CSET211 - Statistical Machine Learning

Course Type - Specialized Core – I L-T-P Format 3-0-2 Credits - 4

COURSE SUMMARY

This course includes machine learning concepts, Statistical Theories, Supervised learning, high dimensional data and the role of sparsity, Learning theory, Risk minimization, Classification and regression, and EM algorithm. It also covers important topics such as parametric and non-parametric methods, theory of generalization, regularization, the role of sparsity in high dimensional data, and surrogate loss functions. In a broader sense, the course offers a thorough understanding of statistical ML concepts that help students design and implement daily life learning applications.

COURSE-SPECIFIC LEARNING OUTCOMES (CO)

CO1: To articulate key features and methods of Statistical Machine Learning (SML).

CO2: To formulate and design the given application as a statistical machine learning problem.

CO3: To implement and evaluate common statistical machine learning techniques.

Detailed Syllabus

Module 1 (Contact hours: 7)

Statistical Theory, Supervised Learning, Unsupervised Learning, Data and Types, Feature variable, Machine Learning, Statistics terms, Supervised learning, Concentration inequalities, Generalization bounds, Plugin classifiers, Least-squares methods, Bias vs Variance, Theory of generalization, Understand Underfitting, Overfitting, Parametric methods, Maximum likelihood, Bayes algorithm, Minimax algorithm, Expectation-Maximization, Advantages and Disadvantages, Applications of EM Algorithm, Use case of EM Algorithm.

Module 2 (Contact hours: 10)

Bayesian versus Non-Bayesian approaches, Density estimation, Gaussian Distributions, Gaussian Mixture Models, Gaussian Discriminant Analysis, Independent Component Analysis, Convexity and Optimization: Convexity, Conjugate functions, Nonparametric classifications methods, Unconstrained optimization, Constrained optimization, Nonparametric methods, KKT conditions, Lagrangian minimization, Primal feasibility, Dual feasibility, Complementary slackness.

Module 3 (Contact hours: 13)

Basis pursuit, Polynomial Expansion, Feature maps, The “kernel trick”, Vapnik-Chervonenkis (VC) dimension, VC generalization bounds, Sparsity: High dimensional data, The role of sparsity, Sparsistency, Consistency, Persistency, Sparsity in nonparametric regression, Sparsity in graphical models, Greedy algorithms, Sparse linear regression, Compressed sensing, Nonparametric Methods: Nonparametric regression, Density estimation, Factor Analysis, Matrix Factorization, The bootstrap, Subsampling, Nonparametric Bayes.

Module 4 (Contact hours: 12)

Probability Distributions for modelling, Markov Networks, Hidden Markov Model, Advanced Theory: Concentration of measure, Covering numbers, Learning theory, Exact learning (Dana Angluin), Probably approximately correct learning (PAC learning), VC theory (Vladimir Vapnik and Alexey Chervonenkis), Risk minimization and its approaches, Bundle Methods, Graph Analytics, Graph-based machine learning algorithms, Simulation methods, Variational methods, Tsybakov noise conditions, Surrogate loss functions, Minimax rates for classification, Minimax rates for regression, Manifold methods, Spectral methods.

STUDIO WORK / LABORATORY EXPERIMENTS:

Students will gain practical experience with the implementation of different statistical methods by using different statistical machine learning tools. Eventually, the lab works formulate the problem as a statistical machine learning problem followed by its implementation.

TEXTBOOKS/LEARNING RESOURCES:

a) Masashi Sugiyama, Introduction to Statistical Machine Learning (1st ed.), Morgan Kaufmann, 2017. ISBN 978-0128021217.

b) T. M. Mitchell, Machine Learning (1st ed.), McGraw Hill, 2017. ISBN 978-1259096952.

REFERENCE BOOKS/LEARNING RESOURCES:

a) Richard Golden, Statistical Machine Learning A Unified Framework (1st ed.), CRC Press, 2020. ISBN 9781351051490.

CSET212 - Blockchain Foundations

Course Type - Specialized Core – I L-T-P Format 3-0-2 Credits - 4

COURSE SUMMARY

The widespread popularity of digital cryptocurrencies has led the foundation of Blockchain, which is fundamentally a public digital ledger to share information in a trustworthy and secure way. This course includes the fundamental design and architectural primitives of Blockchain, consensus protocols, types of the Blockchain system and the security aspects, along with various use cases from different application domains.

COURSE-SPECIFIC LEARNING OUTCOMES (CO)

CO1: To articulate the fundamentals of blockchain and able to explain cryptographic concepts underlying blockchain technology.

CO2: To examine various types of Blockchain networks and consensus algorithms.

CO3: To make use of wallet transactions, crypto tokens, analyse the block details and Blockchain network.

Detailed Syllabus

Module 1 (Contact hours: 11)

Why Blockchain Technology, Blockchain Bitcoin blockchain, Blockchain Architecture, Conceptualization, Blockchain components, Cryptocurrencies, Characteristics of cryptocurrencies, Alt coins, Crypto wallets, Creation of Blocks, Wallet Transactions, Transaction details in a Block, Merkle Tree, Hash functions, pseudo random numbers, Puzzle friendly and collision resistant hash, public key cryptosystem, Generation of keys, Digital signatures, Zero-knowledge systems.

Module 2 (Contact hours: 12)

Blockchain types-Public Blockchain, Private Blockchain, Federated Blockchain, Permissionless, Permissioned Blockchain Networks, Ethereum blockchain, Go Ethereum, Gas, Gas price, Gas Limit, ETH, MetaMask, Public Test Networks, set up a Ethereum node using Geth, Mining in Blockchain, Steps in Minnig, Double spending, Consensus protocols, PoW, Hashcash, Attacks on Bitcoin, Sybil Attacks, 51% Attack, eclipse attacks, DDoS Attacks, Replay Attacks, Byzantine fault, node failure.

Module 3 (Contact hours: 10)

Proof of Stake, Difference between PoW vs PoS, Byzantine General Problem, BFT (Byzantine fault tolerance), PBFT (Practical Byzantine fault tolerance), Delegated Proof of Stack, Paxos Consensus algorithm, Raft Algorithm, Solo Miner, Pool Miners, Smart contracts in Blockchain, Solidity, Data types in solidity, Operators, State variables, Global Variables, Local variables.

Module 4 (Contact hours: 9)

Remix, Compilation of smart contracts, Deployment environments, JavaScript Environment, Injected Web3, Web3 Provider, Solidity arrays, Solidity functions, Structs in solidity, Inheritance, Special variables, Solidity mapping, Function overloading, Personal Blockchain network, Ganache, Contract deployment to Ganache network, Modifiers in solidity, Events.

STUDIO WORK / LABORATORY EXPERIMENTS:

In studio work, students will learn to implement various cryptographic primitives such as symmetric/asymmetric cryptosystems, digital signatures and hash functions. Further, students will setup a MetaMask Ethereum wallet in the web browser to create wallets (User Accounts) and use it to send and receive Ethers. Use EtherScan to view the transaction details, explore Blockchain test networks to perform transactions, execute smart contracts, and launch Dapps. This lab provides a platform to create a genesis block to set up the private blockchain network using Go Ethereum (Geth) and Mist, start mining with miner.start() command. In addition, students will learn creating and deploying the simple smart contracts like “Hello World”, incrementing/decrementing the counter variable on the Blockchain network.

TEXTBOOKS/LEARNING RESOURCES:

a) Bettina Warburg, Bill Wanger and Tom Serres, Basics of Blockchain (1st ed.), Independently published, 2019. ISBN 978-1089919445.

b) Holbrook and Joseph, Architecting enterprise blockchain solutions (1st ed.), John Wiley & Sons, 2020. ISBN 978-1119557692.

REFERENCE BOOKS/LEARNING RESOURCES:

a) Bashir and Imran, Mastering blockchain: “Distributed ledger technology, decentralization, and smart contracts explained (1st ed.), Packt Publishing Ltd, 2018. ISBN 978-1788839044.

CSET213 - Linux and Shell Programming

Course Type - Specialized Core – I L-T-P Format 2-0-4 Credits - 4

COURSE SUMMARY

This course describes the essential ideas behind the open-source operating system approach to programming. Knowledge of Linux and shell script helps to understand the backbone of cybersecurity. This course involves basic Linux commands, Shell scripting, File structure and management, Processes, Inter-process communication, Socket programming, and security.

COURSE-SPECIFIC LEARNING OUTCOMES (CO)

CO1: To articulate Linux commands that are used to manipulate system operations at an admin level.

CO2: To write Shell Programming using Linux commands.

CO3: To design and write applications to manipulate internal kernel-level Linux File systems.

Detailed Syllabus

Module 1 (Contact hours: 8)

Linux and Linux utilities, Architecture of Linux, features of Linux, Introduction to vi editor, Linux commands, File handling utilities, security by file permissions, process utilities, disk utilities, Networking commands, Text processing utilities, backup utilities, User management.

Module 2 (Contact hours: 8)

Shells need and types, Derived Operators, Linux session, Standard streams, Redirection, Pipes, Tee command, command execution, command-line editing, Quotes, command substitution, job control, aliases, variables, shell/environment customization, Filters, and pipes, File operations.

Module 3 (Contact hours: 12)

Grep Operation, Grep Commands, Grep Address, Grep Application, Seed Scripts, operation, Unix file structure, File descriptors, System calls and device drivers, File management, File structures, System calls for file management, Directory API, Process and Process Structure, Process table, Viewing processes, System processes, Process scheduling, zombie processes, orphan process, Fork and its operation, Signals functions, unreliable signals, interrupted system calls, Signal sets, File locking, Threats and Vulnerabilities analysis of Linux- direct, indirect, veiled, conditional, Security Measures in Linux-SSH key pair, Scan Log files , Close Hidden ports, Linux Malwares- Botnets, Ransomware, Rootkits, Socket, Socket communications, UDP, TCP, AWK, Shell Scripting and Security- Password Tester, Permissions and Access Control Lists, Shell Scripting for DevOps- Using environment variables, Bash Script.

STUDIO WORK / LABORATORY EXPERIMENTS:

Students will use LINUX / UBUNTU to gain hands-on experience on LINUX and Shell programming, Linux commands, their uses and practice, editors: vi, nano etc, Introduction to Shell, Shell basic commands, variables Shell programming environments- filters and pipe, Shell programming File handling, Grep its use and commands. Using of Grep with pipe and filters, Unix file structuring, inodes and related system calls. File handling commands and API, Network Penetration testing tools, Wireshark, Nmap, Hash cat, Process management, creation, termination and other useful commands, Process scheduling. Parent, zombie and orphan process, Process system calls. Fork, exec, wait and signal, various commands. Basics of Socket Programming via UDP socket.

TEXTBOOKS/LEARNING RESOURCES:

a) M. Ebrahim and A Mallett, Mastering Linux Shell Scripting: A Practical Guide to Linux Command-Line Bash Scripting, and She (2nd ed.), Packt Publication, 2018. ISBN 978-17889905542.

b) Richard Blum and Christine Bresnahan, Linux Command Line and Shell Scripting Bible (1st ed.), Wiley, 2020. ISBN 9781119700930.

REFERENCE BOOKS/LEARNING RESOURCES:

a) W.R. Stevens, UNIX Network Programming (3rd ed.), PHI Publications, 2017. ISBN 978-8120307490.

CSET214 - Data Analysis using Python

Course Type - Specialized Core – I L-T-P Format 2-0-4 Credits - 4

COURSE SUMMARY

This course combines the advantages of both Python and Data Science. Here, students will learn to apply the ideas of analytics in real-world problems. Moreover, multiple mathematical operations and scientific computing will be taught using existing and mature python-based libraries.

COURSE-SPECIFIC LEARNING OUTCOMES (CO)

CO1: To articulate the structured and unstructured data for extracting useful information.

CO2: To work with data through visualization and distributions.

CO3: To make use of real-world datasets utilizing various numerical libraries.

Detailed Syllabus

Module 1 (Contact hours: 8)

Purpose of data analysis, Structured and Unstructured data, Steps of data analysis, Python Packages for Data Analysis: Numpy, Scipy, Matplotlib, Plotyly, NLTK. Data Frames, Usage of frames analytical roles, File handling and reading data for processing, Pre-processing data using multiple python frameworks, Data Formatting, Data Manipulation, Data normalization, Data Merging, Data reshaping, Data Wrangling, Aggregation functions.

Module 2 (Contact hours: 10)

String Manipulations, Demonstrating string functions, A regular expression for data, manipulation, Data Visualization, Using Histograms, Using Boxplots, Plotting data, Venn Diagram, Bar Chart, Pie Chart, Line Chart, Scatter Plots and R2, Grouped charts, Area Charts, Descriptive Statistics, Central tendencies, Analyzing variability, Data Distributions, Random Variables, Bernoulli Distribution, Binomial Distribution, Normal Distribution, Statistical Properties, Standard Normal Distribution.

Module 3 (Contact hours: 8)

Exponential distribution, Statistical test, Hypothesis testing, Z-test, Right-tailed test, Two-tailed test, T-Test, Significance of p-value in t-test, Two-sample Z-test, Paired t-test, Visualizing statistical test analysis, Model building, Outlier Detection.

