MS in Computer Sciences

- Title of program that will appear on the degree:

Master of Science in Computer Sciences

Fall Semester 2022

SUBJECT	Credits
CS701 Advanced Analysis of Algorithms	3
CS702 Advanced Computer Architecture	3
CS704 Theory of Automata – II	3
CS705 Advanced Operating Systems	3
Total	**12**

Scheme of studies: The scheme of study was designed according to the guidelines of HEC Computer Science Curriculum-2017 (which includes guidelines for both MS and PhD programs) and HEC NCEAC Artificial Intelligence and Data Science Curriculum March-2020. MS program comprised of four semesters. Semester wise breakup of courses is given bellow. MS program comprised of four semesters. Semester wise breakup of courses is given below.

Semester I

SUBJECT	Credits
CS 7xx Core Course – I	3
CS 7xx Core Course – II	3
CS 7xx Core Course – III	3
Total	**9**

Semester II

SUBJECT	Credits
CS 7xx Core Course – IV	3
CS 7xx Elective Course – I	3
CS 7xx Elective Course – II	3
SS 7xx Research Methodology	**1**
Total	**10**

Semester III

SUBJECT	**Credits**
CS 7xx Elective Course – III	3
CS 7xx MS – Thesis-I	**3**
Total	**6**

Semester IV

SUBJECT	**Credits**
CS 7xx Elective Course – IV	3
CS 7xx MS – Thesis-II	**3**
Total	**6**

List of Core Courses for MS program: At least four courses must be taken from the following

- CS701 Advanced Analysis of Algorithms
- CS705 Advanced Operating Systems
- CS703 Theory of Programming Languages
- CS704 Theory of Automata – II
- CS702 Advanced Computer Architecture

List of Elective Courses for MS Program

Course Code	**Title**	**Credits**
CS709	Internet of Things	3
	Cloud Computing	3
CS708	Software Testing and quality Assurance	3
	Advance Software Requirement Engineering	3
	Digital Image Processing	3
	Artificial Intelligence	3
	Artificial Neural Network	3
CS707	Machine Learning	3

	Computer Vision	3
	Natural Language Processing	3
	Data Mining	3
CS706	Deep Learning	3
	Programming for Artificial Intelligence	3
CS710	Research Methodology	1
CS711	Advance Cryptography	3

Course Outlines

CS501 Advanced Analysis of Algorithms…………………….……………………………………(3 credits)

Objectives:

The major objective of this course is providing comprehensive knowledge of modern computer algorithms and solving scientific and engineering problems efficiently and accurately. The students will be guided, how to analyze complex algorithms comparing efficiencies of these algorithms. Students will not only be taught the design of the existing algorithms but on the other hand it will be focused to teach them designing techniques using rigorous mathematical approaches. The students will be motivated to think about procedures solving real world problems optimally and correctly. Real world problem will be taken as examples to create feelings about the usefulness of this course.

Learning Outcomes:

Upon successful completion of this course, students should be able to:

Argue and prove the correctness of algorithms using rigorous mathematical techniques taught in this course. Analyze average and worst-case running times of given algorithm. Describe the divide-and-conquer technique and arguing when an algorithmic design calls this approach. Derive and solve recurrence relations describing performance of divide-and-conquer algorithms. Describe advanced topics such as dynamic programming and greedy approach and reason to use these approaches for a particular situation. Integrate dynamic programming and recursive approach improving efficiency of an algorithm. Know the importance of graph theory in problem solving. Employing graphs to model science and engineering problems, and to reason about when it is appropriate to use it optimally. Analyze and design algorithms on further advanced topics such as computational geometry, operations research, cryptography, number theoretic, algorithms etc. Analyze several other algorithms of importance such as string matching, NP completeness, approximation algorithms etc.

Course content:

Introduction, Underlying mathematical theory, Induction and recursion techniques in analyzing algorithms, Asymptotic notations, Search techniques, Divide-and conquer technique, Randomized algorithms, Heuristic algorithms, Brute Force approach, Backtracking, branch-and-bound, Optimization techniques in algorithms designing, Dynamic algorithms, Greedy algorithms, Graph Theory, Searching algorithms, Minimal spanning tree algorithms, Polynomials and FFT, Number theoretic notations, Number theoretic algorithms, RSA cryptosystems, String matching, pattern matching, NP completeness and NP completeness proofs.

Suggested readings:

- "Introduction to Algorithms" by Cormen, Leiserson, Rivest and Stein, Second Edition, 2001. Electronic copy available in IC.
- "Randomized Algorithms" by Motwani and Raghavan and "Approximation Algorithms" by Vazirani.

