clustering algorithms and dimensionality reduction methods. This course helps the students to provide machine learning based solutions to real world problems.

COURSE OUTCOMES

CO1	Illustrate Machine Learning concepts and basic parameter estimation methods.	Apply
CO2	Demonstrate supervised learning concepts (regression, linear classification).	Apply
CO3	Illustrate the concepts of Multilayer neural network and Support Vector Machine.	Apply
CO4	Describe unsupervised learning concepts and dimensionality reduction techniques.	Apply
CO5	Solve real life problems using appropriate machine learning models and evaluate the performance measures	Apply

SYLLABUS

Machine learning paradigms-supervised, semi-supervised, unsupervised, reinforcement learning. Basics of parameter estimation - maximum likelihood estimation(MLE) and maximum a posteriori estimation(MAP). Introduction to Bayesian formulation. Regression - solution using gradient descent algorithm. Linear Methods for Classification- Logistic regression, Perceptron, Naive Bayes, Decision tree algorithm ID3. NN - Multilayer feed forward network, Activation functions (Sigmoid, ReLU, Tanh), Backpropagation algorithm. SVM. Clustering -Hierarchical Clustering, K-means partitional clustering, Expectation maximization (EM) for soft clustering. Dimensionality reduction, Multidimensional scaling,

Linear Discriminant Analysis. Classification Performance measures, Case Study: Develop a classifier for face detection.

TEXT BOOKS

1. Ethem Alpaydin, Introduction to Machine Learning, 2nd edition, MIT Press 2010.
2. Mohammed J Zaki and Wagner Meria, Data Mining and Analysis: Fundamental Concept and Algorithms, Cambridge University Press, first South Asia edition, 2016.
3. Jake VanderPlas, Python Data Science Handbook, O'Reilly Media, 2016
4. Tom Mitchell, Machine Learning, McGraw-Hill, 1997.

REFERENCE MATERIALS

1. Christopher Bishop. Neural Networks for Pattern Recognition, Oxford University Press, 1995.
2. Kevin P. Murphy. Machine Learning: A Probabilistic Perspective, MIT Press 2012.
3. Trevor Hastie, Robert Tibshirani, Jerome Friedman, The Elements Of Statistical Learning, Second edition Springer 2007.
4. P. Langley, Elements of Machine Learning, Morgan Kaufmann, 1995.
5. Richert and Coelho, Building Machine Learning Systems with Python.
6. Davy Cielen, Arno DB Meysman and Mohamed Ali. Introducing Data Science: Big Data, Machine Learning, and More, Using Python Tools, Dreamtech Press 2016.

COURSE PLAN

Module	Contents	No. of hours
I	Machine learning paradigms-supervised, semi-supervised, unsupervised, reinforcement learning. Basics of parameter estimation - maximum likelihood estimation(MLE) and maximum a posteriori estimation(MAP). Introduction to Bayesian formulation.	10
II	Regression - Linear regression with one variable, Linear regression with multiple variables, solution using gradient descent algorithm and matrix method, basic idea of overfitting in regression. Linear Methods for Classification- Logistic regression, Perceptron, Naive Bayes, Decision tree algorithm ID3.	10
III	NN - Multilayer feed forward network, Activation functions (Sigmoi ReLU, Tanh), Backpropagation algorithm.SVM - Introduction, Maximu Margin Classification, Mathematics behind Maximum Marg Classification, Maximum Margin linear separators, soft margin SV	13

	classifier, non-linear SVM, Kernels for learning non-linear function polynomial kernel, Radial Basis Function(RBF).	
IV	Clustering - Similarity measures, Hierarchical Agglomerative Clustering, K-means partitional clustering, Expectation maximization (EM) for soft clustering. Dimensionality reduction – Principal Component Analysis, factor Analysis, Multidimensional scaling, Linear Discriminant Analysis.	**13**
V	Classification Performance measures - Precision, Recall, Accuracy, F-Measure, Receiver Operating Characteristic Curve(ROC), Area Under Curve (AUC). Bootstrapping, Cross Validation, Ensemble methods, Bias-Variance decomposition. Case Study: Develop a classifier for face detection.	**14**

MARK DISTRIBUTION

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

CONTINOUS INTERNAL EVALUATION PATTERN

Attendance : 10 marks

Continuous Assessment Tests : 25 marks

Continuous Assessment Assignment : 15 marks

CS0M30C	CLIENT SERVER SYSTEMS	Category	L	T	P	Credit	Year of Introduction
		VAC	3	1	0	4	2020

PRE-REQUISITE: Nil

COURSE OVERVIEW: The syllabus is prepared with the view of preparing the Engineering Graduates to build effective Client/Server applications. This course aims at providing a foundation in decentralized computer systems, using the client/server model. The course content is decided to cover the essential fundamentals which can be taught within the given slots in the curriculum.

COURSE OUTCOMES: After the completion of the course the student will be able to

CO 1	Identify the basics of client/server systems and the driving force behind the development of client/server systems.	Understand
CO 2	Outline the architecture and classifications of client/server systems.	Understand
CO 3	Summarize the client/server network services for an application.	Understand
CO 4	Identify management services and issues in network	Understand
CO 5	Outline the Client/Server technology in respect of databases and Client/Server database architecture	Understand

SYLLABUS

Introduction-Client/Server Classification-Client/Server Application Components-Client/ Server Systems Services and Support-Client/Server Technology and Databases

TEXT BOOKS

1. Patrick Smith &Steve Guengerich, Client / Server Computing, PHI
2. Subhash Chandra Yadav, Sanjay Kumar Singh, An Introduction to Client/Server Computing, New Age International Publishers.

REFERENCE MATERIALS

1. Jeffrey D.Schank, "N0ovell's Guide to Client-Server Application & Architecture" Novell Press
2. Robert Orfali, Dan Harkey, Jeri Edwards, Client/Server Survival Guide, Wiley-India Edition, Third Edition
3. Dawna Travis Dewire, Client Server Computing- McGraw Hill
4. W.H Inman, Developing Client Server Applications, BPB

COURSE PLAN

Module	Contents	No. of hours
I	Introduction to Client/Server computing - Basic Client/Server Computing Model, Server for Every Client- File Server, Print Server, Application Server, Mail Server, Directory Services Server, Web Server, Database Server, Transaction Servers. Client/Server-Fat or Thin, Stateless or Stateful, Servers and Mainframes, Client/Server Functions. Driving Forces behind Client/Server Computing- Business Perspective, Technology Perspective.	12
II	Client/Server Types-Single Client/Single Server, Multiple Clients/Single Server, Multiple Clients/Multiple Servers, Integration With Distributed Computing, Alternatives To Client/Server Systems. Classification of Client/Server Systems- Two-Tier Computing, Middleware, Three-Tier Computing- Model View Controller (MVC), Principles behind Client/Server Systems. Client/Server Topologies. Existing Client/Server Architecture. Architecture for Business Information System.	11
III	Client- Services, Request for services, RPC, Windows services, Print services, Remote boot services, other remote services, Utility Services. Server- Detailed server functionality, Network operating system, Available platforms, Server operating system. Organizational Expectations, Improving performance of client/server applications, Single system image, Downsizing and Rightsizing, Advantages and disadvantages of Client/Server computing, Applications of Client/Server.	12
IV	Services and Support- System administration, Availability, Reliability, Scalability, Observability, Agility, Serviceability. Software Distribution, Performance, Network management. Remote Systems Management- RDP, Telnet, SSH, Security. LAN and Network Management issues.	13

V	Client/Server Technology and Databases - Storing Data, Database System Architectures. Client/Server In Respect Of Databases- Client/Server Databases, Client/Server Database Computing, Database Computing Vs. Mainframe, PC/File Server Computing. Client/Server Database Architecture - Process-Per-Client Architecture, Multi-Threaded Architecture, Hybrid Architecture. Database Middleware Component - Application Programming Interface, Database Translator, Network Translator.	**12**
		60

MARK DISTRIBUTION

Total Marks	**CIE**	**ESE**	**ESE Duration**
150	50	100	3 hours

CONTINOUS INTERNAL EVALUATION PATTERN

Attendance : 10 marks

Continuous Assessment Test1 (for theory, for 2 hrs) : 20 marks

Continuous Assessment Test2 (for Lab, Internal Examination, for 2 hrs) : 20 marks

B.Tech (HONOURS)

CS1H30A	CRYPTOGRAPHIC ALGORITHMS	Category	L	T	P	Credit	Year of Introduction
		VAC	3	1	0	4	2020

PRE-REQUISITE:CS1H20A-Number Theory

COURSE OVERVIEW:

The course on Cryptographic Algorithms aims at exploring various algorithms deployed in offering confidentiality, integrity, authentication and non-repudiation services. This course covers classical encryption techniques, symmetric and public key crypto-system, key exchange and management, and authentication functions. The concepts covered in this course enable the learners in effective use of cryptographic algorithms for real life applications.

COURSE OUTCOMES: After the completion of the course the student will be able to

CO1	Identify the security services provided for different types of security attacks.	Understand
CO2	Summarize the classical encryption techniques for information hiding.	Apply
CO3	Illustrate symmetric / asymmetric key cryptographic algorithms for secure communication.	Apply
CO4	Interpret key management techniques for secure communication.	Understand
CO5	Summarize message authentication functions in a secure communication scenario.	Understand

SYLLABUS

Need for security, Principles of security, Types of attacks, OSI Security Architecture, Classical encryption techniques- Encrypting communication channels. Symmetric key cryptographic

Algorithms, DES, Block cipher principles, Differential and Linear cryptanalysis, Block cipher modes of operation, IDEA, AES, Stream cipher, RC4. Principles of public key cryptosystems, RSA algorithm, ElGamal cryptographic system, Knapsack algorithm, Diffie-Hellman key exchange algorithm, Elliptical curve cryptosystems. Key Distribution-Public key infrastructure. Authentication requirements, functions, Algorithms and Services.

TEXT BOOKS

1. William Stallings, Cryptography and Network Security Principles and Practice, Pearson Edu, 6e.
2. BruiceSchneier, Applied Cryptography Protocols, Algorithms and source code in C, Wiley,2e.

REFERENCE MATERIALS

1. Behrouz A. Forouzan, Cryptography and Network Security, McGraw Hill, 2e.
2. Johannes A. Buchmann, Introduction to Cryptography, Springer, 2e.
3. DouglasR. Stinson, Cryptography Theory and Practice, 3e, Chapman& Hall/CRC, 2006.
4. Bernard Menezes, Network Security and Cryptography, Cengage Learning, 2011.

COURSE PLAN

Module	Contents	No. of hours
I	Need for security, Security approaches, Principles of security, Types of attacks, OSI Security Architecture, Classical encryption techniques - Substitution techniques, Transposition techniques. Stream cipher, Block cipher, Public key cryptosystems vs. Symmetric key cryptosystems, Encrypting communication channels.	12
II	Overview of symmetric key cryptography, Block cipher principles, Data Encryption Standard (DES), Differential and Linear cryptanalysis, Double DES, Triple DES, International Data Encryption Algorithm (IDEA), Advanced Encryption Algorithm (AES),Block cipher modes of operation, Stream cipher, RC4.	13
III	Principles of public key cryptosystems, RSA algorithm, RSA illustration, Attacks, ElGamal cryptographic system, Knapsack	12

	algorithm, Diffie-Hellman key exchange algorithm, Elliptical curve cryptosystems.	
IV	Symmetric key distribution using symmetric encryption, Symmetric key distribution using asymmetric encryption, Distribution of public keys, Generating keys, transferring keys, Verifying keys, Updating keys, Storing keys, Backup keys, Compromised keys, Public key infrastructure.	**12**
V	Authentication requirements, Authentication functions, Message authentication codes (MAC), Hash functions, Security of Hash functions and MAC, Message Digest 5 (MD5), Secure Hash Algorithm (SHA)-512, Hash-based Message Authentication Code (HMAC), Cipher-based Message Authentication Code (CMAC), X.509 Authentication services.	**11**

MARK DISTRIBUTION

Total Marks	CIE	ESE	ESE Duration
150	50	100	3 hours

CONTINOUS INTERNAL EVALUATION PATTERN

Attendance : 10 marks

Continuous Assessment Tests : 25 marks

Continuous Assessment Assignment : 15 marks

CS1H30B	NEURAL NETWORKS AND DEEP LEARNING	Category	L	T	P	Credit	Year of Introduction
		VAC	3	1	0	4	2020

PRE-REQUISITE: CS1H20B Computational Fundamentals of Machine Learning

COURSE OVERVIEW:

Neural networks is a biologically inspired programming paradigm which enables a computer to learn from observational data and deep learning is a powerful set of techniques for training neural networks. This course introduces the key concepts in neural networks, its architecture and learning paradigms, optimization techniques, basic concepts in deep learning, Convolutional Neural Networks and Recurrent Neural Networks. The students will be able to provide best solutions to real world problems in domains such as computer vision and natural language processing.