STUDIO WORK / LABORATORY EXPERIMENTS:

Studio work focuses on Different Data Analysis Methods, Techniques, Algorithms using PythonData manipulation using numpy and scipy. Make use of numpy arrays, matrices, indexing and slicing options with the demonstration of numerical packages for data analysis.

TEXTBOOKS/LEARNING RESOURCES:

a) Bharti Motwani, Data Analytics using Python (1st ed.), Wiley, 2020. ISBN 8126502959.

b) Klosterman and Stephen, Data Science Projects with Python: A Case Study Approach to Successful Data Science Projects Using P (1st ed.), Packt Publishing Limited, 2019. ISBN 978-1838551025.

REFERENCE BOOKS/LEARNING RESOURCES:

a) Suresh Kumar Mukhiya and Usman Ahmed, Hands-On Exploratory Data Analysis with Python: Perform EDA techniques to understand, summarize, and investigate your data (1st ed.), Packt Publishing Limited, 2020. ISBN 978-1789537253.

CSET215 - Graphics and Visual Computing

Course Type - Specialized Core – I L-T-P Format 3-0-2 Credits - 4

COURSE SUMMARY

Introduction to computer graphics, graphics primitives, 2 D and 3 D transformations, representations and visualization, clipping, basic rendering, perceptual and cognitive foundations, basic animation techniques, keyframing.

COURSE-SPECIFIC LEARNING OUTCOMES (CO)

CO1: To demonstrate various graphics primitives and 2-D, 3-D geometric transformations and clipping techniques.

CO2: To comprehend the concepts related three-dimensional object representations.

CO3: To demonstrate the use of OpenGL to create interactive computer graphics applications.

Detailed Syllabus

Module 1 (Contact hours: 10)

Importance of Computer Graphics, Video Display Devices, Random scan displays, raster scan displays, Direct view storage tube, Flat panel displays, Latest display on mobile devices, I/O Devices, DDA Algorithms for Line drawing, Bresenham's algorithm, Circle, Ellipse, Arcs and Sectors, Boundary Fill Algorithm, Flood Fill algorithm, Transformations and Projections, 2D and 3D Scaling, Translation, Rotation, Shearing, Reflection, Composite transformation, Composite transformation, Window to View port Transformation, Orthographic projection, Perspective Projections.

Module 2 (Contact hours: 13)

Cohen Sutherland clipping, Liang Barky clipping algorithm, Sutherland Hodgeman polygon clipping, Weiler Atherton Polygon clipping algorithm, Three-Dimensional Object Representations, 3D Modelling transformations, Parallel and Perspective projection, Sutherland Hodgeman 3D clipping, Curved lines & Surfaces, Spline representations, Spline specifications, Bezier Curves & surfaces, B-spline curves & surfaces, Rational splines, Displaying Spline curves & surfaces, Rendering, Polygonal representation.

Module 3 (Contact hours: 12)

Affine and coordinate system transformations, Visibility and occlusion, Depth buffering, Painter's algorithm, Ray Tracing, Forward rendering, Backward rendering, Phong Shading, Per pixel per vertex Shading, Visualization of 2D/3D scalar fields, Color mapping, ISO surfaces, Direct volume data rendering, Raycasting, Transfer functions, segmentation, Visualization of: Vector fields, flow data.

Module 4 (Contact hours: 7)

Interpolation, Time-varying data, High-dimensional data, dimension reduction, Parallel coordinates, Non-spatial data, Multi-variate, Texture mapping, Animation Techniques, Keyframing.

STUDIO WORK / LABORATORY EXPERIMENTS:

Lab work would be conducted in OpenGL. All the basic drawing, filling, transformation and clipping algorithms will be covered.

TEXTBOOKS/LEARNING RESOURCES:

a) Pulasthi Gunawardhana, Computer Graphics: Theory and Practical (1st ed.), Scientific Research Publishing, Inc. USA, 2020. ISBN 9781618969021 .

b) Aditi Majumder and M. Gopi, Introduction to Visual Computing: Core Concepts in Computer Vision, Graphics, and Image Processing (2nd ed.), Taylor & Francis Group, 2018. ISBN 9781482244915 .

REFERENCE BOOKS/LEARNING RESOURCES:

a) William Thompson, Roland Fleming, Sarah Creem-Regehr and Jeanine Kelly Stefanucci, Visual Perception from a Computer Graphics Perspective (3rd ed.), Taylor & Francis Group, 2016. ISBN 9780367659288.

CSET216 - UI/UX Design for Human Computer Interface

Course Type - Specialized Core – I L-T-P Format 2-0-4 Credits - 4

COURSE SUMMARY

Overview of UI frameworks specifically Design, and Experience is targeted in this course. UI framework that lets front-end web developers add style and structure to websites very quickly. You will learn about how to layer elements in the z-axis, modify the shape and position of elements using transform, and create simple state transitions and animations. Introduction to UI/UX to improve coding abilities by learning how to keep track of the different versions of your code and configuration files. At the end of this course, you will be able to set up, design and style a web page using designing tools and their components.

COURSE-SPECIFIC LEARNING OUTCOMES (CO)

CO1: To explain UI design with components and user experience designing.

CO2: To implement UI/UX components and libraries.

Detailed Syllabus

Module 1 (Contact hours: 11)

Historical evolution GUI, Interactive system design: Concept of usability, HCI and software engineering, GUI design and aesthetics, Prototyping techniques, Heuristic Evaluation, Experimental Design, Importance and benefits of good design, Screen design. Scenarios, Design Process. Characteristics of user interface, Web user – Interface popularity, Psychology and Human factors, Conceptual Models, Mistakes and Error, Cognitive models, Socio-Organizational and stakeholder requirements, Social Computing, Experiments designing concepts and methods, Communication and collaboration models, Mobile Ecosystem: Platforms.

Module 2 (Contact hours: 10)

Application frameworks, Widgets, Applications, Mobile Design: Elements, Tools, Mobile design case studies, Human interaction with computers, design rules, Universal Design Models and Theories, Interface implementation & evaluation, Communication and collaboration models, Power of the Crowd, crowdsourcing, Internet of things in HCI, Experimental Design. Experiments designing concepts, Exploring design for HCI, GOMS for HCI, Identify Color Guidelines, Stages of action in interaction, Menu types and design, Construction of Prototype, Design Principles for HCI.

Module 3 (Contact hours: 7)

HCI for navigation design, Forms and behavior, Design and data for HCI, One factor test with human subjects, Ubiquitous computing with design analysis, Interface implementation, A/B testing, T-tests, Data assumptions and distributions, Interpretation of non-numeric response, Generalized linear models Case studies.

STUDIO WORK / LABORATORY EXPERIMENTS:

Students will learn to understand the trouble of interacting with machines and design a system based on user-centered approach.

TEXTBOOKS/LEARNING RESOURCES:

a) Samit Bhattacharya, Human-Computer Interaction User-Centric Computing for Design (1st ed.), McGraw-Hill, 2019. ISBN 9789353168056.

b) Nirmalya Thakur; Parameshachari B.D, Human-Computer Interaction and Beyond: Advances Towards Smart and Interconnected Environments (Part I), (1st ed.), Bantham Books, 2021. ISBN 9789814998826.

REFERENCE BOOKS/LEARNING RESOURCES:

a) Helen Sharp, Jennifer Preece, Yvonne Rogers, Interaction Design Beyond Human-Computer Interaction, (1st ed.), Wiley, 2021. ISBN 9781119547358.

CSET217 - Software Development with DevOps

Course Type - Specialized Core – I L-T-P Format 3-0-2 Credits - 4

COURSE SUMMARY

DevOps practices aims at merging development, quality assurance, and operations (deployment and integration) into a single, continuous set of processes. The key topics consists of advanced DevOps concepts, Cloud Services, Integration, Deployment, Testing, Configuration Management Tools, Containerization, Continuous Monitoring.

COURSE-SPECIFIC LEARNING OUTCOMES (CO)

CO1: To articulate the fundamentals of Linux, Containerization, Build and Release Management.

CO2: To examine DevOps tools for DevOps automation.

CO3: To make use Web Applications and platforms using DevOps tools.

Detailed Syllabus

Module 1 (Contact hours: 11)

Fundamentals of Linux: Linux vs Unix, Linux File System Commands, Start and Stop Services, Web Application on Cloud, Overview of addressing in networks, Allocation, Association and Releasing of Elastic IP Address, Storage Services on Cloud, Buckets on Cloud (Creation, Storing Data, Cross Region Replication), Elastic Block Storage (EBS) and its types, Types of EBS Volumes, Creation, Attaching and Detaching Volumes, ELB (Elastic Load Balancer), Configuration of ELB, Addition of Webserver under ELB, IAM (Identity Management Access) Security using IAM, Creating IAM Users and Managing Password Policies.

Module 2 (Contact hours: 8)

Container: Objective and Concept, Container Vs Virtual Machine, Docker introduction and installation on Different OS (CentOS, Windows, etc.), Docker Commands to Maintain Container, Docker Compose, Docker: Registry, Hub, Communication/Networking within Single Docker Container, Communication/Networking across Docker Container, Building Docker Images.

Module 3 (Contact hours: 10)

CICD and its Pipeline, Implementation and Principles of CICD Pipeline, Tools for CICD DevOps, CICD Project Overview, Industries providing CICD Services, Package Management, Release Pipeline, Build Management, Build Reporting and Status, Release Planning, Dependency Management, Dependency Identification, Dependency Scope, Dependency Features, Dependency Management Examples (Parent POM, Child POM-a, Child POM-b), Project Release Cycle, Deploying build to production.

Module 4 (Contact hours: 13)

DevOps Automation, The Software Delivery Pipeline, Fully Automated Software Delivery Process, Automated build, Automated Test, Automated Deployment, Automation Scenarios, Archiving Logs and its auto discard, Auto Backups, Web Server Automation, Automated Usage Alarm/message, Scenarios Where Automation Prevents Errors, Writing Automation Scripts, Automated Task Scheduling, Best Practices for Automated Scripting, Configuration Management (CM), Use of CM in Industries, CM Tools, Configuration of tools (Ansible, Puppet, Chef, SaltStack, Managing CM Infrastructure, Comparison of Ansible, Puppet, Chef, and SaltStack.

STUDIO WORK / LABORATORY EXPERIMENTS:

The Studio work includes the practical understanding of multiple tools and languages that are used in DevOps Industry. DevOps practices aims at merging development, quality assurance, and operations (deployment and integration) into a single, continuous set of processes.

TEXTBOOKS/LEARNING RESOURCES:

a) Rafał Leszko, Continuous Delivery with Docker and Jenkins: Create secure applications by building complete CI/CD pipelines (2nd ed.), Packt Publishing, 2019. ISBN 978-1838552183.

b) Steven Branson, DevOps: Building Software with Lean Process for Modern Business (2nd ed.), Independently published, 2019. ISBN 978-1675741214.

REFERENCE BOOKS/LEARNING RESOURCES:

a) Eric Chou, Michael Kennedy and Mandy Whaley, Mastering Python Networking: Your one-stop solution to using Python for network automation, programmability, and DevOps (3rd ed.), Packt Publishing, 2020. ISBN 978-1839214677.

CSET218 - Full Stack Development

Course Type - Specialized Core – I L-T-P Format 2-0-4 Credits - 4

COURSE SUMMARY

The course covers Full Stack development, Client-server Architecture, Front end programming tools, frameworks, and libraries.

COURSE-SPECIFIC LEARNING OUTCOMES (CO)

CO1: To understand the Client-Server architecture and the basics of client-side programming languages and frameworks.

CO2: To examine of client-side languages: HTML, CSS, and JavaScript.

CO3: To implement client-side application development.

Detailed Syllabus

Module 1 (Contact hours: 8)

Why Full Stack Development? Web development vs FullStack Development , Client-Server architecture , Rules of three-tier architecture, MEAN, MERN, Rails, Django Stack and LAMP, MEAN vs MERN stack, Front End Frameworks and Libraries, Web, Web Browser, Web Server, Anatomy of a Website, Developer tools, inspector, Wireframing , Web hosting steps, HTML, HTML Document Object Model, W3C standards for HTML, HTML Validation, Structural markup , Semantic markup, HTML Lists, Links , Absolute versus relative path names , URL: Anatomy, Types, HTML Formatting , HTML Tables, Meta tags, Structural tags, Character entities, escape codes, Image maps, Font awesome Icons , Forms Input Types.

Module 2 (Contact hours: 10)

Why CSS? W3C CSS Validator, Syntax, Types, CSS Selectors, Cascading, Inheritance, Specificity, Units of Measure, Width and Height of element, Box Model Layout, Border Box Versus Content Box, Responsive website Design Bootstrap Grid System, CSS pre-processor: Less, Sass and features.

Module 3 (Contact hours: 10)

JavaScript, JavaScript Events, JavaScript APIs, Expert Lecture from Industry, Ajax Web App Flow, Ajax Process Refreshing ES6 Specifications and Features ECMA Script, ES6 let and const, The arrow functions, ES6 Classes Inheritance, Spread Operator, Iterators and Generators, React, React in HTML, React in CSS, Built in libraries Leveraging Virtual DOM, Setting up React, Configure git/client, Creating Repositories, push local repositories to GitHub, Git Branching and Merging , Web Publishing ethics, Hosting a website.