CS505 Advanced Operating Systems …………………………………………………………(3 credits)

Objectives: The objectives of the course are to explore the design principles of computer operating systems and to emphasize on implementation details of various sub-modules of operating systems. The course will focus on the engineering and performance trade-offs in the design of operating systems. The purpose will be to teach not only what operating systems are and how they work today, but also why they are designed the way they are and how they are likely to evolve in the future. The emphasize will be on the practical aspects of the topics through the case study of Linux kernel as an example of a commercial operating system. The course will cover programming aspects through some programming assignments. It will also introduce the state-of-the-art OS research topics through a few research papers reading assignments.

Learning Outcomes:

Understand in detail how and why different parts of an operating system work.

Understand the engineering tradeoffs involved in the design of various sub-modules of an operating system. Understand how operating systems are structured, what are alternative OS architectures and how different modules interact together to form a cohesive and complex system. Write non-trivial programs in C or C++ that would invoke OS services via system calls in an efficient manner. Understand various contemporary research issues in operating systems e.g. security and protection, efficient memory management, I/O buffering, device handling, faster and more efficient file systems and OS architectures.

Course content: Introduction, Operating System: top-down and bottom up views, design issues, modules and components, services and system calls, structure and organization. Executable file formats, Static Linking, Dynamic linking, Loading programs into memory, Anatomy of a Process, Exceptional control flow, Context switching, Process related system calls, Threads, Concurrency and synchronization: Race conditions, Mutual exclusion and problems arising from them, Locks, semaphores and condition variables, Design of Inter-process communication mechanisms, Readers/writers and producer/consumer problems, Re-entrant code and thread-safe libraries, Deadlock, Design alternatives for highly concurrent servers (research topic). CPU scheduling: Scheduling policies and algorithms, Scheduling algorithm comparison, Real-time and multi-processor scheduling, Linux case study. Dynamic memory management: Internal design alternatives for malloc and free routines, Garbage collection. OS memory management: Memory protection, Program relocation, Memory partitioning techniques, Virtual memory, Paging and segmentation, TLB and cache management. File systems: Naming issues, Design alternatives for file systems, Example file systems and their comparison. I/O management: Memory mapped Vs Direct I/O, Interrupt driven Vs Polled I/O, Device controllers and device drivers, Naming issues, OS I/O architecture, Buffering techniques, Disk devices and their management. Timer management in OS, Event notification mechanisms, UNIX signals, Security and protection: Security aspects in operating systems, Authentication, Authorization, Accounting, Security domains and security models, Protection against worms and viruses, Protection against buffer overflow attacks, Windows case study, Linux case study, Java Security, State of the art OS research topics.

Suggested readings:

- "Modern Operating Systems", Tanenbaum (2001), Author: Andrew S. Tanenbaum, Edition: 2nd Edition, ISBN: 978-0136006633, Publisher: Prentice Hall
- Book Title: Operating Systems Internals and Design Principles, Stallings (2004), Author: William Stallings, Edition: 5th edition, ISBN: 978-0131479548, Publisher: Prentice Hall, URL: http://williamstallings.com/OS/OS5e.html

CS507 Theory of Programming Languages ……………………………………(3 credits)

Objectives: Thousands of different programming languages have been designed by different people over the last 60 years. The major content of this is different programming languages paradigm and implementation issues. Some objectives of this course are:

1. To study programming language constructs and features.
2. To become familiar with language design principles.
3. To become familiar with the different semantic issues of programming languages.
4. To become competent in problem solving using different language paradigms.
5. To become proficient in implementing a lexical analyzer for a programming language based on its specification.
6. To become familiar with the principal methods of formal semantics: operational, denotational, and axiomatic.
7. To experience a diverse range of programming languages, constructs, and implementation issues.
8. To introduce students to programming language theory including predicate calculus, the lambda calculus, and semantic algebras.

Learning Outcomes: Students will be knowledgeable of a wide range of programming language constructs and features. Students will have an appreciation of the implications of language design principles. Students will have experience problem solving using different language paradigms. Students will be familiar with the principal methods of formal semantics. Students will have experience in learning programming languages in a relatively short period of time.

Course content: Introduction, Evolution of the Major Programming Languages: Syntax and Semantics, Lexical, Syntax Analyses, Names Binding, Type Checking and Scopes, Data Types and abstract Data Types, Expressions, Assignment Statement, and Control Structures , Subprograms and their implementation, Exception Handling and Event Handling Functional, Logic and Object Oriented Programming Languages, Concurrency, Programming Paradigms and language extensions.