COURSE OUTCOMES:After the completion of the course the student will be able to

CO1	Demonstrate the basic concepts of machine learning models and performance measures.	Understand
CO2	Illustrate the basic concepts of neural networks and its practical issues	Apply
CO3	Outline the standard regularization and optimization techniques for deep neural networks	Understand
CO4	Build CNN and RNN models for different use cases.	Apply
CO5	Explain the concepts of modern RNNs like LSTM, GRU	Understand

SYLLABUS

Machine Learning basics - Learning algorithms, Estimators -Bias and Variance. Challenges in machine learning. Simple Linear Regression, Logistic Regression, Performance measures. Single layer perceptrons, Multi Layer Perceptrons (MLPs), Activation functions. Risk minimization, Loss function, backpropagation, Practical issues in neural network training - Computational Challenges. Applications of neural networks. Introduction to deep learning, Deep feed forward network, Training deep models, Optimization techniques - Gradient Descent (GD), GD with momentum, Nesterov accelerated GD, Stochastic GD, AdaGrad, RMSProp, Adam. Regularization Techniques. Convolutional Neural Networks Efficient convolution algorithms. Practical use cases for CNNs, Case study - Building CNN model AlexNet with handwritten digit dataset MNIST. Recurrent neural networks, deep recurrent networks, recursive neural networks, modern RNNs LSTM and GRU, Practical use cases for RNNs. Case study - Natural Language Processing.

TEXT BOOKS

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. Fundamentals of Deep Learning: Designing Next-Generation Machine Intelligence Algorithms (1st. ed.). Nikhil Buduma and Nicholas Locascio. 2017. O'Reilly Media, Inc.

REFERENCE MATERIALS

1. Satish Kumar, Neural Networks: A Classroom Approach, Tata McGraw-Hill Education, 2004.
2. Yegnanarayana, B., Artificial Neural Networks PHI Learning Pvt. Ltd, 2009.
3. Michael Nielsen, Neural Networks and Deep Learning, 2018.

COURSE PLAN

Module	Contents	No. of hours
I	Machine Learning basics - Learning algorithms - Supervised, Unsupervised, Reinforcement, Overfitting, Underfitting, Hyper parameters and Validation sets, Estimators -Bias and Variance. Challenges in machine learning. Simple Linear Regression, Logistic Regression, Performance measures - Confusion matrix, Accuracy, Precision, Recall, Sensitivity, Specificity, Receiver Operating Characteristic curve(ROC), Area Under Curve(AUC).	10
II	Introduction to neural networks -Single layer perceptrons, Multi LayerPerceptrons (MLPs), Representation Power of MLPs, Activation functions - Sigmoid, Tanh, ReLU, Softmax. Risk minimization, Loss function, Training MLPs with backpropagation, Practical issues in neural network training - The Problem of Overfitting, Vanishing and exploding gradient problems, Difficulties in convergence, Local and spurious Optima, Computational Challenges. Applications of neural networks.	10
III	Introduction to deep learning, Deep feed forward network, Training deep models, Optimization techniques - Gradient Descent (GD), GD with momentum, Nesterov accelerated GD, Stochastic GD, AdaGrad, RMSProp, Adam. Regularization Techniques - L1 and L2 regularization, Early stopping, Dataset augmentation, Parameter sharing and tying, Injecting noise at input, Ensemble methods, Dropout, Parameter initialization.	14
IV	Convolutional Neural Networks – Convolution operation, Motivation, Pooling, Convolution and Pooling as an infinitely strong prior, Variants of convolution functions, Structured outputs, Data types, Efficient convolution algorithms. Practical use cases for CNNs, Case study - Building CNN model AlexNet with handwritten digit dataset MNIST.	13
V	Recurrent neural networks – Computational graphs, RNN design, encoder – decoder sequence to sequence architectures, deep recurrent networks, recursive neural networks, modern RNNs LSTM and GRU, Practical use cases for RNNs. Case study - Natural Language Processing.	13

MARK DISTRIBUTION

Total Marks	CIE	ESE	ESE Duration
150	50	100	3 hours

CONTINOUS INTERNAL EVALUATION PATTERN

Attendance	: 10 marks
Continuous Assessment Tests	: 25 marks
Continuous Assessment Assignment	: 15 marks

CS1H30C	PRINCIPLES OF MODEL CHECKING	Category	L	T	P	Credit	Year of Introduction
		VAC	3	1	0	4	2020

PRE-REQUISITE: Nil

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

CO1	Illustrate an application for model checking.	Understand
CO2	Describe finite-state modelling of hardware and software.	Understand
CO3	Identify the linear-time properties required to represent the requirements of a system.	Apply
CO4	Specify a given linear-time property in Linear Temporal Logic (LTL).	Apply
CO5	Perform LTL model checking with the tool SAL (Symbolic Analysis Laboratory).	Apply

SYLLABUS

System Verification – Hardware and Software Verification - Model Checking, Characteristics - Transition Systems – Direct Predecessors and Successors, Terminal State, Deterministic Transition System - Execution Fragment: Maximal and Initial - - Reachable States - Modeling Hardware and Software Systems - Linear-Time (LT) Properties - Deadlock, Behavior - Paths and State Graph, Path Fragment, Maximal and Initial Path Fragment, Traces - LT Properties, Satisfaction Relation, Trace Equivalence. Safety Properties and Invariants - Liveness Properties - Safety vs. Liveness Properties. Fairness: Unconditional, Weak and Strong Fairness, Strategies, Fairness and Safety. Regular Safety property: Verifying. Automata on Infinite Words - ω -Regular Languages and Properties, Nondeterministic and Deterministic Buchi Automata Generalised Buchi Automata, Nested

Depth-First Search. Linear Temporal Logic (LTL) - Syntax, Semantics, Equivalence, Weak Until, Release and Positive Normal Form, . Automata Based LTL Model Checking. Introduction to the tool Symbolic Analysis Laboratory (SAL). The Language - The expression language, The transition Language, The module language, SAL Contexts. SAL Examples - Mutual Exclusion.

TEXT BOOKS

1. Christel Baier and Joost-Pieter Katoen, Principles of Model Checking, The MIT Press. (Modules 1 - 4)
2. Leonardo de Moura, Sam Owre and N. Shankar, The SAL Language Manual, SRI International (http://sal.csl.sri.com/doc/language-report.pdf, Chapters 1, 3, 4, 5, 6, 7) (Module 5).

REFERENCE MATERIALS

1. SAL Examples (http://sal.csl.sri.com/examples.shtml) (Module 5)

COURSE PLAN

Module	Contents	No. of hours
I	System Verification– Hardware and Software Verification, Model Checking, Characteristics of Model Checking. Transition Systems – Transition System, Direct Predecessors and Successors, Terminal State, Deterministic Transition System. Executions - Execution Fragment, Maximal and Initial Execution Fragment, Execution, Reachable States. Modeling Hardware and Software Systems- Sequential Hardware Circuits, Data Dependent Systems.	**12**
II	Linear-Time (LT) Properties - Deadlock. Linear-Time Behavior - Paths and State Graph, Path Fragment, Maximal and Initial Path Fragment, Path. Traces - Trace and Trace Fragment, LT Properties - LT Property, Satisfaction Relation for LT Properties, Trace Equivalence and LT Properties. Safety Properties and Invariants - Invariants, Safety Properties, Trace Equivalence and Safety properties. Liveness Properties - Liveness Property, Safety vs. Liveness Properties. Fairness - Fairness, Unconditional, Weak and Strong Fairness, Fairness Strategies, Fairness	**12**

	and Safety. (Definition and examples only for all topics - no proof required).	
III	Regular Properties - Model Checking Regular Safety properties - Regular Safety property, Verifying Regular Safety Properties. Automata on Infinite Words - ω -Regular Languages and Properties, Nondeterministic Buchi Automata (NBA), Deterministic Buchi Automata (DBA), Generalised Buchi Automata (Definitions only). Model Checking ω -Regular Properties - Persistence Properties and Product, Nested Depth-First Search (Only algorithms required).	**12**
IV	Linear Temporal Logic (LTL) - Syntax, Semantics, Equivalence of LTL Formulae, Weak Until, Release and Positive Normal Form, Fairness, Safety and Liveness in LTL (Definitions only). Automata Based LTL Model Checking (Algorithms and examples only).	**12**
V	Introduction - Introduction to the tool Symbolic Analysis Laboratory (SAL). The Language of SAL - The expression language, The transition Language, The module language, SAL Contexts. SAL Examples - Mutual Exclusion, Peterson's Protocol, Synchronous Bus Arbiter, Bounded Bakery protocol, Bakery Protocol, Simpson's Protocol, Stack.	**12**

MARK DISTRIBUTION

Total Marks	**CIE**	**ESE**	**ESE Duration**
150	50	100	3 hours

CONTINOUS INTERNAL EVALUATION PATTERN

Attendance	: 10 marks
Continuous Assessment Tests	: 25 marks
Continuous Assessment Assignment	: 15 marks

SEMESTER VI

CS1U30F	COMPILER DESIGN	Category	L	T	P	Credit	Year of Introduction
		PCC	3	1	0	4	2020

PRE-REQUISITE:CS1U20A Data Structures, CS1U20G Formal Languages & Automata Theory

COURSE OVERVIEW: The purpose of this course is to create awareness among students about the phases of a compiler and the techniques for designing a compiler. This course covers the fundamental concepts of different phases of compilation such as lexical analysis, syntax analysis, semantic analysis, intermediate code generation, code optimization and code generation. Students can apply this knowledge in design and development of compilers.

COURSE OUTCOMES: After the completion of the course the student will be able to

CO1	Explain the phases in compilation process(lexical analysis, syntax analysis, semantic analysis, intermediate code generation, code optimization and code generation).	Understand
CO2	Model a lexical analyser based on the given regular definition.	Apply
CO3	Model language syntax using Context Free Grammar and develop parse tree representation using leftmost and rightmost derivations.	Apply
CO4	Build different types of parsers(Bottom-up and Top-down) and for a given grammar.	Apply
CO5	Build Syntax Directed Translation for a context free grammar, compare various storage allocation strategies and classify intermediate representations	Apply
CO6	Apply code optimization and code generation techniques in compilation.	Apply

SYLLABUS

Analysis of the source program, Compiler writing tools. Bootstrapping, Lexical Analysis, Role of the Syntax Analyser, Top-Down Parsing - Recursive Descent parsing, Predictive Parsing, LL(1) Grammars, Bottom-Up Parsing-Shift Reduce parsing, Operator precedence parsing, LR Parsing, Syntax directed translation, Run-Time Environments, Intermediate Code Generation, Code Optimization, Code Generation.

TEXT BOOKS

1) Aho A.V., Ravi Sethi and D. Ullman. Compilers – Principles Techniques and Tools, Addison Wesley, 2006.

REFERENCES

1) D.M.Dhamdhere, System Programming and Operating Systems, Tata McGraw Hill & Company, 1996.
2) Kenneth C. Louden, Compiler Construction – Principles and Practice, Cengage Learning Indian Edition, 2006.
3) Tremblay and Sorenson, The Theory and Practice of Compiler Writing, Tata McGraw Hill & Company,1984.

COURSE PLAN

Module	Contents	No. of hours
I	Analysis of the source program - Analysis and synthesis phases, Phases of a compiler. Compiler writing tools. Bootstrapping. Lexical Analysis - Role of Lexical Analyser, Input Buffering, Specification of Tokens, Recognition of Tokens.	12
II	Role of the Syntax Analyser – Syntax error handling. Review of Context Free Grammars - Derivation and Parse Trees, Eliminating Ambiguity. Basic parsing approaches - Eliminating left recursion, left factoring. Top-Down Parsing - Recursive Descent parsing, Predictive Parsing, LL(1) Grammars.	14

III	Handle Pruning. Shift Reduce parsing. Operator precedence parsing (Concept only). LR parsing - Constructing SLR, LALR and canonical LR parsing tables.	10
IV	Syntax directed translation - Syntax directed definitions, S-attributed definitions, L-attributed definitions, Bottom-up evaluation of S-attributed definitions. Run-Time Environments - Source Language issues, Storage organization, Storage-allocation strategies. Intermediate Code Generation - Intermediate languages, Graphical representations, Three-Address code, Quadruples, Triples.	14
V	Code Optimization - Principal sources of optimization, Machine dependent and machine independent optimizations, Local and global optimizations. Code generation - Issues in the design of a code generator, Target Language, A simple code generator.	10
	Total Hours	60

MARK DISTRIBUTION

Total Marks	CIE	ESE	ESE Duration
150	75	75	3 hours

CONTINOUS INTERNAL EVALUATION PATTERN

Attendance : 10 marks

Continuous Assessment Tests : 25 marks

Continuous Assessment Assignment : 15 marks

CS1U30G	COMPUTER GRAPHICS AND IMAGE PROCESSING	Category	L	T	P	Credit	Year of Introduction
		PCC	3	1	0	4	2020

PRE-REQUISITE: Nil

COURSE OVERVIEW: The purpose of this course is to make awareness about strong theoretical relationships between computer graphics and image processing. This course helps the learner to understand three- dimensional environment representation in a computer, transformation of 2D/3D objects, basic mathematical techniques and algorithms used to build useful applications, imaging, and image processing techniques. The study of computer graphics and image processing develops the ability to create image processing frameworks for different domains and develops algorithms for emerging display technologies.