STUDIO WORK / LABORATORY EXPERIMENTS:

In studio work, developers develop both the front-end and the server-side of the application, deploy, debug and maintain their databases and servers. Being a Full Stack Web Developer, you will be at an edge as you make better technical decisions and see the big picture.

TEXTBOOKS/LEARNING RESOURCES:

a) Flanagan, D., JavaScript: The Definitive Guide (7^{th} ed.), Shroff and O'Reilly Media, 2020. ISBN 978-935213996.

b) Robbins, J., Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics (5^{th} ed.), O'Reilly Media, 2020. ISBN 978-1491960158.

REFERENCE BOOKS/LEARNING RESOURCES:

a) Meloni, J.C. and Kyrnin J, HTML, CSS, and JavaScript All in One, Sams Teach Yourself (5^{th} ed.), Pearson, 2018. ISBN 978-9389552416.

CSET219 - Quantum Computing Foundations

Course Type - Specialized Core – I L-T-P Format 3-0-2 Credits - 4

COURSE SUMMARY

Introduction to classical and quantum computing, the basic concepts of quantum computing, including a section on mathematical tools, the classical and quantum circuit models, quantum algorithms and quantum computing approaches.

COURSE-SPECIFIC LEARNING OUTCOMES (CO)

CO1: To articulate the differences between quantum and classical computation.

CO2: To understand the mathematical description of quantum states and basic quantum operations.

CO3: To become proficient with engineering requirements for quantum vs classical algorithm implementation.

Detailed Syllabus

Module 1 (Contact hours: 12)

Quantum computing, Evolution of Quantum System on a Classical System, Deterministic Systems, Probabilistic Systems, Quantum System: Physical phenomena, Quantum Superposition principle, Normalization, Orthonormalization, Coherence, Decoherence, Postulates of Quantum Computing-I: Hermitian, Unitary, Pauli matrices, Complex Vector Space, State Space: Hilbert Space. Postulates of Quantum Computing-II: Dirac's Bra-ket Notation, Eigenspace: eigenkets and eigenstate, Dense Matrix, Direct Sum, Tensor, Expectation Values and Variances. Quantum states and Qubit, the atoms of computation, Representing Qubit States, Single Qubit Gates, The Case for Quantum, Multiple Qubits and Entanglement, Qutrit and Qudit, Proving Universality, Classical Computation on a Quantum Computer.

Module 2 (Contact hours: 13)

Defining Quantum Circuits, Quantum Teleportation, Variational Quantum eigen solvers, Quantum Algorithms, Deutsch-Jozsa Algorithm: Classical and quantum solution, Constant and Balanced Oracle, Bernstein-Vazirani Algorithm: Classical and quantum solution, Simon's Algorithm: Classical and quantum solution, Upper bound and Lower bound, Quantum Fourier Transform, Quantum Phase Estimation, Shor's Algorithm: Factoring, Reduction from factoring to period-finding, Shor's period-finding algorithm, Continued fractions, Grover's Algorithm: Unstructured search problems, 2-Qubits and 3-Qubits using oracle, Amplitude amplification, Optimality of Grover's Algorithm, Reduction to Three Inequalities, Proofs of the Three Inequalities, Quantum Counting, Controlled-Grover Iteration, Inverse QFT, Classical random walks, Classical Markov chains, Quantum walks, Quantum Walk search Algorithm, On hypercube, QWSA Collision problem, Finding a triangle in a graph.

Module 3 (Contact hours: 8)

The HHL Algorithm, solving linear systems of equations using HHL, Improving the efficiency of the HHL algorithm, Combinatorial Optimization problems, Quantum Approximate Optimization Algorithm, Satisfiability Problems, Solving Satisfiability problems using Grover's Algorithm, Variational Quantum linear solver, Solving the Travelling salesman problem using, phase estimation, Quantum Query Algorithm, The polynomial method, The quantum adversary method, Quantum Complexity Theory, Classical and quantum complexity classes, classically simulating quantum computers in polynomial space.

Module 4 (Contact hours: 9)

Quantum Merlin-Arthur Problem, The local Hamiltonian problem, Local Hamiltonian is QMA-complete, Quantum Encodings, Mixed states and general measurements, Quantum encodings and their limits, Classical error-correction, Quantum error-correction, Quantum error-correcting codes, Fault-tolerant quantum computation, Concatenated codes and the threshold theorem, Quantum error-correction using repetition codes, Measurement error mitigation, Density Matrix and Mixed States, Quantum Image Processing: FRQI and NEQR image representation, Quantum Edge Detection.

STUDIO WORK / LABORATORY EXPERIMENTS:

In-studio work, students will be setting up the environment for programming using Qiskit, they will be designing and analyzing quantum algorithms.

TEXTBOOKS/LEARNING RESOURCES:

a) Jack D. Hidary, Quantum Computing: An Applied Approach (1st ed.), Springer, 2019. ISBN 978-3-030-23921-3.

b) Ronald de Wolf, Quantum Computing: Lecture Notes (1st ed.), arXiv, 2021. ISBN arXiv:1907.09415v2.

REFERENCE BOOKS/LEARNING RESOURCES:

a) Venkateswaran Kasirajan, Fundamentals of Quantum Computing (1st ed.), missing, 2021. ISBN 9783030636895.

CSET220 - Unmanned Aerial Vehicles

Course Type - Core L-T-P Format 2-0-4 Credits - 4

COURSE SUMMARY

This course covers the fundamental concepts and technological advancements behind rapidly escalating unmanned aerial deployments. Understanding basic geometrical, aeronautical, electronics, sensory, and software components behind drones. Utilize these basic components to design a Multicopter drone using off-the-shelf controllers.

COURSE-SPECIFIC LEARNING OUTCOMES (CO)

CO1: To articulate UAVs, UAV application, needs, deployments, and advancements.

CO2: To understand design concepts, aerodynamic considerations, electronic attachments, and mission-specific deployment of aerial nodes.

CO3: To construct multirotor UAV from scratch using various flight controllers.

Detailed Syllabus

Module 1 (Contact hours: 6)

Why Unmanned Aerial Vehicle (UAV)? Introduction to UAVs/Drones; Unmanned Aerial Systems (UAS); History and Modern Trends; Technological Advancements; UAV Concept of Operations; UAV Chemical, Biological, Radiological and Nuclear (CBRN); UAV Classification: Wing Geometry Based, Payload Based, Size Based, Mission Based; UAV Anatomy; Mission Planning.

Module 2 (Contact hours: 7)

Basic UAV Aerodynamics: Wing Configuration and Thrust; Airframe Configuration; Lift and Drag; Aerodynamic Configuration; Vibration and Noise; Endurance; Gliding Flight; UAV Stability; UAV Propulsion; Propeller Modelling; Motor Modelling.

Module 3 (Contact hours: 8)

Coordinate Frame; Earth-Fixed Coordinate Frame; UAV Communication: Radio Wave Communication, Microwave Communication, Line of Sight Communication, Beyond Line-of-Sight Communication; UAV Navigation; Coordinate Frame; Waypoint Tracking Navigation; Sensor Calibration and Measurements; Tracking using GPS; Tracking using Data; State Estimation; Attitude Estimation; Position Estimation.

Module 4 (Contact hours: 7)

UAV Controllability; Flight Control; Position Control: Euler Angles as Outputs, Rotation Matrix as Outputs; Autonomous Control; Semi-Autonomous Control; UAS Safety; Failsafe.

STUDIO WORK / LABORATORY EXPERIMENTS:

In this course students will start with basic Multicopter design principles, components, and assembly precautions. Then finally design and develop a fully functional Multicopter UAV/Drone using off the shelf Flight Controllers.

TEXTBOOKS/LEARNING RESOURCES:

a) Quan Quan, Introduction to Multicopter Design and Control (1st ed.), Springer, 2017. ISBN 9789811033810.

REFERENCE BOOKS/LEARNING RESOURCES:

a) Mohammad Sadraey, Unmanned Aircraft Design: A Review of Fundamentals (1st ed.), Morgan & Claypool, 2017. ISBN 1681731681.

CSET221 - Robotic Process Automation Essentials

Course Type - Specialized Core – I L-T-P Format 2-0-4 Credits - 4

COURSE SUMMARY

This course covers the use and functionality of Robotic Process Automation (RPA) in business prospects. It familiarizes with the concepts related to RPA and how RPA is useful to automate business processes. The goal is to train the learners about the RPA tool for developing applications that can automate the functionalities in an organization.

COURSE-SPECIFIC LEARNING OUTCOMES (CO)

CO1: To understand the underlying logic/structure related to robotic process automation.

CO2: To examine RPA bots to automate the processes.

CO3: To make use of transactions in business processes using robotic process automation.

Detailed Syllabus

Module 1 (Contact hours: 10)

Process Automation, Application and Evaluation of RPA, Flowchart, Sequence, State Machine, Activities in RPA, Activities to print the log, How to get the input from user, How to send the data to output device i.e. Monitor, Variables & Data type in RPA tool, Managing packages, Publishing of Bot, If Statement, For loop, For each row, Switch, Parallel, Do-while, While, Conversion of Pseudocode in RPA tool, Factorial, Fibonacci code in RPA tool, Selectors, UI Element.

Module 2 (Contact hours: 11)

Debugging, Use of Try-Catch for exception handling, .Net Framework editor, .Net Variable, Loops and conditional statement, Extensible Markup Language, Tools for writing XML, Rules for writing XML, Elements attributes and values, Creating the root and child elements, Attended bot, Unattended bot, Floating bot.

Module 3 (Contact hours: 7)

Orchestrator, Orchestrator Entities, Tenants, and folder, Robot Provisioning and License Distribution, Expert Lecture from Industry, Unattended Automation with Folders, Orchestrator Resources, Libraries and Templates in Orchestrator, Setting up an Orchestration Process Activities, Queues for Long-running Activities, Tasks Long-running Activities, Jobs Long-running Activities, Formation of RPA Team, Process Design Document/Solution, Design Document. Storage Buckets, Queues, Intermission- Transaction Processing Models, Robots Hosting, Processing Hosting, Triggers and SLAs, Monitoring and Alerts, Advanced Topics in RPA, RPA Frameworks; Transaction, Queue, Machine, Variable declaration.

STUDIO WORK / LABORATORY EXPERIMENTS:

The studio work covers working with the RPA tool Community edition, developing the automation applications as the projects with its detailed demonstration.

TEXTBOOKS/LEARNING RESOURCES:

a) Richard Murdoch, Robotic Process Automation: Guide to Building Software Robots, Automate Repetitive Tasks & Become an RPA Consultant (1^{st} ed.), Independently published, 2018. ISBN 978-1983036835.

b) Taulli, T, The robotic process automation handbook (1^{st} ed.), Berkeley, CA: Apress, 2020. ISBN 9781484257289.

REFERENCE BOOKS/LEARNING RESOURCES:

a) Nandan Mullakara and Arun Kumar Asokan, Robotic Process Automation Projects (1^{st} ed.), Packt Publishing, 2020. ISBN 9781839210958 .

CSET222 - Microcontrollers, Robotics & Embedded Systems

Course Type - Specialized Core – I L-T-P Format 2-0-4 Credits - 4

COURSE SUMMARY

The course explains microcontrollers and their components. It also provides exposure to Raspberry Pi as it is required to communicate with hardware devices and develop Embedded Systems. Also, executes Robotics Operating System (ROS) which will provide knowledge to develop various Robots.

COURSE-SPECIFIC LEARNING OUTCOMES (CO)

CO1: To explain the micro-controllers, robotics, embedded system, and their applicability.

CO2: To build the various robots using Robot Operating System.

CO3: To implement the various Embedded system using microcontrollers and ROS.

Detailed Syllabus

Module 1 (Contact hours: 6)

Application of Microcontrollers, Why Embedded system, Application of Robotics, Arduino: Architecture, Serial Port, Serial Communication, Device control using Serial Communication, Arduino Sensors (Humidity, Temperature, Water Detector, Ultrasonic, LDR), Arduino secondary Integration (Relay, DC, Servo motors, RF modules), Pulse width modulation (PWM), I2C communication protocol, BH1750: Digital light sensor, Parallel Communication, Arduino UART, GSM, GPRS Module.

Module 2 (Contact hours: 6)

Raspberry Pi: Architecture, Raspberry Pi Port Identification, Raspberry Pi GPIO, Transistorized Switching, Accepting Digital Input on Raspberry Pi, Enabling I2C to Raspberry Pi, Analog, and Digital sensors, BMP180 with Pi, Sensors Interface with Pi LDR, Sensors Interface with Pi DHTT11, Sensors Interface with Pi using Sense HAT.

Module 3 (Contact hours: 9)

Fundamentals of Robotics, Robot Operating System (ROS), ROS Essentials: ROS Topics, ROS: Services, Actions, Nodes, Build Robot Environment, Unified Robot Description Format (URDF), ROS parameter server, ROS Services, and parameters, Recording and playing back, reading messages from a bag file, using rosed to edit files in ROS, ROS msg and srv.

Module 4 (Contact hours: 7)

Simple Publisher and Subscriber, Examining the Simple Publisher and Subscribe, Simple Service and Client, Examining the Simple Service and Client, Motion in ROS (ROS Noetic), Working with Pluginlib, Nodelets, and Gazebo Plugins, Robot Navigation (moveit), Grasping, Grasping using MoveIt, creating a pick and place task, Grasping in the Real Robot, ROS Controllers and Visualization Plugins.