Suggested readings:

- Concepts of Programming Languages, Robert W. Sebesta, 8th ed, Addison-Wesley Higher Education, 2008, ISBN-10: 0-321-49362-1 ISBN-13: 978-0-321-49362-0
- Programming Languages: Paradigm & Practice, Appleby, VandeKopple, 2th Edition, McGraw-Hill
- Programming Languages Concepts, Carlo Ghezzi and Mehdi Jazayeri, 3rd ed, John Wiley & Sons

CS534 Theory of Automata – II ……………………………………………(3 credits)

Objectives: This is the first course in theoretical computer science. As opposed to other courses in computer science, the purpose of this course is to ask very fundamental questions about computations. The main questions addressed in this course are:

1. What is a computation?
2. Is there a universal model of computation?
3. Can every problem be computed?
4. Can we identify problems that are not computable?
5. How much resources are needed to solve a problem?
6. Can we classify problems that can be solved in principle (given a lot of resources) but cannot be solved in practice?

Learning Outcomes: Learn about the introduction and Review of Proof Techniques, learn about the basic Computational Model, Learn about the NP and NP Completeness, Student can also get knowledge of, Space Complexity. Understand the Polynomial Time Hierarchy, Randomized Computation, Interactive Proofs, The PCP Theorem and Inapproximability

Course Content: Introduction and Review of Proof Techniques, Basic Computational Model, Turing Machines and Universal Turing Machines, Undecidability and Halting problem, NP and NP Completeness, The class P, The classes NP and coNP, The Cook-Levin Theorem, Reductions of problems in NP, Space Complexity, PSPACE and NL completeness, Savitch's Theorem, Polynomial Time Hierarchy, Randomized Computation, Interactive Proofs, The PCP Theorem and Inapproximability

Suggested readings:

- Michael Sipser, Introduction to the Theory of Computation (2nd Edition, Thomson Publishing, 2006)
- John Martin, Introduction to Languages and the Theory of Computation (3rd Edition, McGraw-Hill, 2002)

EE502 Advanced Computer Architecture ……………………………………….(3 credits)

Objective: This course will provide the students with an understanding of the various levels of studying computer architecture, with emphasis on instruction set level and register transfer level. This course prepares the students to use basic combinational and sequential building blocks to design larger structures like Arithmetic Logic Units, memory subsystems, I/O subsystems etc.

Learning Outcomes: At the end of the course, you should be able to:

Understand Instruction Set Architecture design and Central Processing Units of the RISC (Reduced Instruction Set Computers) and the CISC (Complex Instruction Set Computers) type. Describe the behavior and structure of a computer using RTL (Register transfer language)

1. Explain Pipelining and instruction level Parallelism
2. Explain the I/O sub systems
3. Understand Magnetic disk drives
4. Explain the memory module of computer
5. Understand Number Systems and Radix Conversion

Course Content: Distinction between Computer Architecture, Organization and design, Levels of abstraction in digital design ,Perspectives of different people about computers, General operation of a stored program digital computer, The Fetch – Execute process, Concept of an ISA ,A taxonomy of computers and their instructions, Instruction set features, Addressing Modes, RISC and CISC architectures, Measures of performance, Introduction to the ISA and instruction formats, Coding examples and Hand assembly, Using Behavioral RTL to describe the SRC, Implementing Register Transfers using Digital Logic Circuits, Introduction to the ISA of the FALCON – A, FALCON-E, EAGLE and Modified EAGLE, The Design Process, A Uni-Bus implementation for the SRC, Structural RTL for the SRC instructions, Logic Design for the 1-Bus SRC, The Control Unit, The 2-and 3-Bus Processor Designs. The Machine Reset, Machine Exceptions, Pipelining, Microprogramming, I/O interface design, Programmed I/O Interrupt driven I/O, Direct memory access (DMA),Addition, subtraction, multiplication & division for integer unit, Floating point unit, Memory organization and design, Memory hierarchy, Cache memories, Virtual memory.

Suggested readings:

- Computer Systems Design and Architecture, Heuring, Jordan (2003), Author: Vincent P. Heuring & Harry F. Jordan, Edition: 2nd, ISBN: 0130484407, Publisher: Prentice-Hall
- Book Store: http://www.amazon.com/Computer-Systems-Design-Architecture-2nd/dp/0130484407/ref=dp_ob_title_bk

Internet of Things……………………………………………………………….(3 credits)

Objectives: The explosive growth of the “Internet of Things” is changing our world and the rapid drop in price for typical IoT components is allowing people to innovate new designs and products at home. In this first class in the specialization you will learn the importance of IoT in society, the current components of typical IoT devices and trends for the future. IoT design considerations, constraints and interfacing between the physical world and your device will also be covered. You will also learn how to make design trade-offs between hardware and software. We'll also cover key components of networking to ensure that students understand how to connect their device to the Internet.