COURSE OUTCOMES: After the completion of the course the student will be able to

CO1	Describe the working principles of graphics devices.	Understand
CO2	Apply line drawing, circle drawing and polygon filling algorithms.	Apply
CO3	Apply geometric transformations on 2D & 3D objects, clipping algorithms and projection algorithms.	Apply
CO4	Summarize visible surface detection methods.	Understand
CO5	Summarize the concepts of digital image representation, processing and demonstrate pixel relationships.	Understand
CO6	Solve image enhancement and segmentation problems using spatial domain techniques.	Apply

SYLLABUS

Basics of Computer Graphics and its applications. Video Display devices. Line drawing algorithms. Circle drawing algorithms. Filled Area Primitives. Two dimensional transformations, Matrix representations and homogeneous coordinates. Basic 3D transformations. Window to viewport transformation. Cohen Sutherland Line clipping algorithm. Sutherland Hodgeman

Polygon clipping algorithm. Three dimensional viewing pipeline. Projections- Parallel and Perspective projections. Visible surface detection algorithms- Depth buffer algorithm, Scan line algorithm. Introduction to Image processing and applications. Fundamental steps in image processing. Components of image processing system. Coordinate conventions. Sampling and quantization. Spatial and Gray Level Resolution. Basic relationship between pixels– neighbourhood, adjacency, connectivity. Fundamentals of spatial domain-convolution operation. Basic gray level transformation functions. Histogram equalization. Basics of spatial filtering - Sharpening spatial filters. Fundamentals of Image Segmentation. Thresholding. Region based Approach - Region Growing, Region Splitting and Merging. Edge Detection - Edge Operators- Sobel and Prewitt.

TEXT BOOKS

1) Donald Hearn and M. Pauline Baker, Computer Graphics, PHI, 2e, 1996
2) Rafael C. Gonzalez and Richard E. Woods, Digital Image Processing. Pearson, 4e, 2017

REFERENCES

1) William M. Newman and Robert F. Sproull, Principles of Interactive Computer Graphics. McGraw Hill, 2001.
2) Zhigang Xiang and Roy Plastock, Computer Graphics (Schaum's outline Series), McGraw Hill 2019.
3) David F. Rogers , Procedural Elements for Computer Graphics, Tata McGraw Hill,2001.
4) M. Sonka, V. Hlavac, and R. Boyle, Image Processing, Analysis, and Machine Vision, Thomson India Edition, 4e, 2017.

COURSE PLAN

Module	Contents	No. of hours
I	Basics of Computer Graphics and its applications. Video Display devices- Refresh Cathode Ray Tubes, Random Scan Displays and systems, Raster scan displays and systems. Line drawing algorithms- DDA, Bresenham's algorithm. Circle drawing algorithms- Midpoint Circle generation algorithm, Bresenham's algorithm.	10
II	Filled Area Primitives- Scan line polygon filling, Boundary filling and flood filling. Two dimensional transformations-Translation, Rotation, Scaling, Reflection and Shearing, Composite transformations, Matrix representations and homogeneous coordinates. Basic 3D transformations.	10

III	Window to viewport transformation. Cohen Sutherland Line clipping algorithm. Sutherland Hodgeman Polygon clipping algorithm. Three dimensional viewing pipeline. Projections- Parallel and Perspective projections. Visible surface detection algorithms- Depth buffer algorithm, Scan line algorithm.	13
IV	Introduction to Image processing and applications. Image as 2D data. Image representation in Gray scale, Binary and Colour images. Fundamental steps in image processing. Components of image processing system. Coordinate conventions. Sampling and quantization. Spatial and Gray Level Resolution. Basic relationship between pixels– neighbourhood, adjacency, connectivity. Fundamentals of spatial domain-convolution operation.	13
V	Basic gray level transformation functions - Log transformations, Power-Law transformations, Contrast stretching. Histogram equalization. Basics of spatial filtering - Smoothing spatial filter- Linear and nonlinear filters, and Sharpening spatial filters-Gradient and Laplacian. Fundamentals of Image Segmentation. Thresholding - Basics of Intensity thresholding and Global Thresholding. Region based Approach - Region Growing, Region Splitting and Merging. Edge Detection - Edge Operators- Sobel and Prewitt.	14
	Total Hours	60

MARK DISTRIBUTION

Total Marks	CIE	ESE	ESE Duration
150	50	100	3 hours

CONTINOUS INTERNAL EVALUATION PATTERN

Attendance : 10 marks

Continuous Assessment Tests : 25 marks

Continuous Assessment Assignment : 15 marks

CS1U30B	COMPUTER NETWORKS	Category	L	T	P	Credit	Year of Introduction
		PCC	3	1	0	4	2020

PRE-REQUISITE : Nil

COURSE OVERVIEW: Study of this course provides the learners a clear understanding of how computer networks from local area networks to the massive and global Internet are built, how they allow computers to share information and communicate with one another. This course covers the physical aspects of computer networks, layers of OSI Reference model, and inter-networking. The course helps the learners to compare and analyse the existing network technologies and choose a suitable network design for a given system.

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

CO1	Explain the features of computer networks, protocols, and network design models	Understand
CO2	Describe the fundamental characteristics of the physical layer and identify the usage in network communication	Understand
CO3	Explain the design issues of data link layer, link layer protocols, bridges and switches	Understand
CO4	Illustrate wired LAN protocols (IEEE 802.3) and wireless LAN protocols (IEEE 802.11)	Understand
CO5	Select appropriate routing algorithms, congestion control techniques, and Quality of Service requirements for a network	Apply
CO6	Illustrate the functions and protocols of the network layer, transport layer, and application layer in inter-networking	Understand

SYLLABUS

Introduction-Reference models-Physical Layer-Data link layer - Medium Access Control (MAC) sublayer - Wireless LANs - 802.11 -Network layer - Routing Algorithms- Congestion Control Algorithms- Quality of Service (QoS)- Network Layer in Internet--Transport Layer – User Datagram Protocol (UDP)- Transmission Control Protocol (TCP) – Application Layer protocols.

TEXT BOOKS

1. Andrew S. Tanenbaum, Computer Networks, 4/e, PHI (Prentice Hall India).
2. Behrouz A Forouzan, Data Communication and Networking, 4/e, Tata McGraw Hill.

REFERENCES

1. Larry L Peterson and Bruce S Dave, Computer Networks – A Systems Approach, 5/e, Morgan Kaufmann.
2. Fred Halsall, Computer Networking and the Internet, 5/e.
3. James F. Kurose, Keith W. Ross, Computer Networking: A Top-Down Approach, 6/e.
4. Keshav, An Engineering Approach to Computer Networks, Addison Wesley, 1998.
5. W. Richard Stevens. TCP/IP Illustrated Volume 1, Addison-Wesley, 2005.
6. William Stallings, Computer Networking with Internet Protocols, Prentice-Hall, 2004.
7. Request for Comments (RFC) Pages - IETF -https://www.ietf.org/rfc.html.

COURSE PLAN

Module	Contents	No. of hours
I	Introduction – Uses of computer networks, Network hardware, Network software. Reference models – The OSI reference model, The TCP/IP reference model, Comparison of OSI and TCP/IP reference models. Physical Layer – Modes of communication, Physical topologies, Signal encoding, Repeaters and hub, Transmission media overview. Performance indicators – Bandwidth, Throughput, Latency, Queuing time, Bandwidth–Delay product.	12
II	Data link layer - Data link layer design issues, Error detection and correction, Sliding window protocols, High-Level Data Link Control(HDLC)protocol. Medium Access Control (MAC) sublayer –Channel allocation problem, Multiple access protocols, Ethernet, Wireless LANs - 802.11, Bridges & switches - Bridges	13

	from 802.x to 802.y, Repeaters, Hubs, Bridges, Switches, Routers and Gateways.	
III	Network layer design issues. Routing algorithms - The Optimality Principle, Shortest path routing, Flooding, Distance Vector Routing, Link State Routing, Multicast routing, Routing for mobile hosts. Congestion control algorithms. Quality of Service (QoS) - requirements, Techniques for achieving good QoS.	12
IV	IP protocol, IP addresses, Internet Control Message Protocol (ICMP), Address Resolution Protocol (ARP), Reverse Address Resolution Protocol (RARP), Bootstrap Protocol (BOOTP), Dynamic Host Configuration Protocol (DHCP). Open Shortest Path First(OSPF) Protocol, Border Gateway Protocol (BGP), Internet multicasting, IPv6, ICMPv6.	11
V	Transport service – Services provided to the upper layers, Transport service primitives. User Datagram Protocol (UDP). Transmission Control Protocol (TCP) – Overview of TCP, TCP segment header, Connection establishment &release, Connection management modeling, TCP retransmission policy, TCP congestion control. Application Layer –File Transfer Protocol (FTP), Domain Name System (DNS), Electronic mail, Multipurpose Internet Mail Extension (MIME), Simple Network Management Protocol (SNMP), World Wide Web(WWW) – Architectural overview.	12
	Total Hours	60

MARK DISTRIBUTION

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

CONTINOUS INTERNAL EVALUATION PATTERN

Attendance	: 10 marks
Continuous Assessment Tests	: 25 marks
Continuous Assessment Assignment	: 15 marks

HS0U30A	INDUSTRIAL ECONOMICS & FOREIGN TRADE	Category	L	T	P	Credit	Year of Introduction
		HSC	3	0	0	3	2020

PRE REQUISITE : NIL

COURSE OVERVIEW:

The course enables students to make better economic decisions in wage employment and entrepreneurship using economic alternatives and investment alternatives.

COURSE OUTCOMES

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

CO 1	Explain the problem of scarcity of resources, consumer behaviour and the equilibrium condition of demand and supply.	Understand
CO 2	Demonstrate the production function and equilibrium condition of a producer.	Understand
CO 3	Survey the functional requirement of a firm under various competitive conditions.	Analyse
CO 4	Infer the overall performance of the economy, the regulation of economic fluctuations and its impact on various sections in the society.	Analyse
CO 5	Compare the profitability of projects and businesses with the help of capital budgeting methods.	Evaluate
CO 6	Determine the current impact of global economic policies on the business opportunities of a firm.	Apply

SYLLABUS

Scarcity and choice - Basic economic problems- PPC – Firms and its objectives – types of firms – Utility – Law of diminishing marginal utility – Demand and its determinants – law of demand – elasticity of demand – measurement of elasticity and its applications – Supply, law of supply and determinants of supply – Equilibrium – Changes in demand and supply and its effects – Consumer surplus and producer surplus (Concepts) – Taxation and deadweight loss.

Production function – law of variable proportion – economies of scale – internal and external economies – Isoquants, isocost line and producer's equilibrium – Expansion path – Technical progress and its implications – Cobb-Douglas production function - Cost concepts – Social cost: private cost and external cost – Explicit and implicit cost – sunk cost - Short run cost curves - long run cost curves – Revenue (concepts) – Shutdown point – Break-even point.

Perfect and imperfect competition – monopoly, regulation of monopoly, monopolistic completion (features and equilibrium of a firm) – oligopoly – Kinked demand curve – Collusive oligopoly (meaning) – Non-price competition – Product pricing – Cost plus pricing – Target return pricing – Penetration pricing – Predatory pricing – Going rate pricing – Price skimming. Principles of taxation - Direct Tax – Indirect Tax – GST. Concepts of demonetization. Cryptocurrency

Circular flow of economic activities – Stock and flow Gross. National Income – Concepts - Methods of measuring national income – Inflation- causes and effects – Measures to control inflation. Monetary and fiscal policies – Business financing- Bonds and shares -Money market and Capital market – Stock market – Demat account and Trading account - SENSEX and NIFTY. Capital Budgeting - Methods of Investment analysis - Pay back, ARR, NPV, IRR and B/C ratio

Advantages and disadvantages of international trade - Absolute and Comparative advantage theory - Heckscher - Ohlin theory - Balance of payments – Components – Balance of Payments deficit and devaluation – Trade policy – Free trade versus protection – Tariff and non-tariff barriers

REFERENCES

1. Gregory N Mankiw, 'Principles of Micro Economics', Cengage Publications 2015
2. Gregory N Mankiw, 'Principles of Macro Economics', Cengage Publications 2012
3. Dwivedi D.N., 'Macro Economics', Tata McGraw Hill, New Delhi
4. Mithani D M, 'Managerial Economics', Himalaya Publishing House, Mumbai
5. Tulsian, 'Financial Management' S Chand & Company 2017
6. Francis Cherunilam, 'International Economics', McGraw Hill, New Delhi

COURSE PLAN

Module	Contents	No. of hours
I	Scarcity and choice - Basic economic problems- PPC – Firms and its objectives – types of firms – Utility – Law of diminishing marginal utility – Demand and its determinants – law of demand – elasticity of demand – measurement of elasticity and its applications – Supply, law of supply and determinants of supply – Equilibrium – Changes in demand and supply and its effects – Consumer surplus and producer surplus (Concepts) – Taxation and deadweight loss.	8
II	Production function – law of variable proportion – economies of scale – internal and external economies – Isoquants, isocost line and producer's equilibrium – Expansion path – Technical progress and its implications – Cobb-Douglas production function - Cost concepts – Social cost: private cost and external cost – Explicit and implicit cost – sunk cost - Short run cost curves - long run cost curves – Revenue (concepts) – Shutdown point – Break-even point.	8
III	Perfect and imperfect competition – monopoly, regulation of monopoly, monopolistic completion (features and equilibrium of a firm) – oligopoly – Kinked demand curve – Collusive oligopoly (meaning) – Non-price competition – Product pricing – Cost plus pricing – Target return pricing – Penetration pricing – Predatory pricing – Going rate pricing – Price skimming. Principles of taxation - Direct Tax – Indirect Tax – GST. Concepts of demonetization. Cryptocurrency	9
IV	Circular flow of economic activities – Stock and flow Gross. National Income – Concepts - Methods of measuring national income – Inflation- causes and effects – Measures to control inflation. Monetary and fiscal policies – Business financing- Bonds and shares -Money market and Capital market – Stock market – Demat account and Trading account - SENSEX and NIFTY. Capital Budgeting - Methods of Investment analysis - Pay back, ARR, NPV, IRR and B/C ratio	11
V	Advantages and disadvantages of international trade - Absolute and Comparative advantage theory - Heckscher - Ohlin theory - Balance of payments – Components – Balance of Payments deficit and devaluation – Trade policy – Free trade versus protection – Tariff and non-tariff barriers	9