STUDIO WORK / LABORATORY EXPERIMENTS:

Studio work focuses to develop different kind of robots starting from object detection robot, line follower robot. It also consists of introduction to well exist robots such NAO, SOFIA, Pepper.

TEXTBOOKS/LEARNING RESOURCES:

a) Wyatt Newman, A Systematic Approach to Learning Robot Programming with ROS (1st ed.), Chapman and Hall/CRC, 2017. ISBN 9781498777827.

REFERENCE BOOKS/LEARNING RESOURCES:

a) Jonathan W. Valvano, Introduction to Robotics (1st ed.), Second printing, 2019. ISBN 9781074544307.

CSET223 - Augmented Reality Foundations

Course Type - Specialized Core – I L-T-P Format 3-0-2 Credits - 4

COURSE SUMMARY

Augmented Reality gives students the opportunity to create immersive Augmented Reality (AR) content for mobile devices. To provide students with a solid background in alternative 3D compositing techniques using computer vision with applications in interactive interfaces most notably augmented reality interfaces on mobile devices. Provide students with a comprehensive knowledge in 3D vision Develop skills in the design and development of interactive augmented reality games. Topics included: The human visual system, 3D Vision, Tracking system, Design and implementation of an immersive user experience, Interaction Techniques for Augmented Reality Collaborative Augmented Reality.

COURSE-SPECIFIC LEARNING OUTCOMES (CO)

CO1: Build a solid background in alternative 3D compositing techniques using computer vision with applications in interactive interfaces – most notably augmented reality interfaces on mobile devices.

CO2: Develop skills in the design and development of interactive augmented reality games and understand the practical issues regarding multi-platform reconstruction.

CO3: Develop interactive augmented reality applications for PC based mobile devices using a variety of novel input devices.

Detailed Syllabus

Module 1 (Contact hours: 10)

Industrial applicability of AR, AR Systems Overview, Input and Output Devices for AR, Optical vs. Video, See-Through AR, AR system architecture, Senses, Simple sensing/perception model, Human visual system, 3D Vision, Tracking system, Tracking for Augmented Reality, Importance of Accurate Head Tracking, Tracking Problem, Tracking Technologies.

Module 2 (Contact hours: 10)

Head motion prediction, Registration, Calibration – static and dynamic, Real Time Performance, Characteristics – spatial, temporal, system robustness, Scheduling and Fusing Sensor Information, mixed reality UI design, Design of immersive user experience, Alternative Interface Paradigms, Usability guidelines, immersive environments, Space, Scale, Ergonomics.

Module 3 (Contact hours: 10)

Physical locomotion techniques, Target based techniques, Steering, Comfort and distress, Gaze direction, comfort range test, Motion Sickness, Simulator Sickness, Cybersickness, AR Interface Design, Properties of AR Environments, Collaborative AR Interfaces, Heterogeneous AR User Interfaces, Tangible and Graspable Interaction.

Module 4 (Contact hours: 12)

Tracking for Augmented Reality, Augmented Reality Interaction, Augmented Reality, Information Browsers, AR Widgets, Graphical Interface Elements, Evaluating AR Interfaces, video mixing, optical blending, projection devices, spatially augmented reality, Immersive virtual reality, Desktop metaphor, mobile/wearable computing, Ubiquitous computing, Tangible user interfaces, Distributed graphics.

STUDIO WORK / LABORATORY EXPERIMENTS:

The laboratory of Augmented reality foundation is designed to provide a practical exposure to the students about the concepts and topics taught in the classroom sessions. Case Studies: - Face-to-Face Collaboration – Shared Space - Remote Collaboration – AR Conferencing, Wearable AR Conferencing - Seamless Collaboration – The Magic Book.

TEXTBOOKS/LEARNING RESOURCES:

a) Jesse Glover and Jonathan Linowes, Complete Virtual Reality and Augmented Reality Development with Unity (1st ed.), Packt Publishing;, 2019. ISBN 9781838644865.

b) Doug A Bowman, Ernest Kuijff, Joseph J LaViola, Jr and Ivan Poupyrev, 3D User Interfaces, Theory and Practice (2nd ed.), Pearson Education, USA, 2017. ISBN 978-0134034324.

REFERENCE BOOKS/LEARNING RESOURCES:

a) Erin Pangilinan, Steve Lukas and Vasanth Mohan, Creating Augmented and Virtual Realities (1st ed.), O'Reilly Publishers, 2019. ISBN 978-1492044192.

b) Dieter SCHMALSTIEG and Tobias HOLLERER, Principals and Practice, Augmented Reality (1st ed.), Pearson Education India, 2016. ISBN 978-0321883575.

CSET224 - Cloud Computing

Course Type - Specialized Core – I L-T-P Format 3-0-2 Credits - 4

COURSE SUMMARY

This course covers evolutionary computing paradigms from multi-processor systems to Cloud, Edge, and Fog Computing. The course design follows cloud deployment models, service models, virtualization techniques and cloud architectural solutions. It also elaborates cloud compliances and security at fine-grained level by following a shared responsibility model. It reveals a design pattern to the students, enabling them to think through the process of designing and implementing cloud infrastructure and optimal IT solutions.

COURSE-SPECIFIC LEARNING OUTCOMES (CO)

CO1: To articulate cloud computing principle and its business need.

CO2: To identify the design principles of virtualization techniques in cloud resource management.

CO3: To design and development of cloud architectural solution with its detailed monitoring.

Detailed Syllabus

Module 1 (Contact hours: 11)

Cloud Computing, Adoption of cloud-based IT resources, Service Models: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), Software-as-a-Service (SaaS), Deployment models: Public Cloud, Private Cloud, Hybrid Cloud, Community Cloud, Cloud Computing Characteristics, Challenges of cloud computing, Virtualization concept, Types of virtualizations, Demo of virtualization, Virtualization Merits, Role of virtualization in cloud computing, Virtualization Demerits, VM Placement, VM Migration, VM Migration Demo, VM clustering, Design Issues in VM Clustering, Need of Dockers and Containers, Docker Eco-System, Hypervisor vs Docker.

Module 2 (Contact hours: 12)

Microservices, Service-Oriented Architecture, REST API, IP Addressing, Subnetting, Supernetting, Designing of Virtual Private Cloud, Demo of VPC, VPC Peering, VPC Case Study, Cloud Storage, Serverless Computing, Cloud API Gateway, Cloud Databases, Resource Provisioning, Time shared and space shared, Efficient VM Consolidation on cloud server, Task/DAG Scheduling Algorithms, Min-Min, Max-Min, MET, B-level Demo, T-level Demo, Task-VM Mapping, Auto Scaling, Load Balancing.

Module 3 (Contact hours: 10)

Case Study: Cloud Market analysis, Security and Compliances, Shared security model in IAAS/PAAS/SAAS, Shared technology issues, Data loss or leakage, Account or service hijacking, Implementation of cloud security, Security Groups, Network Access Control Lists, Cloud databases, Parallel Query Execution with NoSQL Database, Big Data, Handling Big Data on Cloud Platform, Map- Reduce framework for large clusters using Hadoop, Design of data applications based on Map Reduce in Apache Hadoop.

Module 4 (Contact hours: 9)

Comparative study/analysis of public clouds, Edge Computing, Fog Computing, Data Offloading, Cloud-Based DevOps Tools, Task Partitioning, Data Partitioning, Data Synchronization, Distributed File System, Data center, Ongoing Research Topics.

STUDIO WORK / LABORATORY EXPERIMENTS:

Practical experience on global cloud infrastructure by performing experiments on Amazon Web Services (AWS), Google Cloud Platform (GCP) and Microsoft Azure platform. The essential services and their hands-on is compulsory on Core IaaS, PaaS and SaaS.

TEXTBOOKS/LEARNING RESOURCES:

a) Lizhe Wang, Rajiv Ranjan, Jinjun Chen and Boualem Benatallah, Cloud Computing (1st ed.), CRC Press, 2017. ISBN 978-1351833097.

b) Judith S. Hurwitz and Daniel Kirsch, Cloud Computing For Dummies (2nd ed.), Hoboken: John Wiley & Sons, 2020. ISBN 978-1119546658.

REFERENCE BOOKS/LEARNING RESOURCES:

a) Prerna Sharma, Moolchand Sharma and Mohamed Elhoseny, Applications of Cloud Computing (1st ed.), CRC Press, 2020. ISBN 9780367904128.

CSET225 - Intelligent Model Design using AI

Course Type - Specialized Core – II L-T-P Format 3-0-2 Credits - 4

COURSE SUMMARY

Here students will learn to apply the knowledge of machine learning and AI in practical business problems. They will use their critical thinking to solve multiple and existing issues faced by industry. Further, the course will involve looking at any problem from multiple perspectives like engineering, marketing, and deployment.

COURSE-SPECIFIC LEARNING OUTCOMES (CO)

CO1: To articulate key features of various Machine learning and Deep Learning tools for Intelligent Modelling.

CO2: To examine and implement intelligent applications using Machine Learning and Deep Learning tools.

CO3: To implement Machine Learning and Deep Learning models for design thinking and evaluate them using relevant metrics.

Detailed Syllabus

Module 1 (Contact hours: 8)

Why Intelligent Model? Intelligent Citizens of the World, Thinking More Clearly, Empathize, Define, Ideate, Prototype, Test, Using and Understanding Data, Using Models to Decide, Strategize, and Design, Automation/Augmentation Opportunities, AI in Sensitive Applications, AI in Healthcare and Security, Pervasive, AI Systems, AI in IOT Devices, Business Value of Data: Case Study on Data Dependent Company, Data Valuation, Data Quality, Data Driven Marketing and Commerce, Biases in AI Models, Handling Gender, Race, Religion, Cultural Biases, Unintended Biases.

Module 2 (Contact hours: 10)

User privacy, Data Privacy aware AI, Data Sharing Regulations, AI Model Security, Attacks on AI models, Adversarial Attack, Relevance Feedback, Adaptive Learning, Online learning, Model fine-tuning, Transfer Learning, Domain Adaption, Explainable AI, ML Accountability and Fairness, Model Interpretation, Class imbalance in Modelling, Handling Data Drift, Human-Machine Co-learning.

Module 3 (Contact hours: 12)

Aligning Mental and Machine Learning Models, Coding Automation, Rapid Prototyping in AI, Market Demand Analysis, Automated Requirement Analysis, Customer opinion analysis, Attribute based opinion mining, Geospatial Analytics, Spatial analysis, Spatial-temporal analysis, Geo-clustering, Market Segmentation, Demographic, Cultural, Behavioural and Psychographic Segmentation, Dynamic Pricing, Price Optimization, Personalization, Recommender Systems, Feature Attribution, SHAP (SHapley Additive exPlanations), Lime, Shapash, Dalex.

Module 4 (Contact hours: 12)

Speed vs Accuracy Trade-off, optimizing models for resource constrained devices, Deep Learning Model compression, Pruning, Low-rank factorization, Automatic Machine Learning, Neural Architecture Search, Deploying Machine Learning Models, Staging Automation, Cognitive Modelling, Conversational Systems, Data driven persona, User personas, Reasoning Under Uncertainty, Multi-agent System, Reinforcement Learning, General Intelligence, Super Intelligence, Decision Making Systems.

STUDIO WORK / LABORATORY EXPERIMENTS:

Students will gain practical experience with the implementation of different intelligent methods. The lab work consists of exploratory data analysis using tools, libraries and python programming language by using Machine Learning and Deep Learning models.

TEXTBOOKS/LEARNING RESOURCES:

a) Charu C. Aggarwal, Neural Networks and Deep Learning (1st ed.), Springer International, 2018. ISBN 978-3030068561.

REFERENCE BOOKS/LEARNING RESOURCES:

a) Garry Kasparov and Mig Greengard, Deep Thinking: Where Machine Intelligence Ends and Human Creativity Begins (1st ed.), John Murray, 2018. ISBN 978-1473653511.

CSET226 - Blockchain Engineering

Course Type - Specialized Core – II L-T-P Format 3-0-2 Credits - 4

COURSE SUMMARY

This course aims at the technical concepts of blockchain and how to develop software applications that use the career-enhancing skills needed to help you move into this fast-developing area of technology. Further, in this course, you will learn the technical concepts that underpin blockchains, how to design and implement blockchain applications, how to create and manage blockchain networks, distributed ledger technologies, and the future of Blockchain technology.

COURSE-SPECIFIC LEARNING OUTCOMES (CO)

CO1: To articulate blockchain concepts and software applications that are used to develop the blockchain use cases.

CO2: To perform compilation, migrating, testing, and deploying the blockchain applications on the decentralized network.

CO3: To make use of building distributed ledger technologies for Blockchain applications and able to solve double spending problems in the digital payments using cryptocurrencies.

Detailed Syllabus

Module 1 (Contact hours: 10)

Blockchain platforms, distributed ledger technologies, distributed consensus algorithms, endless chains, Proof-of-elapsed time, Proof of Burn, Hyperledger, Frameworks, Fabric network, Transaction flow, Fabric– Membership, Identity management, Fabric components, Fabric deployment, Hyperledger Composer, Application Development, Composer – Network Administration.