Learning Outcomes: On completion of the course, the student should be able to:

1. Describe what IoT is and how it works today
2. Recognize the factors that contributed to the emergence of IoT
3. Design and program IoT devices
4. Use real IoT protocols for communication
5. Secure the elements of an IoT device
6. Design an IoT device to work with a Cloud Computing infrastructure.
7. Transfer IoT data to the cloud and in between cloud providers
8. Define the infrastructure for supporting IoT deployments

Course Content: The course covers the following areas:

- Introduction to IoT: Sensing, Actuation, Networking basics, Communication Protocols, Sensor Networks, Machine-to-Machine Communications, IoT Definition, Characteristics. IoT Functional Blocks, Physical design of IoT, Logical design of IoT, Communication models & APIs.
- M2M to IoT-The Vision-Introduction, From M2M to IoT, M2M towards IoT-the global context, A use case example, Differing Characteristics. Definitions, M2M Value Chains, IoT Value Chains, An emerging industrial structure for IoT
- M2M vs IoT An Architectural Overview–Building architecture, Main design principles and needed capabilities, An IoT architecture outline, standards considerations. Reference Architecture and Reference Model of IoT.
- IoT Reference Architecture- Getting Familiar with IoT Architecture, Various architectural views of IoT such as Functional, Information, Operational and Deployment. Constraints affecting design in IoT world- Introduction, Technical design Constraints.
- Domain specific applications of IoT: Home automation, Industry applications, Surveillance applications, Other IoT application.

- Developing IoT solutions: Introduction to Python, Introduction to different IoT tools, Introduction to Arduino and Raspberry Pi Implementation of IoT with Arduino and Raspberry, Cloud Computing, Fog Computing, Connected Vehicles, Data Aggregation for the IoT in Smart Cities, Privacy and Security Issues in IoT.

Suggested Readings:

- Computer networking: a top-down approach, Author: Kurose, James F.; Ross, Keith W. Edition: 5th ed., international ed.: Boston, Mass.: Pearson, cop. 2010

Cloud Computing ……………………………………….……………………………(3 credits)

Objective: This course provides a hands-on comprehensive study of Cloud concepts and capabilities across the various Cloud service models including Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS), and Business Process as a Service (BPaaS). IaaS topics start with a detailed study the evolution of infrastructure migration approaches from VMWare/Xen/KVM virtualization, to adaptive virtualization, and Cloud Computing / on-demand resources provisioning. Mainstream Cloud infrastructure services and related vendor solutions are also covered in detail. PaaS topics cover a broad range of Cloud vendor platforms including AWS, Google App Engine, Microsoft Azure, Eucalyptus, OpenStack and others as well as a detailed study of related platform services such as storage services that leverage Google Storage, Amazon S3, Amazon Dynamo, or other services meant to provide Cloud resources management and monitoring capabilities. The SaaS and PaaS topics covered in the course will familiarize students with the use of vendor-maintained applications and processes available on the Cloud on a metered on-demand basis in multi-tenant environments. The course also covers the Cloud security model and associated challenges and delves into the implementation and support of High-Performance Computing and Big Data support capabilities on the Cloud. Through hands-on assignments and projects, students will learn how to configure and program IaaS services. They will also learn how to develop Cloud-based software applications on top of various Cloud platforms, how to integrate application-level services built on heterogeneous Cloud platforms, and how to leverage SaaS and BPaaS solutions to build comprehensive end-to-end business solutions on the Cloud.

Learning Outcomes: On completion of the course, the student should be able to:

1. Use public and private cloud solutions for computational science and engineering applications.
2. Discuss key concepts of cloud computing services, such as Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS);
3. Asses the suitability of cloud computing infrastructures for different scientific applications.
4. Implement software for cloud-based distributed computing using the technology presented in the course.
5. Critically analyses and present solutions and implementations in writing and orally.

Course Content: Introduction to Cloud Computing, Virtualization, Cloud computing Services, Cloud Computing and Business Value, Open Source Cloud Implementation and Administration, Cloud Deployment Techniques, Security, Architecture for Cloud Application, Cloud Programming, Adoption and Use of Cloud, Risks of Cloud Computing and Related Costs, AAA Administration for Clouds, Security as a service, Mobile Cloud Computing

Suggested Readings:

- "Cloud Computing: Concepts, Technology & Architecture" by Thomas Erl
- "Cloud Computing for Programmers" by Daniele Casal
- "Cloud Computing: From Beginning to End" by Mr Ray J Rafaels

Software testing and quality assurance……………………………..…………(3 credits)

Objectives: The aim and objective of this course is to teach students the concepts and skills needed for SQA and Testing. Software quality assurance (SQA or simply QA) is viewed as an activity that runs through the entire development process. It encompasses activities and related techniques to ensure the implementation of appropriate functionality that satisfy the requirements/needs of its targeted client/users for the intended software system, product, or service both correctly and efficiently.