	Total hours	45

MARK DISTRIBUTION

Total Marks	CIE	ESE	ESE Duration
150	50	100	3 hours

CONTINOUS INTERNAL EVALUATION PATTERN

Attendance	: 10 marks
Continuous Assessment Tests	: 25 marks
Continuous Assessment Assignment	: 15 marks

CS1U30I	COMPREHENSIVE COURSE WORK	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PCC	1	0	0	1	2020

Preamble: The objective of this Course work is to ensure the comprehensive knowledge of each student in the most fundamental core courses in the curriculum. Five core courses credited from semesters 3, 4 and 5 are chosen for the detailed study in this course work. This course helps the learner to become competent in cracking GATE, placement tests and other competitive examinations

Prerequisite:

1. Data Structures (CS1U20A)
2. Operating Systems (CS1U30K)
3. Computer Organization And Architecture (CS1U20D)
4. Database Management Systems (CS1U20E)
5. Formal Languages And Automata Theory (CS1U20G)

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

CO1	Comprehend the concepts and applications of data structures.	Understand
CO2	Comprehend the concepts, functions and algorithms in Operating System.	Understand
CO3	Comprehend the organization and architecture of computer systems.	Understand
CO4	Comprehend the fundamental principles of database design and manipulation.	Understand
CO5	Comprehend the concepts in formal languages and automata theory.	Understand

Assessment Pattern

Bloom's Category	End Semester Examination
Remember	**10**
Understand	**20**
Apply	**20**
Analyse	
Evaluate	
Create	

Mark distribution

Total Marks	CIE	ESE	ESE Duration
50	**0**	**50**	**1 hour**

End Semester Examination Pattern: Objective Questions with multiple choice, a maximum of four options. Question paper includes fifty questions of one mark each, distributed equally from all the five identified courses.

Syllabus

Full Syllabus of all five selected Courses.

1. Data Structures (CS1U20A)
2. Operating Systems (CS1U30K)
3. Computer Organization and Architecture (CS1U20D)
4. Database Management Systems (CS1U20E)
5. Formal Languages and Automata Theory (CS1U20G)

Course Contents and Lecture Schedule

No	Topic	No. of Lectures
1	**DATA STRUCTURES**	
1.1	Mock Test on Module 1, Module 2 and Module 3	1 hour
1.2	Mock Test on Module 4 and Module 5	1 hour
1.3	Feedback and Remedial class	
2	**OPERATING SYSTEMS**	
2.1	Mock Test on Module 1 and Module 2	1 hour
2.2	Mock Test on Module 3, Module 4 and Module 5	1 hour
2.3	Feedback and Remedial class	1 hour
3	**COMPUTER ORGANIZATION AND ARCHITECTURE**	
3.1	Mock Test on Module 1, Module 2 and Module 3	1 hour
3.2	Mock Test on Module 4 and Module 5	1 hour
4	**DATABASE MANAGEMENT SYSTEMS**	
4.1	Mock Test on Module 1, Module 2 and Module 3	1 hour
4.2	Mock Test on Module 4 and Module 5	1 hour
4.3	Feedback and Remedial class	
5	**FORMAL LANGUAGES AND AUTOMATA THEORY**	
5.1	Mock Test on Module 1, Module 2 and Module 3	1 hour
5.2	Mock Test on Module 4 and Module 5	1 hour
5.3	Feedback and Remedial class	1 hour

CS1U38C	NETWORKING LAB	CATEGORY	L	T	P	Credit	Year of Introduction
		PCC	0	0	3	2	2020

PRE-REQUISITE: ES0U10F-Introduction to Computer Programming, CS1U20A- Data Structures

COURSE OVERVIEW: The course enables the learners to get hands-on experience in network programming using Linux System calls and network monitoring tools. It covers implementation of network protocols and algorithms, configuration of network services and familiarization of network simulators. This helps the learners to develop, implement protocols and evaluate its performance for real world networks.

COURSE OUTCOMES: After the completion of the course the student will be able to

CO1	Make use of network related commands and configuration files in Linux Operating System.	Apply
CO2	Develop network application programs and protocols.	Apply
CO3	Make use of network monitoring tools to analyze network traffic.	Apply
CO4	Design and setup a network and configure different network protocols.	Apply
CO5	Develop simulation of fundamental network concepts using a network simulator.	Apply

SYLLABUS

*****Mandatory

(Note: At least one program from each topic in the syllabus should be completed in the Lab)

1) Getting started with the basics of network configuration files and networking commands in Linux.*
2) To familiarize and understand the use and functioning of system calls used for network programming in Linux.*
3) Implement client-server communication using socket programming and TCP as transport layer protocol.*
4) Implement client-server communication using socket programming and UDP as transport layer protocol.*
5) Simulate sliding window flow control protocols.* (Stop and Wait, Go back N, Selective Repeat ARQ protocols)
6) Implement and simulate algorithm for Distance Vector Routing protocol or Link State Routing protocol.*
7) Implement Simple Mail Transfer Protocol.
8) Implement File Transfer Protocol.*
9) Implement congestion control using a leaky bucket algorithm.*
10) Understanding the Wireshark tool.*
11) Design and configure a network with multiple subnets with wired and wireless LANs using required network devices. Configure commonly used services in the network.*
12) Study of NS2 simulator.*

REFERENCES

1) W. Richard Stevens, Bill Fenner, Andy Rudoff, UNIX Network Programming: Volume 1, The Sockets Networking API, 3rd Edition, Pearson, 2015
2) Lisa Bock, Learn Wireshark: Confidently navigate the Wireshark interface and solve real-world networking problems, Packt Publishing, 2019
3) TeerawatIssariyakul, Ekram Hossain, Introduction to Network Simulator NS2,2nd Edition, Springer,2019

MARK DISTRIBUTION

Total Marks	CIE	ESE	ESE Duration
150	75	75	3 hours

CONTINOUS INTERNAL EVALUATION PATTERN

Attendance	: 15 marks
Continuous Evaluation in Lab	: 30 marks
Continuous Assessment Test	: 15 marks
Viva-voce	: 15 marks

CS1U39A	MINI PROJECT	Category	L	T	P	Credit	Year of Introduction
		PCC	0	0	3	2	2020

PRE-REQUISITE: A sound knowledge in any programming language and fundamental concepts of Software Engineering.

COURSE OVERVIEW:

The objective of this course is to apply the fundamental concepts of Software Engineering principles for the effective development of an application/research project. This course helps the learners to practice the different steps to be followed in the software development process such as literature review and problem identification, preparation of Software Requirement Specification &Software Design Document (SDD), testing, development and deployment. Mini project enables the students to boost their skills, widen the horizon of thinking and their ability to resolve real life problems.

COURSE OUTCOMES: After the completion of the course the student will be able to

CO1	Identify technically and economically feasible problems.	Apply
CO2	Identify and survey the relevant literature for getting exposed to related solutions and get familiarized with software development processes	Apply
CO3	Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions of minimal complexity by using modern tools & advanced programming techniques.	Apply
CO4	Prepare technical report and deliver presentation	Apply
CO5	Apply engineering and management principles to achieve the goal of the project.	Apply

Guidelines

Student Groups with 3 or 4 members should identify a topic of interest in consultation with a Faculty/Advisor. Review the literature and gather information pertaining to the chosen topic. State the objectives and develop a methodology to achieve the objectives. Carryout the design/fabrication or develop codes/programs to achieve the objectives by strictly following steps specified in the teaching plan. Innovative design concepts, performance, scalability, reliability considerations,

aesthetics/ergonomic, user experience and security aspects taken care of in the project shall be given due weight. The progress of the mini project is evaluated based on a minimum of two reviews. The review committee may be constituted by the Head of the Department comprising HoD or a senior faculty member, Mini Project coordinator and project guide. The internal evaluation shall be made based on the progress/outcome of the project, reports and a viva-voce examination, conducted internally by a 3-member committee. A project report is required at the end of the semester. The project has to be demonstrated for its full design specifications.

MARK DISTRIBUTION

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	75	75	-

CONTINOUS ASSESSMENT EVALUATION PATTERN

Attendance	10 marks
Project Guide	15 marks
Project Report	10 marks

PROGRAM ELECTIVE I

CS1U31A	FOUNDATIONS OF MACHINE LEARNING	Category	L	T	P	Credit	Year of Introduction
		PEC	2	1	0	3	2020

PRE-REQUISITE : Nil

COURSE OVERVIEW: This course enables the learners to understand the mathematical foundations of Machine Learning concepts. This course covers Linear Algebra, Probability and Distributions. Concepts in this course help the learners to identify the inherent assumptions & limitations of the current methodologies and develop new Machine Learning solutions.

COURSE OUTCOMES: After the completion of the course the student will be able to

CO 1	Illustrate operations and applications of linear equations, matrix algebra, vector spaces, eigen values & eigenvectors .	Understand
CO 2	Illustrate the concepts of orthogonality & diagonalization.	Understand
CO 3	Solve computational problems using probability and random variables.	Apply
CO 4	Identify an appropriate probability distribution for a given discrete or continuous random variable and use its properties.	Apply
CO 5	Illustrate moment generating function, law of large numbers and central limit theorems	Understand

SYLLABUS

Systems of Linear Equations – Matrices, Solving Systems of Linear Equations. Vector Spaces - Linear Independence, Basis and Rank, Linear Mappings. Norms. Orthonormal Basis, Orthogonal Complement, Orthogonal Projections. Matrix Decompositions, Eigen decomposition and Diagonalization. Probability Space. Random Variables - Discrete Random Variables (Bernoulli Random Variables, Binomial Distribution, Geometric and Poisson Distribution, Continuous Random Variables (Exponential Density, Gamma Density, Normal Distribution, Beta Density). Functions of a Random Variable. Joint Distributions, Conditional Distributions, Functions of Jointly Distributed Random Variables. Expected Values, Expectations of Functions of Random Variables, Expectations of Linear Combinations of Random Variables, Variance and Standard Deviation, Covariance and Correlation, Conditional Expectation. Moment-Generating Function. Limit Theorems, Convergence in Distribution and the Central Limit Theorem. Distributions derived from the Normal Distribution, Sample Mean and the Sample Variance.

TEXT BOOKS

1) Marc Peter Deisenroth, A. Aldo Faisal, Cheng Soon Ong, Mathematics for Machine Learning, Cambridge University Press (freely available at https://mml – book.github.io)
2) John A. Rice, Mathematical Statistics and Data Analysis,University of California, Berkeley, Third Edition, published by Cengage.

REFERENCES

1) Gilbert Strang, Linear Algebra and Its Applications, 4th Edition.
2) Axler, Sheldon, Linear Algebra Done Right, 2015 Springer.
3) Stephen Boyd and LievenVandenberghe, Introduction to Applied Linear Algebra, 2018 published by Cambridge University Press.

COURSE PLAN

Module	Contents	No. of hours
I	Systems of Linear Equations – Matrices, Solving Systems of Linear Equations. Vector Spaces- Linear Independence, Basis and Rank, Linear Mappings.	6
II	Norms - Inner Products, Lengths and Distances, Angles and Orthogonality. Orthonormal Basis, Orthogonal Complement, Orthogonal Projections. Matrix Decompositions - Eigenvalues and Eigenvectors, Eigen decomposition and Diagonalization.	9
III	Probability Space - Sample Spaces, Probability Measures, Computing Probabilities, Conditional Probability, Baye's Rule, Independence. Random Variables - Discrete Random Variables (Bernoulli Random Variables, Binomial Distribution, Geometric and Poisson Distribution, Continuous Random Variables (Exponential Density, Gamma Density, Normal Distribution, Beta Density)	10
IV	Functions of a Random Variable. Joint Distributions - Independent Random Variables, Conditional Distributions, Functions of Jointly Distributed Random Variables. Expected Values - Expected Value of a Random Variable, Expectations of Functions of Random Variables, Expectations of Linear Combinations of Random Variables, Variance and Standard Deviation, Covariance and Correlation, Conditional Expectation	10

V	Moment-Generating Function. Limit Theorems(Proof not expected) - Law of Large Numbers, Convergence in Distribution and the Central Limit Theorem. Distributions derived from the Normal Distribution - Chi-square, t, and F Distributions, Sample Mean and the Sample Variance.	10
	Total hours	45

MARK DISTRIBUTION

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

CONTINOUS INTERNAL EVALUATION PATTERN

Attendance : 10 marks

Continuous Assessment Tests : 25 marks

Continuous Assessment Assignment : 15 marks

CS1U31B	DATA ANALYTICS	Category	L	T	P	Credits	Year of Introduction
		PEC	2	1	0	3	2020

PRE-REQUISITE: NIL

COURSE OVERVIEW: This course helps the learner to understand the basic concepts of data analytics. This course covers mathematics for data analytics, predictive and descriptive analytics of data, 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.