Module 2 (Contact hours: 11)

IPFS, Decentralized web, Privacy and encryption on IPFS, nodes, IPFS commands, IPFS vs HTTP, Location-based addressing, Content-based addressing, Distributed Hash Table (DHT), Merkle DAGs, IPNS, IPFS Gateway, IPFS Daemon, DNS Link, On-chin Transactions, Off-chain Transactions, Forking, Hard forks, Softworks, Double spending issues, Improving throughputs, Algorand, DAG-Chains, Hyper ledger Indy, Hyperledger Sawtooth.

Module 3 (Contact hours: 10)

Decentralized Autonomous Organizations, DAO Membership, Problems with automated immutable systems, Challenges with the decentralized web, Defi (decentralized finance), Defi on Ethereum, Non-Fungible Tokens (NFTs), NFT use cases, Gas pricing, Ethereum development using Whisper, Swarm, and Raiden Network, State Channels, Case study, Costless Verification: Blockchain Technology, last mile problem, Verification cost reduction in blockchain, Bootstrapping network effects.

Module 4 (Contact hours: 11)

Development with Solidity, Truffle framework, Sublime, testRPC, Function modifiers, contract-driven development, testing, DApps development, Native application development using Java, RPC, Membership Service Provider (MSP), Certificate authority, Blockchain in Financial Service, Revolutionizing Global Trade, Blockchain in Supply Chain, Blockchain with AI, digital privacy, Blockchain in IoT, Blockchain research trends: Cybersecurity with Blockchain, Blockchain in Drone research, Blockchain in SDN & 5G.

STUDIO WORK / LABORATORY EXPERIMENTS:

The laboratory work enables students to get practical experience in various Blockchain platforms like Ethereum and Hyper ledger to design, deploy Decentralised Applications (DApps) on Blockchain network using chain code, which include:1) Simple Land Registry DApp based on the Ethereum blockchain. 2) A blockchain-based Tollbooth Management System. 3) Creating a decentralized platform for the distribution of scholarship. 4) DApp for Organ Donor list and an Organ Recipient list which will be stored in a distributed data system. 5) A simple decentralized application, where officials can register for birth/death, and generate a certificate which can then be reverified at any point of time in the future (DocVerify), etc.

TEXTBOOKS/LEARNING RESOURCES:

a) Gaur, Nitin and et al, Hands-On Blockchain with Hyperledger: Building decentralized applications with Hyperledger Fabric and Composer (1st ed.), Packt Publishing Ltd, 2018. ISBN 9781788994521.

b) Pathak, Nishith and Anurag Bhandari, IoT, AI, and Blockchain for. NET: Building a Next Generation Application from the Ground Up (1st ed.), Apress, 2018. ISBN 9781484237083.

REFERENCE BOOKS/LEARNING RESOURCES:

a) Bashir and Imran, Mastering blockchain: “Distributed ledger technology, decentralization, and smart contracts explained” (1st ed.), Packt Publishing Ltd, 2018. ISBN 9781787125445.

CSET227 - System and Network Security

Course Type - Specialized Core – II L-T-P Format 3-0-2 Credits - 4

COURSE SUMMARY

System and network security courses cover various types of attacks, security issues and defense mechanisms for the systems working in a networked environment. This course includes the following key topics: Basic concepts related to security, security issues and mitigation schemes in Windows OS, malware, network-based attack and security mechanisms, firewalls.

COURSE-SPECIFIC LEARNING OUTCOMES (CO)

CO1: To articulate the system vulnerabilities, exploitation, and defense mechanisms.

CO2: To examine the spyware, security issues and different types of attacks.

CO3: To do programming for system and mobile security.

Detailed Syllabus

Module 1 (Contact hours: 11)

Threats and Vulnerabilities of Windows, Threats and Vulnerabilities of Linux, Controls, Risk Management, Basics of Confidentiality, Integrity and Availability, Generic Security Policies, Security Mechanisms, Assurance, Prevention, and Detection, Security Issues in Windows Operating System (Windows XP, Windows Vista, Windows 7 and Windows 10), Linux Kernel TCP/IP Vulnerabilities, TCP SACKs (25), SACK Panic, Mitigation Schemes and Patches Developed for Each Security Issue, Boot Loader Security Issues, Grub Security Flaw.

Module 2 (Contact hours: 9)

Security in Broadcast Networks, Security in Centralised Networks, DDOS Attacks, Volume-based DDoS attacks, Protocol attacks, Application-layer DDoS attacks, Introduction to Malicious Softwares- Botnet, Logical Bombs, Grayware, Spyware, Adware, File Binders, Antivirus, Honeypot, Trapdoor, Latest Malwares Definitions and Their Defence Mechanisms. Android-based Malwares and their Security Patches.

Module 3 (Contact hours: 9)

Introduction to Network Security, Network-based Attacks, Active and Passive Attacks, Phishing and Its Types, Tailgating, Impersonation, Dumpster Diving, Shoulder Surfing, Hoax, Waterhole Attacks, Denial of Service Attacks and its Types, Man in the Middle Attack, Buffer Overflow Attack, Cross-Site Scripting, SQL Injection Attack, Privilege Escalation, Man in the Browser, Zero-Day Attack, SQL Injection Attack, Privilege Escalation, Man in the Browser.

Module 4 (Contact hours: 13)

Generic Security Software's, Windows Firewalls, Linux Firewalls, Access Control Lists of Firewalls, Types and Examples of Firewalls, Network-based Honeypots and Trapdoors, Virtual Private Network and its Implementation, Network Intrusion Detection System, Network Intrusion Prevention System, Router Security, Switch Security, Proxy Server and its Configurations, Load Balancers, IPv6 and IPv6 Security, Secure Forwarding in Overlay Networks.

STUDIO WORK / LABORATORY EXPERIMENTS:

The lab component is meant to analyse various network security-related concepts and deals with launching different types of attacks and creating a network blueprint of an organization.

TEXTBOOKS/LEARNING RESOURCES:

a) R. Bragg et al, Network Security: The Complete Reference (1st ed.), TMH Publications, 2017. ISBN 978-0070586710.

b) W. Stallings, Network Security Essentials: Applications and Standards (3rd ed.), Pearson, 2018. ISBN 978-0132380331.

REFERENCE BOOKS/LEARNING RESOURCES:

a) Joseph Migga Kizza, Guide to Computer Network Security (5th ed.), springer, 2020. ISBN 978303038140.

b) Jaydip Sen, Computer and Network Security (1st ed.), IntechOpen, 2020. ISBN 9781838808549.

CSET228 - Data Mining and Predictive Modelling

Course Type - Specialized Core – II L-T-P Format 3-0-2 Credits - 4

COURSE SUMMARY

This course exposes multiple techniques of understanding and analysing the data from a mathematical point of view. In addition, they will also use multiple predictive models to analyse the future trend. This will be done in a purely statistical manner.

COURSE-SPECIFIC LEARNING OUTCOMES (CO)

CO1: To articulate data preparation for data mining and analyzing based on pre-processing techniques.

CO2: To examine predictive analysis in various use cases.

CO3: To make use of exploratory data analysis to gain insights and prepare data for predictive modelling.

Detailed Syllabus

Module 1 (Contact hours: 11)

Purpose of Data mining, Procedures of Data Mining, Functionality of Data Mining, Knowledge data discovery process, Data and attribute type, Properties of data, Discrete and continuous attribute, Dataset types, Data quality measurement, Noise Analysis and its importance, Techniques of Data pre-processing, Aggregation, Sampling, Curse of dimensionality, Dimensionality reduction, Feature selection and generation, Discretization and vectorization, Binarization, Attribute transformation correlation, Association rule mining, Apriori algorithm, Rule generation, Pattern Mining in: Multilevel, Multidimensional Space Pattern Mining .

Module 2 (Contact hours: 7)

Rule-based reasoning, Memory-based reasoning, measuring data similarity, Similarity Metrics: Distance based measure, Information based measures, Set similarity measure, Jaccard Index, Sorenson Dice Coefficient, Model Selection Problem, Error Analysis, Case study, Startups in Data Analysis.

Module 3 (Contact hours: 10)

Outlier analysis in classification and clustering, Probabilistic models for clustering, Clustering high dimensional data: Subspace clustering, Projection Based clustering, Exploratory data analysis, Data summarization and visualization, Dataset exploration, Data Exploration Tools, Interactive Data Exploration, Predictive models, Design Principles, Parametric Models, Non-Parametric Models, ANOVA, Regression Analysis, Frequent Pattern Mining, Mining Closed and Max Patterns.

Module 4 (Contact hours: 14)

Linear discriminant analysis, Fisher discriminant analysis, Time series Model: ARMA, ARIMA, ARFIMA, Factor Analysis, Uncertainty quantification, Forward uncertainty propagation, Inverse uncertainty quantification, Non-Negative Matrix Factorization, Sequential Matrix Factorization. Exact Matrix Factorization, Expert Lecture from Industry, Recommendation System and Collaborative Filtering, Multidimensional Scaling, Mining Textual Data, Temporal mining, Spatial mining, Visual and audio data mining, Ubiquitous and invisible data mining- Privacy, Security, Social Impacts of data mining.

STUDIO WORK / LABORATORY EXPERIMENTS:

Data pre-processing and vectorization. Quality analysis of data. Feature selection and Ranking. Association rule mining and implementation of the Apriori algorithm. Data Similarity and set similarity. Error analysis and model selection. Frequent pattern mining and regression. Discriminant Analysis. Factor Analysis. Matrix Factorization. Recommendation System.

TEXTBOOKS/LEARNING RESOURCES:

a) Bruce Ratner, Statistical and Machine-Learning Data Mining: Techniques for Better Predictive Modeling and Analysis (3^{rd} ed.), Chapman and Hall/CRC, 2017. ISBN 978-1498797603.

b) Dursun Delen, Predictive Analytics (1^{st} ed.), missing, 2020. ISBN 9780136738516 .

REFERENCE BOOKS/LEARNING RESOURCES:

a) Mohammed J. Zaki and Wagner Meira, Jr, Data Minimg and Machine Learning (1^{st} ed.), missing, 2020. ISBN 9781108473989 .

CSET229 - Game Engine and Architecture

Course Type - Specialized Core – II L-T-P Format 3-0-2 Credits - 4

COURSE SUMMARY

This course is to design and build their own functional game using an existing game-engine along with practice of video game programming like a real game development team.

COURSE-SPECIFIC LEARNING OUTCOMES (CO)

CO1: To explain concepts of underlying game engines. Solve practical 3D game problems.

CO2: To Implement Game control systems for multiplayer games using Human Interface Devices and analyse the basic features of Physics for Games; Illustrate the collision detection systems used in games.

Detailed Syllabus

Module 1 (Contact hours: 10)

Game Engine, Engine Differences Across Genres, Runtime engine architecture: Base layer, Memory, layer, Utility, Module, Management, Gameplay, Build layers, Asset Pipeline, Pre-rendering pipeline, Full 3D pipeline, Game profiler, Tools for optimization of games, Types of profilers, Version control in game development, Best Practices in version control, Issues and solutions, Unity engine and features, Unreal engine and features, Amazon Lumberyard and features, cryEngine for VR and features.

Module 2 (Contact hours: 10)

Engine Levels, Game Engine VR Modes actors, Landscape, Water rendering, Lightening and Environment, Capsule shadow, light mobility, and types of lights, Volumetric clouds, Fog effects, Level streaming, Hierarchal level of details, Blueprints, Variables, functions, and flow control, Timelines and blueprint classes, Render To Texture Blueprint Toolset, Occlusion Culling, Render targets, Real time ray tracing, Frame Rendering tool, Programming and scripting, Actor Communication, Blueprint visual scripting, Gameplay Architecture, Gameplay classes and functions, Interfaces, Graphics Programming, Shader development, Threaded rendering, Mesh drawing pipeline, Slate UI Framework.

Module 3 (Contact hours: 10)

Human Interface Devices, Types of HID for games, Interfacing with HID, Devices in practice, features of popular devices, Gameplay Framework, Pawn, Controller, Camera, AI Entities in gaming, Behavior trees, Navigation system, Environment query system, Physics in gaming, UMG UI Designer, Networking and multiplayer, Animation Characters and Objects, Animation rigging toolset, Skeletal Mesh Animation System, Sound in gaming, Ambient Zones, Sound classes, Audio Engine and mixer, Sound Attenuation, Sound cue.

Module 4 (Contact hours: 12)

Importing and integration media objects, Asset management, Asset Build tool, Automation tool, Testing and Optimization, Performance and profiling, Animation System Architecture, Animation Blending, Animation Pipeline, Skinning and Matrix Palette Generation, Skeletons 496x, Animation controller Collision and Rigid Body Dynamics, Collision/Physics Middleware, The Collision Detection System, Rigid Body Dynamics, and Advance physics features.

STUDIO WORK / LABORATORY EXPERIMENTS:

The lab covers the practice of game engine software development methodologies for Iterative and agile development of a vertical slice of a game from end-to-end e.g. SCRUM. XP, Kanban, TDD etc. and to the design and implementation of subsystems for rendering, collision, physics animation.

TEXTBOOKS/LEARNING RESOURCES:

a) Jason Gregory, Jeff Lander and Matt Whiting, Game Engine Architecture (3rd ed.), A K Peters/CRC Press, 2018. ISBN 978-1138035454.