Learning Outcomes: Students can understand quality management, review techniques and about software testing strategies for different systems.

Course Content: Introduction about Quality Management, Review Techniques, Software quality assurance, Software testing strategies, Testing strategies for Conventional Software's, System testing, testing conventional applications, Testing Object Oriented applications, Testing Web Applications, Formal modeling and verification.

Suggested Reading:

- Software Engineering - A Practitioner's Approach (7th edition) by Roger S. Pressman. ISBN 13. 9780073375977.
- Software Testing and Quality Assurance: Theory and Practice by Kshirasagar Naik, Priyadarshi

Digital Image Processing ……………………..……………………………….(3 credits)

Objectives: To introduce basic concepts and methodologies for the formation, representation, compression, enhancement and analysis of digital images. To provide a foundation for developing applications and for further study in the field. To gain practical experience in the design and implementation of image processing algorithms.

Learning Outcomes: Students can introduction to basic concepts and methodologies for the formation, representation, compression, enhancement and analysis of digital images. It can provide a foundation for developing applications and for further study in the field. Students can gain practical experience in the design and implementation of image processing algorithms.

Course Content: Introduction of Applications of digital image processing and Elements of digital image processing system, Digital Image Fundamentals, Intensity Transformation and Spatial Filtering, Filtering in the frequency domain, Image Restoration and Reconstruction, Wavelets and multiresolution processing, Image Compression, Image Segmentation, Image representation and description, Color image processing, Morphological image processing

Suggested Readings:

- **“**Digital Image Processing” by Rafael C Gonzalez and Richard E Woods
- “Image Processing, Analysis and Machine Vision” by Milan Sonka and Vaclav Hlavac and Roger Boyle

Course Name: **Artificial Intelligence**
Credit Hours: 2-1 **Contact Hours:** 2-3
Pre-requisites: Object Oriented Programming

Course Introduction: Artificial Intelligence has emerged as one of the most significant and promising areas of computing. This course focuses on the foundations of AI and its basic techniques like Symbolic manipulations, Pattern Matching, Knowledge Representation, Decision Making and Appreciating the differences between Knowledge, Data and Code. AI programming language Lisp has been proposed for the practical work of this course.

CLO No.	Course Learning Outcomes	Bloom Taxonomy
CLO-1	Understand the fundamental constructs of Lisp programming language.	C2 (Understand)
CLO-2	Understand key concepts in the field of artificial intelligence	C2 (Understand)
CLO-3	Implement artificial intelligence techniques and case studies	C3 (Apply)

Course Content: Introduction (Introduction, basic component of AI, Identifying AI systems, branches of AI, etc.); Reasoning and Knowledge Representation (Introduction to Reasoning and Knowledge Representation, Propositional Logic, First order Logic); Problem Solving by Searching (Informed searching, Uninformed searching, Local searching.); Constraint Satisfaction Problems; Adversarial Search (Min-max algorithm, Alpha beta pruning, Game-playing); Learning (Unsupervised learning, Supervised learning, Reinforcement learning) ;Uncertainty handling (Uncertainty in AI, Fuzzy logic); Recent trends in AI and applications of AI algorithms (trends, Case study of AI systems, Analysis of AI systems)

Suggested Readings:

- Stuart Russell and Peter Norvig, Artificial Intelligence. A Modern Approach, 3rd edition, Prentice Hall, Inc., 2010.
- Hart, P.E., Stork, D.G. and Duda, R.O., 2001. Pattern classification. John Willey & Sons.
- Luger, G.F. and Stubblefield, W.A., 2009. AI algorithms, data structures, and idioms in Prolog, Lisp, and Java. Pearson Addison-Wesley

Course Name:	**Artificial Neural Networks**
Credit Hours: 2-1	**Contact Hours:** 2-3
Pre-requisites:	Programming for Artificial Intelligence

Course Introduction: This course will introduce Artificial Neural Networks, their basic architecture and how they mimic the human brain using simple mathematical models. Many of the important concepts and techniques around brain computing and the major types of ANN will also be introduced. Emphasis is made on the mathematical models, understanding learning laws, selecting activation functions and how to train the networks to solve classification problems. Students would be able to understand and use different types of neural networks and would be able to use different activation functions and construct layered networks to solve classification problems.