COURSE OUTCOMES: After the completion of the course the student will be able to

CO1	Illustrate the mathematical concepts for data analytics	Understand
CO2	Explain the basic concepts of data analytics	Understand
CO3	Illustrate various predictive and descriptive analytics algorithms	Understand
CO4	Describe the key concepts and applications of Big Data Analytics	Understand
CO5	Demonstrate the usage of Map Reduce paradigm for Big Data Analytics.	Understand
CO6	Use R programming tool to perform data analysis and visualization	Apply

SYLLABUS

Descriptive statistics, Association of two variables, Ordinal and Continuous variable, Probability calculus, Inductive statistics, Interval estimation, Hypothesis Testing, t- test. Introduction to Data Analysis, Analytics Process Model, Analytical Model Requirements. Data Analytics Life Cycle overview. Basics of data collection, sampling, preprocessing and dimensionality reduction. Supervised Learning. Unsupervised Learning. Association Rule Mining - Apriori algorithm. Big Data Overview, Example Applications.Big Data Analytics using Map Reduce and Apache Hadoop. Overview of modern data analytic tools.Data Analysis Using R, Graphical User Interfaces, Data Import and Export, Exploratory Data Analysis - Visualization Before Analysis, Statistical Methods for Evaluation.

TEXT BOOKS

1) Bart Baesens," Analytics in a Big Data World: The Essential Guide to Data Science and its Business Intelligence and Analytic Trends", John Wiley & Sons, 2013.
2) David Dietrich, "EMC Education Services, Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data", John Wiley & Sons, 2015.
3) Jaiwei Han, Micheline Kamber, "Data Mining Concepts and Techniques", Elsevier, 2006.
4) Christian Heumann and Michael Schomaker, "Introduction to Statistics and DataAnalysis", Springer, 2016

REFERENCES

1) Margaret H. Dunham, Data Mining: Introductory and Advanced Topics. Pearson, 2012.
2) Michael Berthold, David J. Hand, Intelligent Data Analysis, Springer, 2007.

COURSE PLAN

Module	Contents	No. of hours
I	Descriptive statistics - Measures of central tendency and dispersion, Association of two variables - Discrete variables, Ordinal and Continuous variable, Probability calculus - probability distributions, Inductive statistics - Point estimation, Interval estimation, Hypothesis Testing - Basic definitions, t- test	9
II	Introduction to Data Analysis - Analytics, Analytics Process Model, Analytical Model Requirements. Data Analytics Life Cycle overview. Basics of data collection, sampling, preprocessing and dimensionality reduction	6

III	Supervised Learning - Classification, Naive Bayes, KNN, Linear Regression. Unsupervised Learning - Clustering, Hierarchical algorithms – Agglomerative algorithm, Partitional algorithms - K- Means. Association Rule Mining - Apriori algorithm	10
IV	Big Data Overview – State of the practice in analytics, Example Applications - Credit Risk Modeling, Business Process Analytics.Big Data Analytics using Map Reduce and Apache Hadoop, Developing and Executing a HadoopMapReduce Program.	10
V	Overview of modern data analytic tools.Data Analysis Using R - Introduction to R - R Graphical User Interfaces, Data Import and Export, Attribute and Data Types, Descriptive Statistics, Exploratory Data Analysis - Visualization Before Analysis, Dirty Data, Visualizing a Single Variable, Examining Multiple Variables, Data Exploration Versus Presentation, Statistical Methods for Evaluation	10
	Total hours	45

MARK DISTRIBUTION

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

CONTINOUS INTERNAL EVALUATION PATTERN

Attendance : 10 marks

Continuous Assessment Tests : 25 marks

Continuous Assessment Assignment : 15 marks

CS1U31C	FOUNDATIONS OF SECURITY IN COMPUTING	Category	L	T	P	Credit	Year Of Introduction
		PEC	2	1	0	3	2020

PRE-REQUISITE:MA0U20G- Discrete Mathematical Structures, CS1U30K-Operating Systems, CS1U20E-Database Management Systems

COURSE OVERVIEW: The purpose of this course is to create awareness among learners about the fundamentals of security and number theory. This course covers Integer & Modular Arithmetic, Primes & Congruences, Discrete Logarithms & Elliptic Curve Arithmetic and an overview of computer security. The concepts covered in this course enable the learners in effective use of cryptographic algorithms and to identify the security threats in computing.

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

CO1	Illustrate the operations and properties of algebraic structures, integer arithmetic and modular arithmetic.	Understand
CO2	Use the concepts of prime numbers and factorization for ensuring security in computing systems	Apply
CO3	Illustrate the concepts of Linear Congruence, Primitive Roots, Discrete Logarithms and Elliptic Curve Arithmetic	Understand
CO4	Summarize the threats and attacks related to computer and program security.	Understand
CO5	Outline the key aspects of operating system and database security	Understand

SYLLABUS

Integer arithmetic-Modular arithmetic-Algebraic structures-Prime numbers-Fermat's theorem-Primality testing-Euler's theorem- Factorization-Linear congruence-Simultaneous linear congruence-Solving congruence modulo prime powers-Primitive roots-Elliptic curve arithmetic-Prime curves, Binary curves-Introduction to computer security-Browser attack types-Web attacks -Email attack types-Introduction to program security-Operating system security -Database security.

TEXT BOOKS

1) Behrouz A Forouzan, Cryptography and Network Security, 3/e, Tata McGraw-Hill.

2) Charles P Pfleeger, Shari Lawrence Pfleeger, Jonathan Margulies, Security in Computing, 5/e, Prentice Hall.

3) G.A. Jones & J.M. Jones, Elementary Number Theory, Springer UTM, 2007

REFERENCE

1) William Stallings, Cryptography and Network Security Principles and Practices, 4/e, Pearson Ed.

COURSE PLAN

Module	Contents	No. of hours
I	Integer arithmetic - Integer division, Divisibility, Greatest Common Divisor (GCD), Euclid's algorithm for GCD, Extended Euclid's algorithm, Linear Diophantine Equations. Modular arithmetic - Operations, Properties. Algebraic structures - Groups, Rings, Fields, Finite fields, GF(p), GF (2^n).	8
II	Prime numbers - Prime numbers and prime-power factorization, Fermat and Mersenne primes, Fermat's theorem, Applications, Euler's theorem, Euler's totient function, Applications. Primality testing – Deterministic algorithms and Probabilistic algorithms. Factorization - Fermat's factorization, Pollard p-1 method.	10
III	Linear congruence - Simultaneous linear congruence, Chinese Remainder Theorem (CRT). Congruence with a prime - Power modulus, Arithmetic modulo p, Pseudoprimes and Carmichael numbers, Solving congruence modulo prime powers. Primitive roots - Existence of primitive roots for primes, Discrete logarithms. Elliptic curve arithmetic – Prime curves, Binary curves, Addition of two points, Multiplication of a point by a constant.	10
IV	Introduction to computer security – Threats, Vulnerabilities, Controls. Browser attack types, Web attacks targeting users, Email attack types. Introduction to program security - Non-malicious programming oversights, Malware.	9
V	Operating system security – Security in operating system, Security in design of operating system. Database security – Security requirements of databases, Reliability and integrity, Database disclosure.	8
	Total Hours	**45**

MARK DISTRIBUTION

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

CONTINOUS INTERNAL EVALUATION PATTERN

Attendance : 10 marks

Continuous Assessment Tests : 25 marks

Continuous Assessment Assignment : 15 marks

CS1U31D	AUTOMATED VERIFICATION	Category	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PEC	2	1	0	3	2020

PRE-REQUISITE: NIL

COURSE OVERVIEW : This course is intended to impart the basic theory and algorithm for an automatic verification process namely model checking. This course covers finite-state modelling of hardware/software, linear-time properties, classification of linear-time properties, Linear Temporal Logic (LTL) - a formal language for property specification, LTL model checking algorithm and model checking case studies. This course enables the learners to prove correctness of a hardware/software used in safety critical systems in domains such as avionics, health care and automotive.

COURSE OUTCOMES: After the completion of the course the student will be able to

CO1	Illustrate an application for model checking.	Understand
CO2	Describe finite-state modelling for hardware and software.	Understand
CO3	Identify linear-time properties required to represent the requirements of a system.	Apply
CO4	Identify a given linear-time property in Linear Temporal Logic (LTL).	Apply
CO5	Experiment with LTL model checking using the tool Symbolic Analysis Laboratory (SAL).	Apply

SYLLABUS

System Verification, Transition Systems - Direct Predecessors and Successors, Executions, Linear-Time (LT) Properties - Deadlock - Maximal and Initial Path Fragment. Regular Properties - Automata on Infinite Words - ω -Regular Languages and Properties, Nondeterministic Buchi Automata (NBA), Deterministic Buchi Automata (DBA), Generalised Buchi Automata, Linear Temporal Logic (LTL) - Automata Based LTL Model Checking.

TEXT BOOKS

1) Christel Baier and Joost-Pieter Katoen, Principles of Model Checking, The MIT Press. (Modules 1 - 4)
2) Leonardo de Moura, Sam Owre and N. Shankar, The SAL Language Manual, SRI International (http://sal.csl.sri.com/doc/language-report.pdf, Chapters 1, 3, 4, 5, 6, 7) (Module 5)

REFERENCES

1) SAL Examples (http://sal.csl.sri.com/examples.shtml) (Module 5)

COURSE PLAN

Module	Contents	No. of hours
I	System Verification – Hardware and Software Verification, Model Checking, Characteristics of Model Checking. Transition Systems – Transition System, Direct Predecessors and Successors, Terminal State, Deterministic Transition System. Executions - Execution Fragment, Maximal and Initial Execution Fragment, Execution, Reachable States.	8
II	Linear-Time (LT) Properties - Deadlock. Linear-Time Behavior - Paths and State Graph, Path Fragment, Maximal and Initial Path Fragment, Path. Traces - Trace and Trace Fragment, LT Properties - LT Property, Satisfaction Relation for LT Properties, Trace Equivalence and LT Properties. Safety Properties and Invariants - Invariants, Safety Properties, Trace Equivalence and Safety properties. Liveness Properties - Liveness Property, Safety vs. Liveness Properties. Fairness - Fairness, Unconditional, Weak and Strong Fairness, Fairness Strategies, Fairness and Safety. (Definition and examples only for all topics - no proof required).	10
III	Regular Properties - Model Checking Regular Safety properties - Regular Safety property, Verifying Regular Safety Properties. Automata on Infinite Words - ω -Regular Languages and Properties, Nondeterministic Buchi Automata (NBA), Deterministic Buchi Automata (DBA), Generalised Buchi Automata (Definitions only). Model Checking ω -Regular Properties - Persistence Properties and Product, Nested Depth-First Search (Only algorithms required).	11
IV	Linear Temporal Logic (LTL) - Syntax, Semantics, Equivalence of LTL Formulae, Weak Until, Release and Positive Normal Form, Fairness, Safety and Liveness in LTL (Definitions only). Automata Based LTL Model Checking (Algorithms and examples only).	8

V	Introduction - Introduction to the tool Symbolic Analysis Laboratory (SAL). The Language of SAL - The expression language, The transition Language, The module language, SAL Contexts. SAL Examples - Mutual Exclusion, Peterson's Protocol, Synchronous Bus Arbiter, Bounded Bakery protocol, Bakery Protocol, Traffic Signalling System.	8
	Total Hours	45

MARK DISTRIBUTION

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

CONTINOUS INTERNAL EVALUATION PATTERN

Attendance : 10 marks

Continuous Assessment Tests : 25 marks

Continuous Assessment Assignment : 15 marks (Out of 15, 10 marks shall be given for a model checking projects to be implemented in SAL).

CS1U31E	PROGRAMMING IN PYTHON	Category	L	T	P	Credit	Year of Introduction
		PEC	2	1	0	3	2020

PRE-REQUISITE: NIL

COURSE OVERVIEW: The objective of the course is to equip the learners to develop multi-module software solutions for real world computational problems using Python. It encompasses the Python programming environment, syntax, data representations, intermediate level features, GUI programming, Object Oriented Programming and data processing. This course lays the foundation to develop modular software solutions including complex interactive applications, network applications, and data-driven intelligent applications.

COURSE OUTCOMES: After the completion of the course the student will be able to

CO1	Develop, test and debug Python programs	Apply
CO2	Make use of conditional (if, if-else and if-else-if) and iterative (while and for) statements in Python programs.	Apply
CO3	Develop programs by utilizing the Python programming constructs such as Lists, Tuples, Sets and Dictionaries.	Apply
CO4	Develop graphical user interface for solutions using Python libraries.	Apply
CO5	Implement Object Oriented programs with exception handling.	Apply
CO6	Develop programs in Python to process data stored in files by utilizing Numpy, Matplotlib, and Pandas.	Apply

SYLLABUS

Getting started with Python, Interactive shell, IDLE, iPython Notebooks, The software development process, Numeric data types and character sets, Expressions, Using inbuilt functions and modules. Control statements, Strings and text files,Design with Functions, Lists, Dictionaries. Case Study – Data Structure Selection. Graphics, Image Processing, Graphical User Interfaces, Design with classes, Exceptions, The os and sys modules, NumPy - Basics, Matplotlib, Working with CSV files. Introduction to Micro services using Flask.