REFERENCE BOOKS/LEARNING RESOURCES:

a) Heather Maxwell Chandler, Game Production Handbook (3rd ed.), Jones & Bartlett, 2013. ISBN 978-1449688097.

CSET230 - DevOps Practices and Principles

Course Type - Specialized Core – II L-T-P Format 3-0-2 Credits - 4

COURSE SUMMARY

This course aims at software development lifecycle from requirements analysis through coding and production support. It consists of Culture of DevOps, Cloud Computing and Git, Configuration Management, Configuration Management & Puppet, Modules and Templates.

COURSE-SPECIFIC LEARNING OUTCOMES (CO)

CO1: To articulate DevOps engineering and DevOps terminologies to meet the business requirements.

CO2: To construct different applications of DevOps that can be used on different platforms.

CO3: To design and execute projects using different tools, modules, and templates.

Detailed Syllabus

Module 1 (Contact hours: 12)

Why DevOps? , Terminologies, DevOps Stakeholders and roles, DevOps Perspective: People, Process and Product, Building Teams, Boston Consultancy Group (BCG) Matrix, Three Horizons Model, Autonomous Team and its Pros and Cons, Autonomy Criteria, Decoupling Point for Autonomous Team, Case Study on Autonomous Team, DevOps and ISTM (IT Service and Management), Traditional Vs Agile DevOps, DevOps Phases with diagram representation, Introduction of Source and Version Control (SVC), Algorithms for SVC: Lock-Modify-Unlock and Copy-Modify -Merge, Continuous Integration, and Deployment (CICD), Software Craftsmanship, Software Containerization, System Provisioning, and Configuration Management, Testing and Test Automation.

Module 2 (Contact hours: 11)

Service-Oriented Architecture and Microservices, Build and Release Management, Virtual Systems, Hypervisor Cloud Computing, Need of Cloud Computing with Application, Delivery Models of Cloud Computing, Deployment Model of Cloud Computing, Git Desktop Usage, and Benefits, Connecting Git Desktop and Git Online, Git Version Control, and Types, Distributed Version Control Systems with Example, Centralized Version Control Systems with Example, Configuration Management, Chef Configuration, Workstation Setup, Configuration of the knife, Test Connection between Knife and Workstation, Organization Setup, Creation and Connection of Node to Organization, Operations on Node, Object and Search, Creation of Environments and Connecting them with Servers.

Module 3 (Contact hours: 6)

Puppet, Puppet Working and Architecture, Master and Agents in Puppets, Installing Puppet, Configuring Puppet Master and Agent, Connecting Agents, Puppet Language, Declarative Language, Resources, Files, Exec, Packages, Service in Puppets, Virtual Resources and Exported Resources.

Module 4 (Contact hours: 13)

DevOps Security, Issues in DevOps Security, DevOps Security Needs and Challenges, DevOps Risks Management, Strategies, Policies, Practices for DevOps Security, DevOps Security Tools, DevOps security over Clouds, Security Models, and their Use, Modules and Templates for DevOps, DevOps Manifests, Class Templates Static and Dynamic Content, ERB Module, NTP Module, SSH Module, Sudo Module, Install LAMP with pre-existing Modules.

STUDIO WORK / LABORATORY EXPERIMENTS:

The studio work of this course introduces to setup a home lab for DevOps project for DevOps tools Jenkins, Docker, Ansible, Kubernetes with hands-on experience of concepts taught in the lecture. The lab component also includes installation of Git Bash, Git Hub, Jenkins, Docker, Ansible, Kubernetes.

TEXTBOOKS/LEARNING RESOURCES:

a) Mark Reed, DevOps: The Ultimate Beginners Guide to Learn DevOps Step-by-Step (1st ed.), Publishing Factory, 2020. ISBN 978-1647710941.

b) Gene Kim, Kevin Behr, and George Spafford, The Phoenix Project (5th ed.), IT Revolution Press, 2019. ISBN 978- 1942788294.

REFERENCE BOOKS/LEARNING RESOURCES:

a) Gene Kim and John Willis, Beyond the Phoenix Project: The Origins and Evolution of DevOps (1st ed.), IT Revolution Press, 2018. ISBN 978-1942788256.

CSET231 - Programming Methodologies for Backend Development

Course Type - Specialized Core – II L-T-P Format 2-0-4 Credits - 4

COURSE SUMMARY

This course focuses on the server-side performance need and goals of web applications. It concentrates primarily on the different back-end/Server-side programming languages, which help in building web applications.

COURSE-SPECIFIC LEARNING OUTCOMES (CO)

CO1: To articulate the client-Server architecture and the basics of back-end technologies and tools.

CO2: To examine server-side programming languages like PHP and Node.js.

CO3: To connect with database with the help of server-side scripting-PHP and MySQL, Node.js and MongoDB.

Detailed Syllabus

Module 1 (Contact hours: 12)

Back-end technologies: servers, applications, and databases, Object-oriented based backend programming languages: PHP, Java, Node.js. MERN Stack: Mongo DB, Express.js, React.js, Node.js. functional-based backend programming languages (Mongo DB), Databases, MySQL, XHTML, PHP- a scripting language. PHP: server-side scripting and its syntax, working with variable and constant, PHP code blocks: arrays, strings, function. PHP error handling and Controlling program flow, looping, and branching, file handling, directories, PHP sessions and security, processing forms on server-side, cookies, Web services, exploring Java-based web technologies: JDBC, servlet, JSP. Web architecture model: client-server model, three-tier model, and service-oriented architecture (SOR), MVC architecture: model, view, controller.

Module 2 (Contact hours: 7)

Database programming: Operations and working on metadata, database configuration, connection to MySQL server, execution of MySQL queries, Node.js: Web Applications with Node.js, Core Node.js and Packages, Events Streams, Modules: export, object, class. Loading module from a separate folder, File Systems. MongoDB, Features of MongoDB, MongoDB Database Tier, using Node.js with MongoDB, using Node.js with MySQL, Server-side rendering.

Module 3 (Contact hours: 9)

REPL environment and commands, Backend frameworks: Django, Spring, Express. Docker Containers, GraphQL integration, API documentation: Swagger API, Postman (API testing), REST Principles for API, Containerize APIs with Docker, Serverless computing, DevOps toolkit.

STUDIO WORK / LABORATORY EXPERIMENTS:

Practical experience of server-side programming languages like PHP and Node.js by connecting with database with the help of server-side scripting-PHP and MySQL, Node.js, and MongoDB.

TEXTBOOKS/LEARNING RESOURCES:

a) David Herron, Node.js Web Development: Server-side web development made easy with Node 14 using practical examples (5th ed.), Packt Publishing Limited, 2020. ISBN 978-1838987572.

REFERENCE BOOKS/LEARNING RESOURCES:

a) Frank Zammetti, Modern Full-Stack Development: Using TypeScript, React, Node.js, Webpack, and Docker (1st ed.), Apress, 2020. ISBN 978-1484257371.

CSET232 - Design of Cloud Architectural Solutions

Course Type - Specialized Core – II L-T-P Format 3-0-2 Credits - 4

COURSE SUMMARY

Design of Cloud Architectural Solutions comes up with detailed learning and implementation of the inherent strengths of cloud computing such as global infrastructure, elasticity, automation, monitoring with security. This course aims to develop architectural solutions in a variety of use cases by leveraging the power of global cloud infrastructure and service.

COURSE-SPECIFIC LEARNING OUTCOMES (CO)

CO1: To learn stack of cloud architecture and design guidelines of the final product.

CO2: To build the cloud architectural blueprint for deployment.

CO3: To deploy and implement cloud architectural designed solution.

Detailed Syllabus

Module 1 (Contact hours: 11)

Cloud Architecture Job roles, Solution Architect Certifications, Cloud Solution Architect Competitions, Cloud Computing with characteristics, Core Cloud Services, Loose Coupling, Loose Coupling Strategies, Communication in coupled components, Decoupling of Infrastructure, Scalable Web Application deployment, Multi-Tenancy, Resource Provisioning, Traffic / Demand Monitoring, Capacity Planning and Prediction, Data centre management, Costing and Pricing of public cloud services, Cloud Governing Rules, Cloud Privacy issues, Design Principles for using cloud services, Design Principles for providing cloud services, Cloud architectural design principles, Principles of the Security, Principles of the Reliability Pillar, Principles of the Performance Efficiency Pillar, Cost Optimization, High-Availability Design Patterns, Criteria for cloud service selection, Multi-criteria Decision Making, Cloud Infrastructure security, IAM user, group, roles, and policy.

Module 2 (Contact hours: 11)

Cloud compliances, Authentication protocol implementation, Content Delivery, Domain Name System, Content delivery network, CDN Demo, Traditional network architecture, SDN architecture, SDN controller, OpenFlow protocol, Demo of SDN, Discussion of related Start-up, Traditional infrastructure, Virtual infrastructure, Converged infrastructure, IoT-Cloud Infrastructure, Autoscaling, Launch configuration, Autoscaling groups, ASG Balancing, Autoscaling policies, Load Balancer, Fault tolerance, High availability, Architectural need of LB, Application LB, Network LB, Listener, Target groups, Targets, Cross zone load balancing.

Module 3 (Contact hours: 9)

Case Study of a federated cloud, Mobile "backend" as a service (MBaaS), Serverless computing, Business Processing as a service (BPaaS), Virtualization Demo vs Containerization Demo, Virtualization and Containerization, Virtualized Operating Systems, Resource to OS-Hypervisor communication, Design of virtualized Database Clusters, Cloud Regional Backbone Networks, Network Troubleshooting, Transcoding and Serving Video Files Example, Stream Processing Example, Sensor Network Data Ingestion and Processing Example, Application Backend Example.

Module 4 (Contact hours: 11)

Cloud Monolith Applications, Cloud-Native Applications, Cloud Enterprise Applications, Challenges towards Hypergrowth of cloud-native applications, Evolution of Deployment, Current Cloud Computing System Designs, Modern Cloud Architecture Integration, Structure of a Cloud-Native Application, Characteristics of Cloud-Native Application, Bridging DevOps Culture with cloud architectural solution, Software development on cloud PaaS, Microservices, Microservice Architecture, Microservice Architecture v/s Monolith Architecture, Developing Cloud-Native Applications using Microservices, Flexibility and Scalability in microservice, Pros and Cons of Cloud Native Architecture, Microservice Use Cases, Service Decomposition, Microservice Demo, Strategies for Decomposing a System into Microservice, Migrating Monolithic Solutions to Microservice Ecosystem, Minimizing Dependency to Monolithic Solutions, Orchestration / Choreography of Micro-Services, Shared Data and Communication, Case Study on Cloud-Native Development, Advanced Research Topics.

STUDIO WORK / LABORATORY EXPERIMENTS:

Studio works include Hands-on experience on public industry cloud platform and will provide the cloud-based solution for the business problems. The activities that are mandatory to be completed includes troubleshooting the errors, accessing various AWS/Azure/GCP Support Options, using compute, storage, database, and networking concepts to improve the architecture's performance, identify design principles that can help cloud user to achieve performance efficiency, Evaluate the most important performance metrics for any applications.

TEXTBOOKS/LEARNING RESOURCES:

a) Scott Goessling and Kevin Jackson, Architecting Cloud Computing Solutions: Build cloud strategies that align technology and economics while effectively managing risk (1st ed.), Packt Publishing, 2018. ISBN 978-1788472425.

REFERENCE BOOKS/LEARNING RESOURCES:

a) Arshdeep Bahga and Vijay Madisetti, Cloud Computing Solutions Architect: A Hands-On Approach (1st ed.), Vpt, 2019. ISBN 978194997801X .

CSET233 - Machine Learning with Quantum Computing

Course Type - Specialized Core – II L-T-P Format 2-0-4 Credits - 4

COURSE SUMMARY

Quantum Machine Learning bridges the gap between abstract developments in quantum computing and the applied research on machine learning. It is the integration of quantum algorithms within machine learning programs and relies on the properties of quantum mechanics to compute problems that would be out of reach for classical computers.

COURSE-SPECIFIC LEARNING OUTCOMES (CO)

CO1: To explore and understand whether quantum computers can speed up the time it takes to train or evaluate a machine learning model.

CO2: To examine the interplay of ideas from quantum computing and machine learning.

CO3: To develop new quantum algorithms and explore the properties of quantum systems.

Detailed Syllabus

Module 1 (Contact hours: 8)

Course structure/handout assessment mechanism, why quantum machine learning? Tensor and Tensor Network, Hadamard Transformation, Quantum lambda calculus, Quantum Encode and Decode, Arbitrary State Generator, Quantum Density Operator or Density Matrix, Quantum Probability Theory, Quantum Stochastic Process, Quantum Mean, Variance and Envariance, Quantum Way of Linear Regression, Quadratic Unconstrained Binary Optimization, Quantum Topological Data Analysis, Quantum Bayesian Hypothesis, Haar Transform, Quantum Ridgelet Transform, Variational Quantum Eigen-solver.

Module 2 (Contact hours: 5)

Quantum Perceptron's, Quorn's, Quantum Auto Encoder, Quantum kernel estimation, Quantum Feature Map, Comparisons between QKE and classical kernels, Quantum Classifier, Variational Models, Quantum Approximate Optimization Algorithm, Quadratic Unconstrained Binary Optimization (QUBO), MaxCut problem.