CLO No.	**Course Learning Outcomes**	**Bloom Taxonomy**
CLO-1	Understand the fundamentals of neural networks in AI	C2 (Understand)
CLO-2	Explain how simple ANNs can be designed	C2 (Understand)
CLO-3	Apply ANN for classification Problems	C3 (Apply)
CLO-4	Differentiate between different Networks and their learning laws	C4 (Analyze)

Course Outline: Introduction and history of neural networks, Basic architecture of neural networks, Perceptron and Adaline (Minimum Error Learning) for classification, Gradient descent (Delta) rule, Hebbian, Neo-Hebbian and Differential Hebbian Learning, Drive Reinforcement Theory, Kohonen Self Organizing Maps, Associative memory, Bi-directional associative memory (BAM), Energy surfaces, The Boltzmann machines, Backpropagation Networks, Feedforward Networks; Introduction to Deep learning and its architecture.

Reference Materials:

- Neural Network Design, 2nd Edition, Martin T. Hagan, Howard, B. Demuth, Mark Hudson Beale and Orlando De Jesus, Publisher: Martin Hagan; 2 edition (September 1, 2014), ISBN-10: 0971732116
- An Introduction to Neural Networks, James A Anderson, Publisher: A Bradford Book (March 16, 1995), ISBN-10: 0262011441
- Fundamentals of Artificial Neural Networks, Mohammad Hassoun, Publisher: A Bradford Book (January 1, 2003), ISBN-10: 0262514672

Course Name:	**Machine Learning**
Credit Hours: 2-1	**Contact Hours:** 2-3
Pre-requisites:	Programming for Artificial Intelligence

Course Introduction: Machine learning is one of the fastest growing areas of computer science, with far-reaching applications. The aim of this course is to: a) Present the basic machine learning concepts; b) Present a range of machine learning algorithms along with their strengths and weaknesses; c) Apply machine learning algorithms to solve problems of moderate complexity.

CLO No.	Course Learning Outcomes	Bloom Taxonomy
CLO-1	Describe basic machine learning concepts, theories and applications.	C1 (Knowledge)
CLO-2	Apply supervised learning techniques to solve classification problems of moderate complexity.	C3 (Apply)
CLO-3	Apply unsupervised learning techniques to solve clustering problems of moderate complexity	C3 (Apply)
CLO-4	Apply reinforcement learning algorithms to environments with complex dynamics.	C3 (Apply)
CLO-5	Develop a reasonable size project using suitable machine learning technique	C6 (Create)

Course Outline: Introduction to machine learning; concept learning: General-to-specific ordering of hypotheses, Version spaces Algorithm, Candidate elimination algorithm; Supervised Learning: decision trees, Naive Bayes, Artificial Neural Networks, Support Vector Machines, Overfitting, noisy data, and pruning, Measuring Classifier Accuracy; Linear and Logistic regression; Unsupervised Learning: Hierarchical Agglomerative Clustering. k-means partitional clustering; Self-Organizing Maps (SOM) k-Nearest-neighbour algorithm; Semi supervised learning with EM using labelled and unlabelled data; Reinforcement Learning: Hidden Markov models, Monte Carlo inference Exploration vs. Exploitation Trade-off, Markov Decision Processes; Ensemble Learning: Using committees of multiple hypotheses. Bagging, boosting.

Reference Materials:

1. Machine Learning, Tom, M., McGraw Hill, 1997.

2. Machine Learning: A Probabilistic Perspective, Kevin P. Murphy, MIT Press,2012

Course Name:	**Computer Vision**
Credit Hours: 2-1	**Contact Hours:** 2-3
Pre-requisites:	Artificial Neural Networks

Course Introduction: With a single glance a human interprets the entire scene. How many objects are present in the scene and where they are located. Which person is present in the scene? What will happen next. However, computers lack this capability. We have seen only face detectors so far working in our mobile phones. What is the challenge in understanding the 3D scene, i.e., the identity, the location and the size of the objects present in the scene? In this course we will introduce the basic concepts related to 3D scene modelling from single view and multiple views.