TEXT BOOKS

1. Kenneth A Lambert., Fundamentals of Python : First Programs, 2/e, Cengage Publishing, 2016
2. Wes McKinney, Python for Data Analysis, 2/e, Shroff / O'Reilly Publishers, 2017
3. Flask: Building Python web services, Jack Stouffer, Shalabh Aggarwal, Gareth Dwyer, PACKT Publishing Limited, 2018

REFERENCES

1. Zed A Shaw, Learn Python 3 The Hard Way, Addison-Wesley, 2017
2. Allen B. Downey, Think Python: How to Think Like a Computer Scientist, 2/e, Schroff, 2016
3. Michael Urban and Joel Murach, Python Programming, Shroff/Murach, 2016
4. Charles Severance. Python for Informatics: Exploring Information, CreateSpace Independent Publishing Platform, 2013.

COURSE PLAN

Module	Contents	No. of hours
I	Getting started with Python programming – Interactive shell, IDLE, iPython Notebooks, Detecting and correcting syntax errors, How Python works. The software development process – A case study. Basic coding skills – strings, assignment, and comments, Numeric data types and character sets, Expressions, Using inbuilt functions and modules. Control statements – Iteration with for/while loop, Formatting text for output, A case study, Selection structure (if-else, switch- case), Conditional iteration with while, A case study, Testing control statements, Lazy evaluation.	8
II	Strings and text files – Accessing characters, substrings, Data encryption, Strings and number system, String methods, Text files, A case study on text analysis. Design with Functions – Functions as Abstraction Mechanisms, Problem solving with top-down design, Design with recursive functions, Managing a program's namespace, Higher-Order Functions. Lists - Basic list Operations and functions, List of lists, Slicing, Searching and sorting list, List comprehension. Work with tuples. Sets. Work with dates and times, A case study with lists. Dictionaries - Dictionary functions, dictionary literals, adding and removing keys, accessing and replacing values, traversing dictionaries, reverse lookup. Case Study – Data Structure Selection.	9
III	Graphics – Terminal-based programs, Simple Graphics using Turtle, Operations, 2D Shapes, Colors and RGB Systems, A case study. Image	9

	Processing – Basic image processing with inbuilt functions. Graphical User Interfaces – Event-driven programming, Coding simple GUI-based programs : Windows, Labels, Displaying images, Input text entry, Popup dialog boxes, Command buttons, A case study.	
IV	Design with classes - Objects and Classes, Methods, Instance variables, Constructor, Accessor and Mutator, Data-Modeling Examples, Structuring classes with inheritance and polymorphism. Abstract classes, Interfaces, Exceptions - Handle a single exception, handle multiple exceptions.	9
V	The os and sys modules, NumPy - Basics, Creating arrays, Arithmetic, Slicing, Matrix Operations, Random numbers. Plotting and visualization. Matplotlib - Basic plot, Ticks, Labels, and Legends. Working with CSV files. – Pandas - Reading, Manipulating, and Processing Data. Introduction to Micro services using Flask.	10
	Total Hours	45

MARK DISTRIBUTION

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

CONTINOUS INTERNAL EVALUATION PATTERN

Attendance : 10 marks

Continuous Assessment Tests : 25 marks

Continuous Assessment Assignment : 15 marks

CS1U31F	ADVANCED DATA COMMUNICATION	Category	L	T	P	Credits	Year of Introduction
		PEC	2	1	0	3	2020

PRE-REQUISITE: NIL

COURSE OVERVIEW: The purpose of this course is to prepare learners to understand the communication entities and the associated issues in data transmission. This course covers fundamental concepts of data transmission in digital and analog form, transmission media, concepts of encoding, multiplexing, spread spectrum and switching methods. This course helps the learner to gain insight into the important aspects of data communication and computer networking systems and enables to apply in practical applications.

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

CO1	Identify the characteristics of signals for analog and digital transmissions	Apply
CO2	Identify the issues in data transmission	Apply
CO3	Choose appropriate signal encoding techniques for a given scenario	Apply
CO4	Use multiplexing and spread spectrum technologies	Apply
CO5	Use error detection, correction and switching techniques in data communication	Apply
CO6	Identify the characteristics of Mobile Networks	Apply

SYLLABUS

Periodic analog signals, Transmission impairments, Data rate limits, Nyquist bandwidth, Noisy channel, Shannon's capacity formula. Digital data to a digital signal, Sampling theorem, Pulse Code Modulation (PCM), Delta Modulation (DM). Digital data to analog signal- ASK, FSK, PSK. Analog data to analog signal- AM, FM, PM. Multiplexing, Spread spectrum techniques, Digital data communication techniques, Detecting and correcting errors, Basic principles of switching - Circuit switching, Packet switching, message switching. Mobile Communications, Introduction to GSM-Services, GPRS – Architecture, UMTS – Architecture.

TEXT BOOKS

1) Forouzan B. A., Data Communications and Networking, 5/e, McGraw Hill, 2013.
2) William Stallings, Data and Computer Communication 9/e, Pearson Education, Inc.

REFERENCE

1) Schiller J., Mobile Communications, 2/e, Pearson Education, 2009.
2) Curt M. White, Fundamentals of Networking and Communication 7/e, Cengage learning.

COURSE PLAN

Module	Contents	No. of hours
I	Periodic analog signals- Sine wave, Amplitude, Phase, Wavelength, Time and frequency domain, Bandwidth. Analog & digital data and signals. Transmission impairments - Attenuation, Delay distortion, Noise. Data rate limits - Noiseless channel, Nyquist bandwidth, Noisy channel, Shannon's capacity formula.	9
II	Digital data to digital signal – Non-Return-to-Zero (NRZ), Return-to-Zero (RZ), Multilevel binary, Biphase. Analog data to digital signal - Sampling theorem, Pulse Code Modulation (PCM), Delta Modulation (DM). Digital data to analog signal - Amplitude Shift Keying (ASK), Frequency Shift Keying (FSK), Phase Shift Keying (PSK). Analog data to analog signal - Amplitude Modulation (AM), Frequency Modulation (FM), Phase Modulation (PM).	9
III	Multiplexing - Frequency Division Multiplexing (FDM), Wavelength Division Multiplexing (WDM), Time Division Multiplexing (TDM), Characteristics, Synchronous TDM, Statistical TDM. Spread spectrum techniques - Direct Sequence Spread Spectrum (DSSS), Frequency Hopping Spread Spectrum (FHSS), Code Division Multiplexing, Code Division Multiple Access (CDMA).	9
IV	Digital data communication techniques – Asynchronous transmission, Synchronous transmission. Detecting and correcting errors - Types of errors, Parity check, Checksum, Cyclic Redundancy Check (CRC), Forward Error Correction (FEC), Hamming distance, Hamming code. Basic principles of switching - Circuit switching, Packet switching, Message switching.	9
V	Mobile Communications: Generation of mobile communication Technologies, Introduction to GSM- Services & Architecture, Protocols, Connection Establishment, Routing, GPRS – Architecture, UMTS – Architecture, Handover.	9
	Total Hours	45

MARK DISTRIBUTION

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

CONTINOUS INTERNAL EVALUATION PATTERN

Attendance	: 10 marks
Continuous Assessment Tests	: 25 marks
Continuous Assessment Assignment	: 15 marks

Course Code	Course Name	Category	L	T	P	Credit	Year of Introduction
CS1U31G	Applied Data Science with Python	PEC	2	0	2	3	2022

PRE-REQUISITES: MA0U10A Linear Algebra and Calculus, CS1U20E Database Management Systems

COURSE OVERVIEW: Applied Data Science with Python course, a comprehensive journey designed to equip you with the knowledge, skills, and tools necessary to thrive in today's data-driven world. This course serves as a bridge between theoretical foundations and real-world applications, offering you the hands-on experience required to tackle actual data challenges.

COURSE OUTCOMES

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

CO 1	Make use of Pandas library to effectively process complex datasets.	Apply
CO 2	Create effective data visualization using Python libraries.	Apply
CO 3	Build machine learning and deep learning models using Python libraries.	Apply
CO 4	Apply Natural Language Processing techniques in data analysis.	Apply
CO 5	Develop models using Time series data to solve real world problems.	Apply

SYLLABUS

Introduction to Data Science and Python ,Data Manipulation with Pandas, Advanced Pandas-Merging, Joining and Concatenating, Pivot Tables, Time Series Analysis in Pandas, Data Visualization with Matplotlib and Seaborn, Seaborn for Statistical Visualization, Advanced Visualization Techniques, Exploratory Data Analysis (EDA),Visual EDA , Introduction to Machine Learning with Scikit-learn, Regression Techniques, Classification Techniques, Model Evaluation, Advanced Machine Learning Techniques, Ensemble Methods, Unsupervised Learning, Hyper parameter Tuning, Introduction to Deep Learning with TensorFlow/Keras.

Understanding Neural Networks, Implementing Neural Networks, Introduction to Convolutional Neur Networks, Natural Language Processing with NLTK and SpaCy, Text Classification and Sentimer Analysis, Named Entity Recognition. Time Series Forecasting with Python, Forecasting Models, Da Science Project.

TEXT BOOKS

1) McKinney.W, 2022. Python for data analysis. " O'Reilly Media, Inc.".
2) Goodfellow I, Bengio Y and Courville A, 2016. Deep learning. MIT press.
3) Geron A, 2022 Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow. " O'Reilly Media, Inc.".
4) Hapke.H, Howard.C and Lane. H, 2019. Natural Language Processing in Action: Understanding, analyzing, and generating text with Python. Simon and Schuster.

REFERENCES

1) VanderPla, 2016. Jake. Python Data Science Handbook.
2) Andrew Park, 2021, Data Science for Beginners
3) Muller.A.C and Guido S, 2016. Introduction to machine learning with Python: a guide for data scientists. " O'Reilly Media, Inc.".
4) Muller A.C and Guido S, 2017. Introduction to machine learning with Python. O'Reilly.
5) Heaton, J., 2018. Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning: The MIT Press, 2016, 800 pp, ISBN: 0262035618. Genetic programming and evolvable machines, 19(1-2), pp.305-307.

COURSE PLAN

Module	Contents	Hours
I	Introduction to Data Science and Python, What is Data Science, Introduction to Python, Why Python for Data Science?, Data Manipulation with Pandas, Introduction to Pandas ,Data Frames and Series, Basic Operations, Advanced Pandas	8
II	Data Visualization with Matplotlib and Seaborn, Matplotlib Basics ,Seaborn for Statistical Visualization ,Advanced Visualization Techniques Exploratory Data Analysis (EDA),Understanding EDA , Visual EDA -Univariate, Bivariate, and Multivariate Analysis- Correlation and Heatmaps.	8
III	Introduction to Machine Learning with Scikit-learn , Understanding Machine Learning Regression Techniques, Classification Techniques Logistic Regression. Model Evaluation, Advanced Machine Learning Techniques, Ensemble Methods Unsupervised Learning, Hyperparameter Tuning	10
IV	Introduction to Deep Learning with TensorFlow/Keras, Understanding Neural Networks, Implementing Neural Networks ,Introduction to Convolutional Neural Networks. Natural Language Processing with NLTK and SpaCy, Basics of NLP, Text Classification and Sentiment Analysis, Named Entity Recognition	11
V	Time Series Forecasting with Python, Basics of Time Series Analysis Forecasting Models - ARIMA, Exponential Smoothing Practical Data Science Project-Complete End-to-End Project Walkthrough	8
	Total Hours	45

MARK DISTRIBUTION

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

CONTINOUS INTERNAL EVALUATION PATTERN

Attendance : 10 marks

Continuous Assessment Tests : 25 marks

Continuous Assessment Assignment : 15 marks

SEMESTER 6

B.TECH(MINOR)

CS0M30D	INTRODUCTION TO SOFTWARE TESTING	Category	L	T	P	Credits	Year of Introduction
		VAC	3	1	0	4	2020

PRE-REQUISITE: Nil

COURSE OVERVIEW: This is a course in theoretical computer science that includes test cases for white-box, black- box, and grey-box approaches. This course describes the various techniques for test case design used to test software artifacts, including requirements, design, and code. The course includes different techniques for test case design based on graphs, programming language syntaxes and inputs. The course also covers symbolic execution using the PEX tool.

COURSE OUTCOMES: After the completion of the course the student will be able to:-

CO1	List a range of different software testing techniques and be able to apply specific unit testing methods to the projects using Junit**.**	Understand
CO2	Explain mutation testing method for a given piece of code to identify hidden defects that can't be detected using other testing methods.	Understand
CO3	Explain graph coverage criteria in terms of control flow graph and data flow graph for a given program.	Understand
CO4	Demonstrate the importance of black-box approaches in terms of domain and functional testing.	Understand
CO5	Illustrate the use of PEX tool with symbolic execution.	Apply

SYLLABUS

Some Popular Errors – Ariane 5, Therac 25, Intel Pentium Bug. What is Software testing? Why should it be tested? Software Testing Terminologies - Verification, Validation and Testing, Faults,

Error and Bug, Test cases, Coverage Criteria. Types of Testing-. Testing Methods - Mutation and Mutants, Mutation operators, Mutation score. Junit - Framework for Unit testing. Case Study - Mutation testing using Junit and Muclipse. Overview of Graph Coverage Criteria. Data Flow Criteria - du paths, du pairs. Subsumption Relationships among Graph Coverage Criteria. Graph Coverage for Source Code - Control flow graphs for code, CFG: If statement, CFG: If statement with return, CFG: Switch-case, CFG: Loops, CFG: Exceptions (try-catch). Class inheritance testing: Domain Testing / Input Space Partitioning - Partitions of a set. Input domain modelling - Interface-based approach, Functionality-based approach. Identifying values. Multiple partitions of the input domain - Functional Testing - Functional Testing Concepts of Howden. Functional testing - Important Steps. Types of Functional testing - Equivalence Class Partitioning, Boundary Value Analysis, Decision Tables, Random Testing. An Introduction to PEX - Parameterized Unit Testing, The Testing Problem. Symbolic Execution – Example, Symbolic execution tree. PEX application Case Study – PEX.