Module 3 (Contact hours: 9)

Quantum Annealing, Photonic Implementation of Quantum Neural Network, Quantum Feed Forward Neural Network, Quantum Boltzman Neural Network, Quantum Neural Net Weight Storage, Quantum Upside Down Neural Net, Quantum Hamiltonian Neural Net, Quantum artificial neural network, Quantum Perceptron Network, Quantum approaches to CNNs, Quantum convolutional networks, Dissipative quantum neural networks, Barren plateaus in QNNs, Barren plateaus in deep QNNs, Noise-induced Barren plateaus, Superposition based Architectural Learning Algorithm, Quantum Hamiltonian Learning, Compressed Quantum Hamiltonian Learning, Quantum Classification using Principle Component Analysis, Quantum state classification with Bayesian methods, Quantum K-Nearest Neighbour, Quantum K-Means, Quantum Fuzzy C-Means.

Module 4 (Contact hours: 6)

Quantum Support Vector Machine, Quantum Ant Colony Optimization, Quantum-behaved Particle Swarm Optimization, Quantum Inspired Evolutionary Algorithm, Quantum Genetic Algorithm, Quantum Hidden Markov Models, Quantum Annealing Expectation-Maximization, Quantum Estimation Theory, Quantum Statistical Decision Theory, Quantum Hypothesis Testing, Quantum Chi-squared and Goodness of Fit Testing.

STUDIO WORK / LABORATORY EXPERIMENTS:

In-studio work, students will work on various machine learning models and classical and Quantum algorithms using quantum computers.

TEXTBOOKS/LEARNING RESOURCES:

a) Siddhartha Bhattacharyya, Sourav De, Indrajit Pan, Ashish Mani, Elizabeth Behrman and Susanta Chakraborti, Quantum Machine Learning (1st ed.), De Gruyter, 2020. ISBN 9783110670704.

b) Eric R. Johnston, Nic Harrigan and Mercedes Gimeno-Segovia, Programming Quantum Computers (1st ed.), O'Reilly, 2019. ISBN 9781492039659 .

REFERENCE BOOKS/LEARNING RESOURCES:

a) Klaus-Robert Müller, Kristof T. Schütt, Stefan Chmiela, Alexandre Tkatchenko, Koji Tsuda and O. Anatole von Lilienfeld, Machine Learning Meets Quantum Physics (1st ed.), Springer, 2020. ISBN 9783030402452 .

CSET234 - Drone Remote Sensing

Course Type - Specialized Core – II L-T-P Format 2-0-4 Credits - 4

COURSE SUMMARY

This course elaborates the prospects and challenges associated with UAV remote sensing. Understanding sensors, cameras, mapping, surveying, data acquisition, and aggregation. Utilize this basic knowledge to design a Multicopter drone for remote sensing applications.

COURSE-SPECIFIC LEARNING OUTCOMES (CO)

CO1: To articulate the fundamentals of UAV Remote Sensing.

CO2: To learn techniques and deployment scenarios of UAVs, alongside the challenges and potential solutions to data acquisition, mapping, and processing.

CO3: To construct multirotor, agricultural precision remote sensing UAV from scratch using various sensors, cameras, and flight controllers.

Detailed Syllabus

Module 1 (Contact hours: 8)

Sensing and Control; Application Scenarios and Challenges; Radio Controlled Vehicles to UAVs; Sensors: Sensor Types and Applications; Regulations and Guidelines; Attitude Estimation: Sensors, Complimentary Filters, Kalman Filters; UAV Imagery: Accuracy Requirements, Operational Restrictions.

Module 2 (Contact hours: 7)

Image Sensors; Image Sensor Selection; Computing Image; Imagery Collection; UAV Remote Sensing; Coverage Control; Georeferencing; Feature-Based Stitching; Position and Attitude Based Stitching; Consensus-Based Formation Control; Consensus Algorithms; Profile Measurement and Estimation; Wind Profile Measurement.

Module 3 (Contact hours: 7)

Mission Planning: Surveyed Geography, Alien Terrain; GPS Data Collection; Measurement; Mapping; UAV Path Planning; Path Refinement; Multi-UAV Path; Obstacle Avoidance; Focus of Expansion; Time to Collision; Semi-Autonomous Control; Radio Control; Automatic Control; Control Switching: Radio and Automatic.

Module 4 (Contact hours: 6)

Drone Sensing and Imagery: Riverine, Aquatic, Agriculture, Vegetation and Forest Ecosystem, Oil and Gas, Surveillance, Search and Rescue; UAS Ground Control; Human Machine Interface; Computer Telemetry; UAV Launch System; UAV Recovery System.

STUDIO WORK / LABORATORY EXPERIMENTS:

In this course, students will start with basic remote sensing deployments, develop an understanding of various sensors and cameras, and understand the principles of data acquisition and aggregation. components, and assembly precautions. Then finally design and develop a fully functional agricultural precision remote sensing Multicopter UAV/Drone.

TEXTBOOKS/LEARNING RESOURCES:

a) Serge A. Wich and Lian Pin Koh, Conservation Drones: Mapping and Monitoring Biodiversity (1st ed.), OUP Oxford, 2018. ISBN 0198787618.

REFERENCE BOOKS/LEARNING RESOURCES:

a) Amy Frazier and Kunwar Singh, Fundamentals of Capturing and Processing Drone Imagery and Data (1st ed.), CRC Press, 2021. ISBN 0367245728.

b) David R. Green, Billy J. Gregory and Alexander Karachok, Unmanned Aerial Remote Sensing (1st ed.), CRC Press, 2020. ISBN 1482246074.

CSET235 - Digital Bots Development

Course Type - Specialized Core – II L-T-P Format 2-0-4 Credits - 4

COURSE SUMMARY

To describe the use and functionality of Robotic Process Automation (RPA) in business prospects. Familiarize with the concepts related to RPA and how RPA is useful to automate business processes. The goal is to train the learners about RPA tool for developing applications that can automate the functionalities in an organization.

COURSE-SPECIFIC LEARNING OUTCOMES (CO)

CO1: To understand the underlying logic/structure related to robotic process automation.

CO2: To develop digital bots to automate the processes.

CO3: To implement the transactions in business processes using robotic process automation.

Detailed Syllabus

Module 1 (Contact hours: 6)

Course structure/handout Assessment mechanism, Automation 360, Using Discovery Bot for Process Documentation, Creating Discovery Bot Users, Use Case: Discovering the Customer ID Validation, Create Process, Record Process, Review Recordings and Create Opportunities, Convert Opportunity to Bot Prototype, Resilient Bots, Run time challenges, Handling Unpredictable Exceptions, Handling Predictable Exceptions, Modularity and reusability are the key to scalability, Creating scalable Bots, Universal recorder Excel automation.

Module 2 (Contact hours: 8)

Integration of Java Scripts with Automation Anywhere, Invoking of Java Script using automation 360, Integration of VB Script in Automation Anywhere, Code execution of VB Script using Automation Anywhere, Integration of Python script with Automation Anywhere, Running Python Script in Automation Anywhere. Automation anywhere Robotic Interface with back office business process use case, Creating form Creation of the Bots, Creation of Process, Automation anywhere Robotic Interface with front office business process use case, Creating form, Creation of the Bots, Creation of Process, Automation anywhere Robotic Interface process creation, Automation anywhere Robotic Interface task creation.

Module 3 (Contact hours: 8)

AAE Client in RPA Development, Smart recorder, Screen recorder, Web recorder, Editing a recorded Bot Workbench components, Leveraging workbench commands, Integrating RPA with Cognitive Solutions Understanding IQ Bot as a Cognitive solution, Utilizing the IQ Bot Portal, Following the IQ Bot Workflow Creating an Instance and Triggering Document analysis, Reading Instance Details, Editing an Instance.

Module 4 (Contact hours: 6)

Understanding RPA Analytics, Understanding Operational Analytics, Understanding Business Analytics Roles Generating Business Analytics Understanding the CoE Dashboard, Customizing and Comparing CoE Dashboards, Publishing a CoE Dashboard, Using the RPA Mobile App, Recording of tassk with AISense recorder, Edit a task recorded using AISense, Advanced Topics in AA, Use of variable anchor, Enable debug logs for AISense recorder.

STUDIO WORK / LABORATORY EXPERIMENTS:

In this course, students will learn task organization for automation using RPA Tools.

TEXTBOOKS/LEARNING RESOURCES:

a) Mullakara, Nandan, and Arun Kumar Asokan, Robotic process automation projects: Build real-world RPA solutions using UiPath and Automation Anywhere (1st ed.), Packt Publishing Ltd, 2020. ISBN 9781839217357.

b) Husan Mahey, Robotic Process Automation with Automation Anywhere: Techniques to fuel business productivity and intelligent automation using RPA (1st ed.), Packt Publishing Ltd, 2021. ISBN missing.

REFERENCE BOOKS/LEARNING RESOURCES:

a) Husan Mahey, Robotic Process Automation with Automation Anywhere (1st ed.), Packt Publishing Ltd, 2021. ISBN 9781839215650.

CSET236 - IoT Networks and Protocols

Course Type - Specialized Core – II L-T-P Format 2-0-4 Credits - 4

COURSE SUMMARY

This course articulates IoT device programming sensing and actuating technologies, IoT protocol stacks (Zigbee, 5G, NFC, MQTT, etc), examine the protocols IoT and WSN, and implement IoT solutions using sensors, actuators, and devices on Cloud-based IoT platforms.

COURSE-SPECIFIC LEARNING OUTCOMES (CO)

CO1: To articulate the protocols and standards designed for IoT.

CO2: To examine the basic protocols IoT and WSN.

CO3: To implement IoT solutions using sensors, actuators, and Devices.

Detailed Syllabus

Module 1 (Contact hours: 9)

Internet of Things, Sensors, Hysteresis Error, Quantization error, aliasing error, Actuator, Actuations, IoT Networking, Functional components of IoT, IoT dependencies, IoT service-oriented architecture, IoT categories, IoT gateways, Associated technologies, technical derivation from the regular web, IoT protocols, MQTT, SMQTT, CoAP, XMPP, IEEE802.15.4, AMQP.

Module 2 (Contact hours: 8)

6LoWPAN, LoRAWAN and Reference model, Integration of devices using LoRAWAN, Security in LoRAWAN, Zigbee, 3GPP, NB-IoT, Wireless HART, RFID, ISA100, Z-Wave, WSN, Cluster formation of sensors in WSN, Routing algorithms in WSN, UAV Network, UAV Navigation, 5G based communication among UAVs.

Module 3 (Contact hours: 6)

Machine to Machine communication, Architecture, and components for M2M, Standardization Effort for M2M, Interoperability in IoT, IoT Architecture for Interoperability, Industry Standards, SDN Origins and Evolution, Centralized and Distributed Control, Data Planes, Genesis of SDN, API in SDN, Control mechanism, Switch Deployment, Controller configuration software, SDN for WSNs, SDN-WISE Sensor Nodes, SDN-WISE Protocol Structure, Topology in SDN-WISE.

Module 4 (Contact hours: 5)

Software-Defined WSN Prototype, Situation-Aware Protocol Switching in SDN, Performance Analysis of Software Defined Networks, Sensor cloud, Architecture, Service life cycle model, Layered structure, Management issues in sensor cloud, Optimal composition of virtual sensors, Formation of virtual sensor group.

STUDIO WORK / LABORATORY EXPERIMENTS:

Studio works aim to provide hands-on experience of IoT devices to understand the communication part between the devices. Here, we are sending the data using different communication devices such as WiFi, Zigbee and Bluetooth. There are a set of experiments over web layer protocols such as MQTT, HTTP and CoAP. This course is also having projects to solve a real-life problem by using IoT networks and protocols.

TEXTBOOKS/LEARNING RESOURCES:

a) David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Robert Barton and Jerome Henry, IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things (1st ed.), Cisco Press, 2017. ISBN 1587144565.

b) S. Misra, A. Mukherjee and A. Roy, Introduction to IoT (1st ed.), Cambridge University Press, 2021. ISBN 1108959741.

REFERENCE BOOKS/LEARNING RESOURCES:

a) Perry Lea, IoT and Edge Computing for Architects: Implementing edge and IoT systems from sensors to clouds with communication systems, analytics, and security (2nd ed.), Packt Publishing, 2020. ISBN 183921480.

CSET237 - VR and 360 Video Production

Course Type - Specialized Core – II L-T-P Format 3-0-2 Credits - 4

COURSE SUMMARY

This course is designed to Understand basic VR design principles such as iteration, user testing, and documentation, Design foundations, this course will develop understanding to create an VR use flow. This will course will help students in using Tools and Gaming Platforms to build VR experiences and 360 Video production. It is to make next steps to start building an VR experience using latest tools and technologies.

COURSE-SPECIFIC LEARNING OUTCOMES (CO)

CO1: To learn how to create a VR use flow.

CO2: To use Tools and Gaming Platforms to build VR experiences and 360 Video production.

CO3: Make next steps to start building a VR experience using latest tools and technologies.

Detailed Syllabus

Module 1 (Contact hours: 10)

Intro to design: Understand basic VR design principles such as iteration, user testing, and documentation, Design foundations, ergonomics: Create a VR User Persona, design an ergonomic VR experience, VRText, VR Testing Scene.