CLO No.	**Course Learning Outcomes**	**Bloom Taxonomy**
CLO-1	Understanding the single view geometry concepts	C2 (Understand)
CLO-2	Understanding the multiple view geometry concepts	C2 (Understand)
CLO-3	Apply concepts of CV for solving real world problems	C3 (Apply)

Course Outline: Introduction to Computer Vision (Problems faced, History and Modern Advancements). Image Processing, Image filtering, Image pyramids and Fourier transform, Hough transform. Camera models, setting up a camera model from parameters, Camera looking at a plane, Relationship of plane and horizon line, Rotation about camera canter. Concatenation, Decomposition and Estimation of transformation from point correspondences, Points and planes in 2D/3D, Transformations in 2D/3D, Rotations in 2D/3D. Edge detection, corner detection. Feature descriptors and matching (HoG features, SIFT, SURF). Applications of Computer Vision Traditional Methods: Image Stitching: Making a bigger picture from smaller pictures Single View Geometry: Converting a single image into a 3D model. Applications of CV using Deep Learning: Image Detection (Localization, Historical Techniques, RCNN, FRCNN, YOLO, Retina), Image Segmentation (UNet, SegNet, MaskRCNN), Image Generation (GANN)

Reference Materials: Textbook:

1. Computer Vision: Algorithms and Applications, by Richard Szeliski. Reference Book:

2. Multiple View Geometry in Computer Vision, by Richard Hartley and Andrew Zisserman.

3. Computer Vision: A Modern Approach, by David Forsyth and Jean Ponce. 4. Digital Image Processing, by Rafael Gonzalez and Richard Woods.

Course Name:	**Natural Language Processing**
Credit Hours: 2-1	**Contact Hours:** 2-3
Pre-requisites:	Artificial Neural Networks

Course Introduction: Natural Language Processing (NLP) is the application of computational techniques to the analysis and synthesis of natural language and speech. This course is an introduction to NLP with prior programming experience in Python.

CLO No.	Course Learning Outcomes	Bloom Taxonomy
CLO-1	Understand techniques for information retrieval, language translation, and text classification.	C2 (Understand)
CLO-2	Understand the advantages of using standard corpora. Identify examples of current corpora for a variety of NLP tasks.	C2 (Understand)
CLO-3	Understand and contrast deterministic and stochastic grammars, providing examples to show the adequacy of each	C2 (Understand)
CLO-4	Solve classic and stochastic algorithms for parsing natural language.	C3 (Apply)

Course Outline: Introduction & History of NLP, Parsing algorithms, Basic Text Processing, Minimum Edit Distance, Language Modelling, Spelling Correction, Text Classification, Deterministic and stochastic grammars, CFGs, Representing meaning /Semantics, Semantic roles, Semantics and Vector models, Sentiment Analysis, Temporal representations, Corpus-based methods, N-grams and HMMs, Smoothing and back off, POS tagging and morphology, Information retrieval, Vector space model, Precision and recall, Information extraction, Relation Extraction (dependency, constituency grammar), Language translation, Text classification, categorization, Bag of words model, Question and Answering, Text Summarization

Reference Materials:

1. Daniel Jurafsky and James H. Martin. 2018. Speech and Language Processing: An Introduction to Natural Language Processing, Third Edition. Prentice Hall

2. Foundations of Statistical Natural Language Processing, Manning and Schütze, MIT Press. Cambridge, MA: May 1999

Course Name: **Data Mining**

Credit Hours: 2-1 **Contact Hours:** 2-3

Pre-requisites: Statistics

Course Introduction: Data Mining has emerged at the confluence of artificial intelligence, statistics, and databases as a technique for automatically discovering hidden patterns in large datasets. The main purpose of this course is the ability to analyze and construct knowledge from data. The aims of this course are to:

- Expand on the student's understanding and awareness of the concepts of data mining basics, techniques, and application.
- Introduce the concepts of Data Pre-processing and Summary Statistics.
- Introduce the concepts of Frequent Item Set Generation, Associations and Correlations measures.
- Introduce the concepts of Classification, Prediction, and Clustering algorithms.

Build on the programming and problem-solving skills developed in previous subjects studied by the student, to achieve an understanding of the development of Classification, Prediction, and Clustering applications.

CLO No.	Course Learning Outcomes	Bloom Taxonomy
CLO-1	Apply pre-processing techniques on any given raw data.	C3 (Apply)
CLO-2	Select and apply proper data mining algorithm to discover interesting patterns	C3 (Apply)
CLO-3	Analyze and extract patterns to solve problems and point out how to deploy solution	C4 (Analyze)
CLO-4	Evaluate systematically supervised, semi supervised and unsupervised models and algorithms with respect to their accuracy	C4 (Analyze)

Course Outline: Introduction to data mining and basic concepts, Pre-Processing Techniques & Summary Statistics, Association Rule mining using A priori Algorithm and Frequent Pattern Trees, Introduction to Classification Types, Supervised Classification (Decision trees, Naïve Bae Classification, K-Nearest Neighbours, Support Vector Machines etc.), Unsupervised Classification (K Means, K Median, Hieratical and Divisive Clustering, Kohonan Self Organizing maps), outlier & anomaly detection, Web and Social Network Mining, Data Mining Trends and Research Frontiers. Implementing concepts using Python

Reference Materials:

1. Jiawei Han & Micheline Kamber, Jian Pei (2011). Data Mining: Concepts and Techniques, 3rd Edition.

2. Pang-Ning Tan, Michael Steinbach, and Vipin Kumar (2005). Introduction to Data Mining.

3. Charu C. Aggarwal (2015). Data Mining: The Textbook 4. D. Hand, H. Mannila, P. Smyth (2001). Principles of Data Mining. MIT Press.

Course Name: **Deep Learning**

Credit Hours: 2-1 **Contact Hours:** 2-3

Pre-requisites: Artificial Neural Networks

Course Introduction: Deep neural networks have achieved state of the art performance on several compute vision and speech recognition benchmarks. Deep learning algorithms extract layered high and low-level features from raw data. With increasing non-line hidden layers, the discriminative power of the network improves. This course builds on the fundamentals of Neural networks and artificial intelligence and covers advanced topics in neural networks, convolutional and recurrent network structures, deep unsupervised and reinforcement learning. It also embeds applications of these algorithms to several real-world problem in computer vision, speech recognition, natural language processing, game theory, etc.

CLO No.	Course Learning Outcomes	Bloom Taxonomy
CLO-1	Apply deep learning algorithms to real-world problems	C3 (Apply)
CLO-2	Analyze results from deep learning to select appropriate solutions	C4 (Analyze)
CLO-3	Code the novel neural network architectures from scratch and evaluating the performance on application specific standard benchmarks	C3 (Apply)

Course Outline: Basics of deep learning, learning networks, Shallow vs. Deep learning etc.; Machine learning theory – training and test sets, evaluation, etc. Theory of Generalization; Multi-layer perceptron's, error back-propagation; Deep convolutional networks, Computational complexity of feed forward and deep convolutional neural networks; Unsupervised deep learning including auto-encoders; Deep belief networks; Restricted Boltzman Machines; Deep Recurrent Neural Networks (BPTT, LSTM, etc.); GPU programming for deep learning CuDNN; Generative adversarial networks (GANs); Sparse coding and auto-encoders; Data augmentation, elastic distortions, data normalization; Mitigating overfitting with dropout, batch normalization, dropconnect; Novel architectures, ResNet, GoogleNet, etc

Reference Materials:

1. Deep Learning by Ian Goodfellow, Yoshua Bengio, Aaron Courville (

http://www.deeplearningbook.org/)

2. Deep learning with python by Francoise Chollet, ISBN-10: 9781617294433, 2017

Course Name:	**Programming for Artificial Intelligence**
Credit Hours: 2-1	**Contact Hours:** 2-3
Pre-requisites:	Artificial Intelligence

Course Introduction: This course aims to introduce standard programming practices and to help develop programming skills necessary for designing and implementing Artificial Intelligence systems. The course introduces a modern state of the art programming language for Artificial Intelligence and builds up the necessary programming background for the main courses like Knowledge Representation & Reasoning, Machine Learning, Artificial Neural Networks, and Natural Language Processing. This course will help the students of Artificial Intelligence develop the programming acumen and style. The ultimate aim of this course is to help students in using the programming language to solve problems of interest to them.

CLO No.	Course Learning Outcomes	Bloom Taxonomy
CLO-1	Comprehend the fundamental constructs of programming language for data analysis and representation.	C2 (Understand)
CLO-2	Understand and apply the Object-oriented concepts in the programming language	C2 (Understand)
CLO-3	Solve and analyze programming and data analysis problems using standard libraries and/or toolboxes of the programming language.	C4 (Analyze)

Course Outline: Introduction to Programming language (Python): The first objective of the course is to introduce and then build the proficiency of students in the programming language. The basics include IDE for the language (e.g., Jupyter Notebook or IPython), variables, expressions, operands and operators, loops, control structures, debugging, error messages, functions, strings, lists, object-oriented constructs and basic graphics in the language. Special emphasis is given to writing production quality clean code in the programming language using version control (git and subversion). Introducing libraries/toolboxes necessary for data analysis: The course should introduce some libraries necessary for interpreting, analyzing and plotting numerical data (e.g., NumPy, MatPlotLib, Anaconda and Pandas for Python) and give examples of each library using simple use cases and small case studies.

Reference Materials: Textbook:

1. Severance, C.R., 2016. “Python for everybody: Exploring data using Python 3.” CreateSpace Independent Publ Platform.