TEXT BOOKS

1. Paul Ammann and JeffOffutt , Introduction to Software Testing.
2. KshirasagarNaik and PriyadarshiTripathy, Software Testing And Quality Assurance: Theory And Practice.

REFERENCE MATERIALS

1. https://www.csc.ncsu.edu/academics/undergrad/honors/thesis/muclipsebinder.pdf- Muclipse tutorial.
2. King, James C, “Symbolic Execution and Program Testing”, Association for Computing Machinery, July 1976.

COURSE PLAN

Module	Contents	No. of hours
I	Some Popular Errors – Ariane 5, Therac 25, Intel Pentium Bug. What is Software testing? Why should it be tested? Software Quality, Role of Testing. Testing Process - Level 0 thinking, Level 1 thinking, Level 2 thinking, Level 3 thinking, Level 4 thinking. Software Testing Terminologies - Verification, Validation and Testing, Faults, Error and Bug, Test cases, Coverage Criteria. Types of Testing- Unit testing, integration testing, System testing, Acceptance testing, Beta testing, Functional testing, Stress testing, Performance testing, Usability testing and Regression testing. Testing Methods - Black Box testing, White Box testing, Grey Box testing.	12
II	Concept of Unit testing. Static Unit testing. Dynamic Unit testing - Control Flow testing, Data Flow testing, Domain testing, Functional Program testing. Mutation testing - Mutation and Mutants, Mutation operators, Mutation	10

	score. Junit - Framework for Unit testing. Case Study - Mutation testing using Junit and Muclipse.	
III	Overview of Graph Coverage Criteria. Structural Graph Coverage Criteria - Node/vertex coverage, Edge coverage, Edge pair coverage, Path coverage, Complete path coverage, Prime path coverage, Complete round trip coverage, Simple round trip coverage. Data Flow Criteria - du paths, du pairs. Subsumption Relationships among Graph Coverage Criteria. Graph Coverage for Source Code - Control flow graphs for code, CFG: If statement, CFG: If statement with return, CFG: Switch-case, CFG: Loops, CFG: Exceptions (try-catch). Example program – Statistics. Graph Coverage for Design Elements - Call graphs and classes, Class inheritance testing: Coverage criteria, Coverage criteria on inheritance graph, Data flow at the design level, Inter-procedural DU pairs, Coupling du-pairs example. Example - Quadratic Root. Case Study - Graph Based testing using JUnit Framework.	**14**
IV	Domain Testing / Input Space Partitioning - Partitions of a set. Input domain modelling - Interface-based approach, Functionality-based approach. Identifying values. Multiple partitions of the input domain - All Combinations Coverage (ACoC), Each Choice Coverage (ECC), Pair-wise Coverage, T-wise Coverage, Base Choice Coverage, Multiple Base Choices Coverage. TriTyp example. Functional Testing - Functional Testing Concepts of Howden. Functional testing - Important Steps. Types of Functional testing - Equivalence Class Partitioning, Boundary Value Analysis, Decision Tables, Random Testing. Case Study - Black Box testing approaches using JUnit.	**12**
V	Introduction to Grey Box testing - Why Grey Box testing, Gray Box Methodology, Advantages and Disadvantages. Techniques of Grey Box Testing - Matrix Testing, Regression Testing, Orthogonal Array Testing or OAT, Pattern Testing. An Introduction to PEX - Parameterized Unit Testing, The Testing Problem. Symbolic Execution – Example, Symbolic execution tree. PEX application Case Study – PEX.	**12**

MARK DISTRIBUTION

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

CONTINOUS INTERNAL EVALUATION PATTERN

Attendance : 10 marks

Continuous Assessment Tests : 25 marks

Continuous Assessment Assignment : 15 marks

CS0M30E	CONCEPTS IN DEEP LEARNING	Category	L	T	P	Credits	Year of Introduction
		VAC	3	1	0	4	2020

PRE-REQUISITE:MA0U10A LINEAR ALGEBRA AND CALCULUS

COURSE OVERVIEW: This course aims to introduce the learner to an overview of the concepts and algorithms involved in deep learning. Deep learning is a subfield of machine learning, a subfield of artificial intelligence. Basic concepts and application areas of machine learning, deep networks, convolutional neural network and recurrent neural network are covered here. This is a foundational program that will help students understand the capabilities, challenges, and consequences of deep learning and prepare them to participate in the development of leading-edge AI technology. They will be able to gain the knowledge needed to take a definitive step in the world of AI.

COURSE OUTCOMES:

CO1	Demonstrate basic concepts in machine learning.	Understand
CO2	Illustrate the validation process of machine learning models using hyper-parameters and validation sets.	Understand
CO3	Demonstrate the concept of the feed forward neural network and its training process.	Apply
CO4	Build CNN and Recurrent Neural Network (RNN) models for different use cases.	Apply
CO5	Use different neural network/deep learning models for practical applications.	Apply

SYLLABUS

INTRODUCTION TO DEEP LEARNING, Key components - Data, models, objective functions, optimization algorithms, Learning algorithm. Supervised learning, Unsupervised learning, Reinforcement learning, Historical Trends in Deep Learning. Overfitting, underfitting, hyper parameter tuning. Neural Networks. Gradient Descent solution for Perceptron, Multilayer perceptron. Introduction to optimization linear least squares. Stochastic gradient descent, Building ML algorithms and challenges. Convolutional Neural Networks. Practical challenges of common deep learning architectures. Case study-pretrained models. Recurrent neural networks, recurrent networks, recursive neural networks, modern RNNs LSTM and GRU, Practical use cases for RNNs. Applications – computer vision, speech recognition, natural language processing. Research Areas – auto encoders, representation learning, boltzmann machines, deep belief networks.

TEXT BOOKS

1. Ian Goodfellow, YoshuaBengio, Aaron Courville, Deep Learning, MIT Press 2015 ed.
2. Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola, Dive into Deep Learning, August 2019.
3. Neural Networks and Deep Learning, Aggarwal, Charu C., c Springer International Publishing AG, part of Springer Nature 2018

REFERENCE MATERIALS

1. Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks by Russell Reed, Robert J MarksII, A Bradford Book,2014
2. Practical Convolutional Neural Networks by MohitSewak, Md.Rezaul Karim, Pradeep Pujari, Packt Publishing 2018
3. Hands-On Deep Learning Algorithms with Python by Sudharsan Ravichandran,Packt Publishing 2019
4. Deep Learning with Python by Francois Chollet, Manning Publications Co.,2018.

COURSE PLAN

Module	Contents	No. of hours
	INTRODUCTION TO DEEP LEARNING (General Instructions: Instructors are to introduce students to any one software platform and demonstrate the working of the algorithms in the syllabus using suitable use cases and public datasets to give a better	

	understanding of the concepts discussed. Tutorial hour may be used for this purpose)	
I	Key components - Data, models, objective functions, optimization algorithms, Learning algorithm. Supervised learning- regression, classification, tagging, web search, page ranking, recommender systems, sequence learning, Unsupervised learning, Reinforcement learning, Historical Trends in Deep Learning. Other Concepts - overfitting, underfitting, hyperparameters and validation sets, estimators, bias and variance.	**10**
II	Neural Networks –Perceptron, Gradient Descent solution for Perceptron, Multilayer perceptron, activation functions, architecture design, chain rule, back propagation, gradient based learning. Introduction to optimization– Gradient based optimization, linear least squares. Stochastic gradient descent, Building ML algorithms and challenges.	**10**
III	Convolutional Neural Networks – convolution operation, motivation, pooling, Structure of CNN, Convolution and Pooling as an infinitely strong prior, variants of convolution functions, structured outputs, data types, efficient convolution algorithms. Practical challenges of common deep learning architectures- early stopping, parameter sharing, dropout. Case study: AlexNet, VGG, ResNet	**13**
IV	Recurrent neural networks – Computational graphs, RNN design, encoder – decoder sequence to sequence architectures, deep recurrent networks, recursive neural networks, modern RNNs LSTM and GRU, Practical use cases for RNNs.	**13**
V	Applications – computer vision, speech recognition, natural language processing, common word embedding: continuous Bag-of-Words, Word2Vec, global vectors for word representation (GloVe). Research Areas – autoencoders, representation learning, boltzmann machines, deep belief networks	**14**

MARK DISTRIBUTION

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

CONTINOUS INTERNAL EVALUATION PATTERN

Attendance : 10 marks

Continuous Assessment Tests : 25 marks

Continuous Assessment Assignment : 15 marks

CS0M30F	WIRELESS NETWORKS AND IoT APPLICATIONS	Category	L	T	P	Credit	Year of Introduction
		VAC	3	1	0	4	2020

PRE-REQUISITE: CS0M20C- Data Communication, CS0M20F- Introduction to Computer Networks

COURSE OVERVIEW: This course equips the learners with fundamental wireless technologies for the Internet of Things (IoT) and the IoT ecosystem. It covers the underlying concepts in wireless networks, communication mechanisms, protocols, hardware, software, and the cloud platforms for IoT. The students will be able to design smart IoT applications for real world problems.

COURSE OUTCOMES: After the completion of the course the students will be able to

CO1	Recognize wireless technologies required for IoT ecosystem	Understand
CO2	Perceive the concept of IoT and M2M architecture, IoT examples, and Data Management in IoT	Apply
CO3	Outline the hardware components used in IoT including Sensors, Actuators and development boards	Understand
CO4	Explain the software components of IoT	Understand
CO5	Demonstrate the protocols used in IoT and build IoT Programs	Apply
CO6	Build IoT-based smart real-time applications such as Smart Healthcare, Smart Agriculture, Smart Environment and Smart Home	Apply

SYLLABUS

Internet of Things-Role of Things and the Internet-Wireless IoT-Network Topologies, Types of Networks-Role of Wireless Standards in IoT-Protocols for Wireless IoT-Internet of Things - IoT Architectural View-IoT Application Areas-Design Principles for Web Connectivity-Internet Connectivity Principles-IP addressing in the IoT-Data Acquiring And Storage for IoT Services-Big data Acquiring Methods-Cloud Computing for Data storage-Sensor Technologies for IoT Devices-Sensor Data Communication Protocols-Embedded Computing Basics, Embedded Hardware Unit-Programming using Arduino-Business Models and Processes using IoT-Business Models and Processes using IoT.

TEXT BOOKS

1. Daniel Chew, “Wireless Internet of Things -A Guide to the lower layers”, IEEE Standards and Association, IEEE Press, Wiley
2. Rajkamal, “Internet of Things : Architecture and Design Principles”, McGraw Hill (India) Private Limited.

REFERENCE MATERIALS

1. ArshadeepBahga, Vijay Madisetti, “Internet of Things: A hands-on approach”, University Press, 2015 (First edition)
2. Dieter Uckelmann, Mark Harrison, Michahelles Florian (Ed.), Architecting the internet of things, Springer, 2011
3. Dr.OvidiuVermesan, Dr. Peter Friess, Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems, River Publishers, 2013
4. Simon Monk, “Programming Arduino: Getting Started with Sketches”, McGraw Hill Publications.

COURSE PLAN

Module	Contents	No. of hours
I	Internet of Things, Role of Things and the Internet, Wireless IoT. Wireless Networks - Network Topologies, Types of Networks. Role of Wireless Standards in IoT. Protocol Stack - OSI Model, TCP/IP Model, IEEE 802 Reference Model, Protocols for Wireless IoT. Bluetooth - Transceiver, Frequency Channels, Typical Range, Access and Spread Spectrum, Modulation and Data Rate, Error Correction and Detection, Network Topology. ITU G.9959, Zwave, IEEE 802.15.4, Zigbee Specification, Thread, WiFi, 6LowPAN, IPv6, LoRaWAN.	10
II	Internet of Things - IoT Architectural View, Technology Behind IoT - Server End Technology, Sources of Internet of Things, M2M Communication. IoT Application Areas. IoT Examples. IoT Data Management - Device Management Gateways. Design Principles for Web Connectivity - Web Communication Protocols for Connected Devices, Web Connectivity for Connected Devices using Gateways.	8

	Internet Connectivity Principles – Internet Connectivity, Internet based communication, IP addressing in the IoT.	
III	Data Acquiring and Storage for IoT Services- Organization of Data, Big data, Acquiring Methods, Management Techniques, Analytics, Storage Technologies. Cloud Computing for Data storage - IoT Cloud based Services using Xively, Nimbits, and Other Platforms. Sensor Technologies for IoT Devices - Sensor Technology, Participatory Sensing, Industrial IoT and Automotive IoT, Actuators for Various Devices, Sensor Data Communication Protocols, Wireless Sensor network Technology	**9**
IV	Embedded Computing Basics, Embedded Hardware Unit. Embedded Platforms for Prototyping - Arduino, Intel Galileo, Intel Edison, Raspberry Pi, BeagleBone, mBed. Prototyping and Designing the Software for IoT Applications- Introduction, Prototyping Embedded DeviceSoftware- Programming using Arduino, Programming for an Arduino Controlled Traffic Control Lights at a Road Junction, Basic Arduino Programs to Blink LED, Find the Distance using Ultrasonic Sensor, Estimate Room Temperature, Measuring Soil Moisture Level	**10**
V	Business Models and Processes using IoT. Value Creation in the Internet of Things. Cloud PaaS- Xively, Nimbits, IBM Bluemix, CISCO IoT, AWS IoT, TCS Connected AWS Platform, Case studies- Smart Home, Smart Environment, Smart healthcare, Smart agriculture.	**8**

MARK DISTRIBUTION

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

CONTINOUS INTERNAL EVALUATION PATTERN

Attendance : 10 marks

Continuous Assessment Tests : 25 marks

Continuous Assessment Assignment : 15 marks

B.TECH(HONOURS)

CS1H30D	NETWORK SECURITY	Category	L	T	P	Credits	Year of Introduction
		VAC	3	1	0	4	2020

PRE-REQUISITE: CS1H20A- Number Theory and CS1H30A- Cryptographic Algorithms.

COURSE OVERVIEW: The purpose of this course is to create a better understanding of the network security concepts. This course covers network security standards, email security services, web security mechanisms, firewalls and wireless security mechanisms. This course helps the learner to gain insight into the key aspects of secure network communication and enables to apply in real-life scenarios.

COURSE OUTCOMES: After the completion of the course the student will be able to

CO1	Identify the key aspects of security, intrusion detection systems and digital signature schemes	Apply
CO2	Explain the security standards used in network communication	Understand
CO3	Identify the mechanisms in email security services.	Apply
CO4	Summarize the protocols used to provide web security	Understand
CO5	Explain the fundamental concepts of wireless network security and firewalls.	Understand

SYLLABUS

Introduction to network security-Malicious programs-Digital signatures-Kerberos v4-Cryptographic algorithms-Public Key Infrastructure (PKI)-Real-time communication security-Denial-of-Service protection-Internet Protocol Security (IPSec)-Internet Key Exchange (IKE) phases-Introduction to email security-Privacy Enhanced Mail (PEM)-Introduction to web security-IEEE 802.11 Wireless LAN -Wired Equivalent Privacy (WEP), Wi-Fi Protected Access (WPA)-Firewalls .

TEXT BOOKS

1. C. Kaufman, R. Perlman and M. Speciner, "Network Security: Private Communication in a Public World", 2/e, PHI.
2. William Stallings, "Cryptography and Network Security Principles and Practice", 5/e, Pearson Education Asia.

REFERENCE MATERIALS

1. Behrouz A. Forouzan, DebdeepMukhopadhyay, "Cryptography and Network Security", 3/e, Tata McGraw Hill.
2. Tyler Wrightson, "Wireless Network Security A Beginner's Guide", 2012, Tata McGraw Hill.
3. William Stallings, "Network Security Essentials: Applications and Standards", 4/e, Prentice Hall.
4. Schiller J., Mobile Communications, 2/e, Pearson Education.
5. Roberta Bragg et. al., "Network Security: The Complete Reference", Tata McGraw Hill.

COURSE PLAN

Module	Contents	No. of hours
I	Introduction to network security - Security requirements, Challenges of security, Network security model. Malicious programs – Worms, Viruses, Trojans, Spyware, Adware. Intrusion Detection Systems (IDS) - Uses, Techniques. Digital signatures - ElGamal, Schnorr, Digital Signature Standard (DSS).	8
II	Kerberos v4 – Configuration, Authentication, Encryption, Message formats. Kerberos v5 – Cryptographic algorithms, Message formats. Public Key Infrastructure (PKI) – Trust models, Revocation. Real-time communication security – Perfect Forward Secrecy (PFS), Denial-of-Service protection, Endpoint identifier hiding, Live partner reassurance. Internet Protocol Security (IPSec) - Authentication Header (AH), Encapsulating Security Payload (ESP), Internet Key Exchange (IKE) phases.	10

III	Introduction to email security - Security services for email, Establishing keys, Privacy, Authentication, Message integrity, Non-repudiation. Privacy Enhanced Mail (PEM) – Encryption, Source authentication and integrity protection, Message formats. Secure/Multipurpose Internet Mail Extensions (S/MIME) – Messages, Differences from PEM. Pretty Good Privacy (PGP) - Encoding, Certificate and key revocation, Anomalies, Object formats.	**10**
IV	Introduction to web security - Web security considerations, Threats. Secure Sockets Layer (SSL) – Architecture, Protocols, Transport Layer Security (TLS) – Differences from SSL. Hypertext Transfer Protocol Secure (HTTPS) – Connection initiation, Closure. Secure Shell (SSH) – Transport layer protocol, User authentication protocol, Connection protocol	**8**
V	IEEE 802.11 Wireless LAN - Network components, Architectural model, Services. IEEE 802.11i wireless LAN security - Services, Phases of operation. Wired Equivalent Privacy (WEP), Wi-Fi Protected Access (WPA), WPA2, Wireless Application Protocol (WAP) – Services, Protocol architecture. Firewalls – Need for firewalls, Packet filters, Circuit-level firewalls, Application layer firewalls.	**9**

MARK DISTRIBUTION

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

CONTINOUS INTERNAL EVALUATION PATTERN

Attendance : 10 marks

Continuous Assessment Tests : 25 marks

Continuous Assessment Assignment : 15 marks

CS1H30E	ADVANCED TOPICS IN MACHINE LEARNING	Category	L	T	P	Credit	Year of Introduction
		VAC	3	1	0	4	2020

PRE-REQUISITE : Nil

COURSE OVERVIEW: This course enables the learners to understand the advanced concepts and algorithms in machine learning. The course covers the standard and most popular supervised learning algorithms such as linear regression, logistic regression, decision trees, Bayesian learning and the naive Bayes algorithm, basic clustering algorithms, auto encoders, sampling methods and PAC learning. This course helps the students to provide machine learning based solutions to real world problems.

COURSE OUTCOMES:

CO1	Illustrate the concepts of regression and classification techniques	Apply
CO2	Demonstrate various unsupervised learning techniques	Apply
CO3	Choose suitable model parameters for different machine learning techniques and to evaluate a model performance	Apply
CO4	Explain the framework of PAC learning, basic concepts of VC dimension and non- uniform learnability.	Understand
CO5	Construct Bayesian models for data and apply computational techniques to draw inferences.	Apply
CO6	Illustrate the concepts of sampling algorithms, auto encoder, generative adversarial networks.	Apply

SYLLABUS

Overview of machine learning - supervised, semi-supervised, unsupervised learning, reinforcement learning Regression algorithms: least squares linear regression, gradient descent, closed form, normal equations, regularization techniques (LASSO, RIDGE), polynomial regression. Discriminative Methods - Logistic Regression, Decision Tree Learning. Generative Methods - Naive Bayes Classifier, Gaussian Discriminant Analysis (GDA). Clustering - Similarity measures, Hierarchical Agglomerative Clustering, K-means partitional clustering, K-medoids clustering, Gaussian mixture models. Classification Performance measures. Ensemble Methods. Sample complexity, computational complexity of training, Sample complexity for finite hypothesis spaces, PAC results for learning conjunctions, Sample complexity for infinite hypothesis spaces, Vapnik-Chervonenkis(VC) dimension. Graphical models, Markov random fields(MRFs), Inference on chains and factor graphs, inference on clique trees. Monte Carlo methods, Markov chain Monte Carlo(MCMC), Gibbs sampling. Variational methods. Auto Encoder, Variational Auto Encoder, Generative Adversarial Networks.

TEXT BOOKS

1. Christopher M. Bishop. Pattern recognition and machine learning. Springer 2006.
2. Ethem Alpaydin, Introduction to Machine Learning, 2nd edition, MIT Press 2010.
3. Mohammed J. Zaki and Wagner Meira, Data Mining and Analysis: Fundamental Concepts and Algorithms, Cambridge University Press, First South Asia edition, 2016.
4. Ian Good fellow, Yoshua Bengio and Aaron Courville. Deep Learning. MIT Press 2016.
5. Mehryar Mohri, Afshin Rostamizadeh and Ameet Talwalkar. Foundations of Machine Learning. Second edition. MIT Press 2018.
6. Tom Mitchell. Machine Learning. McGraw Hill 1997.
7. Richard O. Duda, Peter E . Hart, David G. Stork. Pattern classification, Second Edition. Wiley.
8. Jiawei Han, Micheline Kamber, Jian Pei. Data Mining Concepts and Techniques, Third Edition. Morgan Kaufmann.
9. David Foster. Generative Deep Learning - Teaching Machines to Paint, Write, Compose, and Play. O'Reilly Media, Inc., June 2019.

REFERENCE MATERIALS

1. Kevin P. Murphy. Machine Learning: A Probabilistic Perspective. MIT Press 2012
2. Carl Edward Rasmussen and Christopher K. I. Williams. Gaussian Processes for Machine Learning. MIT Press 2005.

COURSE PLAN

Module	Contents	No. of hours
I	Overview of machine learning - supervised, semi-supervised, unsupervised learning, reinforcement learning Regression algorithms: least squares linear regression, gradient descent, closed form, normal equations, regularization techniques (LASSO, RIDGE), polynomial regression. Discriminative Methods - Logistic Regression, Decision Tree Learning. Generative Methods - Naive Bayes Classifier, Gaussian Discriminant Analysis (GDA).	10
II	Clustering - Similarity measures, Hierarchical Agglomerative Clustering, K-means partitional clustering, K-medoids clustering, Gaussian mixture models: Expectation Maximization (EM) algorithm for Gaussian mixture model.	10
III	Classification Performance measures - Precision, Recall, Accuracy, F-Measure, ROC, AUC, generalisation and overfitting, cross-validation, bias-variance tradeoff, error estimation, parameter and model selection. Ensemble Methods - Bagging, Boosting, Adaboost, Random Forests	13
IV	Models of learnability- learning in the limit, probably approximately correct (PAC) learning. Sample complexity- quantifying the number of examples needed to PAC learn, Computational complexity of training, Sample complexity for finite hypothesis spaces, PAC results for learning conjunctions, Sample complexity for infinite hypothesis spaces, Vapnik-Chervonenkis.0(VC) dimension.	13
V	Graphical models - Bayesian belief networks, Markov random fields(MRFs), Inference on chains and factor graphs, inference on clique trees. Monte Carlo methods – Basic sampling algorithms, rejection sampling, importance sampling, Markov chain Monte Carlo(MCMC), Gibbs sampling. Variational methods. Auto Encoder, Variational Auto Encoder, Generative Adversarial Networks	14

MARK DISTRIBUTION

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

CONTINOUS INTERNAL EVALUATION PATTERN

Attendance	: 10 marks
Continuous Assessment Tests	: 25 marks
Continuous Assessment Assignment	: 15 marks

CS1H30F	THEORY OF COMPUTABILITY AND COMPLEXITY	Category	L	T	P	Credit	Year of Introduction
		VAC	3	1	0	4	2020

PRE-REQUISITE: CS1U20A Data Structures and CS1U30G Formal Languages and Automata Theory.

COURSE OVERVIEW: This is a theoretical course in computer science to enable the learners to know the fundamentals of computability and complexity theories. It covers the notions of computability/decidability, the process of reduction to prove decidability/undecidability and the classification of problems into class P, class NP and class NP Complete based on the time complexity of solving the problems. This course helps the learner to identify whether a real life problem is decidable/undecidable and also to classify a decidable problem into tractable or intractable, based on the time complexity class it belongs.

COURSE OUTCOMES:

CO1	Illustrate relative computing powers of Finite State Automata, Push Down Automata, Linear Bounded Automata and Turing Machines.	Apply
CO2	Prove that a given language is undecidable/not semi-decidable by using the reduction process.	Apply
CO3	Describe the time complexity of a given problem as a function of the number of steps required by a Turing machine to solve it.	Understand
CO4	Utilize polynomial time reduction to prove that a given problem is NP Complete.	Apply

SYLLABUS

Introduction to formal language theory, Undecidability, Overview of complexity classes, NP completeness and NP complete problems.

TEXT BOOKS

1. Dexter C. Kozen, Automata and Computability, Springer (1999)
2. Michael Sipser, Introduction to the Theory of Computation, Second Edition

REFERENCE MATERIALS

1. Douglas B. West, Introduction to Graph Theory, Second Edition

COURSE PLAN

Module	Contents	No. of hours
I	Finite State Automata, Push Down Automata, Linear Bounded Automata, Turing Machines, Recursive Languages, Recursively Enumerable Languages, Universal Turing Machine, Enumeration Machine, Two Counter Machine.	14
II	Halting Problem, Language representation of a problem, Reduction - applications, Rice's First and Second Theorem with proof.	14
III	Measuring time complexity, Asymptotic notations - Big O and small-o, Analysing algorithms, Complexity relationship among models. Complexity classes- Class P, example problems in class P, Class NP, Polynomial time verification, example problems in class NP.	12
IV	Satisfiability problem, Polynomial time reducibility, Overview of Graphs, NP Complete Problems, Cook-Levin theorem (SAT is NP Complete).	10
V	CLIQUE, Vertex Cover and Hamiltonian path with proof of correctness of NP Completeness	10

MARK DISTRIBUTION

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

CONTINOUS INTERNAL EVALUATION PATTERN

Attendance : 10 marks

Continuous Assessment Tests : 25 marks

Continuous Assessment Assignment : 15 marks