Module 2 (Contact hours: 12)

Introduction to Immersive Media: principles of immersive 360 media such as spherical projection by critically reviewing professional 360 content, workflow of creating 360 Video, Metadata to 360 video and publishing, Analyze a variety of existing 360 video content, Analyze footage for the various techniques, tips, and tricks, Scripting and Planning: Create a script for 360 content, Plan a 360 shoot, Edit scripts to match equipment and budget, 360 cameras: Exploration of existing 360 cameras, Differentiate between monoscopic and stereoscopic capture, different equipment and techniques used for audio capture, lighting techniques for 360 video.

Module 3 (Contact hours: 12)

Stitching: Stitch 360 video, Blend, weight, synchronize, and stabilize footage, Editing Basics: Edit footage, Create transitions in 360 video, Advanced Editing: Color correct 360 footage, Add points of interest, Advanced Stitching: edit and refine stitching using control points and masks, Blending.

Module 4 (Contact hours: 8)

Spatial Audio: Cut spatialized audio, Place sound in 3D space, Interactivity with Gaming platform: Create an interactive 360 experience using Gaming platform's video player, player controls and branching storylines using a custom 360 video player.

STUDIO WORK / LABORATORY EXPERIMENTS:

The experiments to create 360 video with metadata, capture and use 3D audio to 360 videos. Adding interaction control, adding transition in 360 videos, 360 video testing and performance Evaluation.

TEXTBOOKS/LEARNING RESOURCES:

a) Jesse Glover and Jonathan Linowes, Complete Virtual Reality and Augmented Reality Development with Unity (1st ed.), Packt Publishing Limited, 2019. ISBN 9781838644865.

b) Doug A Bowman, Ernest Kuijff, Joseph J LaViola, Jr and Ivan Poupyrev, 3D User Interfaces, Theory and Practice (2nd ed.), Pearson Education, USA, 2017. ISBN 978-0134034324.

REFERENCE BOOKS/LEARNING RESOURCES:

a) Erin Pangilinan, Steve Lukas and Vasanth Mohan, Creating Augmented and Virtual Realities (1st ed.), O'Reilly Publishers, 2019. ISBN 978-1492044192.

b) Dong Hwa Choi, Judi Estes and Amber Dailey-Hebert, Current and Prospective Applications of Virtual Reality in Higher Education (1st ed.), IGI Global, 2020. ISBN 9781799849619.

CSET238 - Product Design Principles and Practices

Course Type - Specialized Core – I L-T-P Format 3-0-2 Credits - 4

COURSE SUMMARY

This course covers the concepts of product design and the development process. It provides the set of product development methods that are delivered in a way that can be put into immediate practice on development projects.

COURSE-SPECIFIC LEARNING OUTCOMES (CO)

CO1: To articulate the concepts of product design and development process.

CO2: To apply of product design and development in industrial applications.

CO3: To implement and apply design or redesign of the product.

Detailed Syllabus

Module 1 (Contact hours: 10)

Characteristics of Successful Product, Product Design and Development, Duration and Cost of Product Development, Challenges of Product Development, Structured Methods, Industrial Examples, Organizational Realities, Product Development Process and Organization, Product Development Process, Concept Development: The Front-End Process, Adapting the Generic Product Development Process, Technology-Push Products, Platform Products, Process-Intensive Products, Customized Products, High-Risk Products, Quick-Build Products Digital Products, Product-Service Systems, Complex Systems, Product Development Process Flows, Tyco Product Development Process.

Module 2 (Contact hours: 10)

Product Development Organizations, Choosing an Organizational Structure, Distributed Product Development Teams, Tyco Product Development, Opportunity Identification, What Is an Opportunity? Types of Opportunities Tournament Structure of Opportunity Identification, Effective Opportunity, Tournaments, Opportunity, Identification Process, Establish a Charter, Generate and Sense Many Opportunities, Techniques for Generating, Opportunities, Screen Opportunities, Develop Promising Opportunities, Select Exceptional Opportunities, Reflect on the Results and the Product Planning, The Product Planning Process, Identify Opportunities, Evaluate and Prioritize Projects, Competitive Strategy, Market Segmentation, Technological Trajectories, Product Platform Planning, Technology Road mapping, Evaluating Fundamentally New Product Opportunities, Balancing the Portfolio, Allocate Resources and Plan Timing, Resource Allocation, Project Timing, The Product Plan, Complete Pre-Project Planning, Mission Statements, Assumptions and Constraints, Staffing and Other Pre-Project Planning, Reflect on the Results and the Process.

Module 3 (Contact hours: 10)

Identifying Customer Needs, Importance of Latent Needs, Process of Identifying Customer Needs, Gather Raw Data from Customers, Choosing Customers, The Art of Eliciting Customer Needs Data, Documenting Interactions with Customers, Interpret Raw Data in Terms of Customer Needs, Organize the Needs into a Hierarchy, Establish the Relative Importance of the Needs, Reflect on the Results and the Process, Product Specifications, What Are Specifications? When Are Specifications Established?, Establishing Target Specifications, Prepare the List of Metrics, Collect Competitive Benchmarking, Set Ideal and Marginally Acceptable, Target Values, Reflect on the, Results and the Process, Setting the Final Specifications, Developing Technical Models of the Product, Developing a Cost Model of the Product, Refine the Specifications, Making Trade-Offs Where Necessary, Flow Down the Specifications as Appropriate, Reflect on the Results and the Process, Target Costing, Concept Generation, The Activity of Concept Generation, Structured Approaches Reduce the Likelihood of Costly Problems.

Module 4 (Contact hours: 12)

Clarify the Problem, Decomposing a Complex Problem into Simpler Subproblems, Focus Initial Efforts on the Critical Subproblems, Search Externally, Interview Lead Users, Consult Experts, Search Patents, Search, Published Literature, Benchmark-Related Products, Generating Solution Concepts, Explore Systematically, Concept Classification Tree, Concept Combination Table, Managing the Exploration Process, Reflecting, Solutions and the Process, Concept Selection, Development Process, Use Method for Choosing a Concept, Structured Method Offers Several Benefits, Overview of Methodology, Concept Screening, Prepare the Selection Matrix, Rate the Concepts, Rank the Concepts, Concept-Screening Matrix Example, Reflect on the Results and the Process, Concept Scoring, Combine and Improve the Concepts, Reflect on the Results and the Process Caveats, Concept-Scoring Matrix Example, Concept Testing, Define the Purpose of the Concept Test, Choose a Survey Population, Choose a Survey Format, Communicate the Concept, Matching the Survey Format with the Means of Communicating the Concept, Issues in Communicating the Concept, Measure Customer Response, Interpret the Results, Reflect on the Results and the Process, Estimating Market Size.

STUDIO WORK / LABORATORY EXPERIMENTS:

In the studio, students will learn the application of structured methods to product development and get a detailed insight into the study and improvement of the development process. The students will be guided to create their development methods, uniquely suited to their personalities, talents, and company environments.

TEXTBOOKS/LEARNING RESOURCES:

a) Karl T. Ulrich, Steven D. Eppinger and Maria C. Yang., Product Design and Development (7th ed.), McGraw Hill, 2020. ISBN 9390113237.

REFERENCE BOOKS/LEARNING RESOURCES:

a) Ali Jamnia, Introduction to Product Design and Development for Engineers (1st ed.), Taylor and Francis, 2018. ISBN 9781138554214.

CSET239 - Product Design Architecture and Delivery

Course Type - Specialized Core – II L-T-P Format 3-0-2 Credits - 4

COURSE SUMMARY

This course combines creativity and practicality to bridge the gap between product design and design engineering. The student will learn the design architecture of the product as its delivery and operations.

COURSE-SPECIFIC LEARNING OUTCOMES (CO)

CO1: To articulate product design architecture and communication in designing.

CO2: To examine all mandatory models by developing the prototype.

CO3: To make use of development methods that can be exploited in the delivery of the product.

Detailed Syllabus

Module 1 (Contact hours: 10)

What Is Product Architecture? Types of Modularity, Implications of the Architecture, Product Change, Product Variety, Component Standardization, Product Performance, Manufacturability, Product Development Management, Establishing the Architecture, Create a Schematic of the Product, Cluster the Elements of the Schematic, Create a Rough Geometric Layout, Identify the Fundamental and Incidental, Platform Planning, Differentiation Plan, Commonality Plan, Defining Secondary Systems, Creating Detailed Interface Specifications, Industrial Design, Assessing the Need for Industrial Design ,Expenditures for Industrial Design, Industrial Design to a Product, User Experience Needs ,Aesthetic Needs, The Impact of Industrial Design, Is Industrial Design Worth the Investment? How Does Industrial Design Establish a Corporate Identity? The Industrial Design Process, Investigation of Customer Needs, Conceptualization, Preliminary Refinement, Further Refinement and Final Concept Selection, Control Drawings or Models, Coordination with Engineering, Manufacturing and External Vendors, Management of the Industrial Design Process, Timing of Industrial Design Involvement, Assessing the Quality of Industrial Design Usability, Emotional Appeal, Ability to Maintain and Repair the Product, Appropriate Use of Resources, Product Differentiation.

Module 2 (Contact hours: 10)

Design for Environment, Environmental Impacts, Herman Miller's Journey toward Design for Environment, The Design for Environment Process, Design for Environment Guidelines, Design for Manufacturing and Supply Chain, Requires a Cross-Functional, Overview of the DFM Method, Strategic Sourcing Decisions, Estimate the Manufacturing Costs, Cost of Goods, Fixed Costs versus Variable Costs, The Bill of Materials, Estimating the Costs of Standard Components, Estimating the Costs of Custom Components, Estimating the Costs of Assembly, Estimating the Overhead Costs, Reduce the Costs of Components, Understand the Process Constraints and Cost Drivers, Redesign Components to Eliminate Processing Steps, Choose the Appropriate Economic Scale for the Part Process, Standardize Components, Adhere to the "Black Box" Component, Procurement, Reduce the Costs of Assembly and Integrate Parts, Maximize Ease of Assembly, Consider Customer Assembly, Reduce the Costs of Supporting, Production, Minimize Systemic Complexity, Error Proofing, Reduce the Costs of Logistics, guidelines for minimizing the volume, Materials Costs, Component Manufacturing Costs, Assembly Costs, Cost Structures, Impact of DFM Decisions: The Impact of DFM on Development Time, The Impact of DFM on Development Cost, The Impact of DFM on Product Quality, The Impact of DFM on the Larger Enterprise.

Module 3 (Contact hours: 14)

Prototyping, Understanding Prototypes, Types of Prototypes, What Are Prototypes Used For? Principles of Prototyping, Analytical Prototypes, Physical Prototypes, Prototyping Technologies, 3D Printing, Planning for Prototypes, Define the Purpose of the Prototype, Establish the Level of, Approximation of the Prototype, Outline an Experimental Plan, Create a Schedule for Procurement, Construction, and Testing, Planning Milestone Prototypes, Robust Design, Design of Experiments , Robust Design Process, Identify Control Factors, Noise Factors, and Performance Metrics, Formulate an Objective Function, Develop the Experimental Plan, Experimental Designs, Testing Noise Factors, Run the Experiment, Conduct the Analysis, Computing the Objective Function, Computing Factor Effects by Analysis of Means, Select and Confirm Factor Setpoints, Reflect and Repeat Caveats, Patents and Intellectual Property, What Is Intellectual Property? Overview of Patents, Utility Patents, Preparing a Disclosure, formulate a Strategy and Plan, Timing of Patent Applications, Type of Application, Scope of Application, Study Prior Inventions, Outline Claims, Write the Description of the Invention Figures, Writing the Detailed Description, Defensive Disclosure, Refine Claims, Writing the Claims, Guidelines for Crafting Claims, Pursue Application, Reflect on the Results and the Process, Advice to Individual Inventors, Service Design Product-Service Systems, In What Ways Are Services and Products Different? Service Design Process, The Service, Development at Zipcar, Service Process Flow Diagram, Subsequent Refinement, Downstream Development Activities in Services, prototyping a Service, Growing Services, Continuous Improvement.

Module 4 (Contact hours: 8)

Product Development Economics, Elements of Economic Analysis, Quantitative Analysis, Qualitative Analysis, Economic Analysis Process, Build a Base-Case Financial Model, Estimate the Timing and Magnitude of Future Cash Inflows and Outflows, Perform Sensitivity Analysis, Development Cost Example, Development Time Example, Understanding Uncertainties, Use Sensitivity Analysis to Understand, Trade-Offs, Potential Interactions, Trade-Off Rules, Limitations of Quantitative Analysis Projects Interact with the Firm, the Market, and the Macro Environment, Carrying Out Qualitative Analysis, Time Value of Money and the Net Present Value Technique, Modeling Uncertain Cash Flows Using Net, Present Value Analysis, Project Management, The Design Structure Matrix, Gantt Charts PERT Charts, Critical Path, Baseline Project Planning, Contract Book, Project Task List, Team Staffing and Organization, Project Schedule, Project Budget, Project Risk Plan, Modifying the Baseline Plan, Accelerating Projects, Project Execution, Coordination Mechanisms, Assessing Project Status Corrective Actions, Postmortem Project Evaluation.

STUDIO WORK / LABORATORY EXPERIMENTS: