COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE

PROGRAMME STRUCTURE & SYLLABUS
[2021 ADMISSIONS ONWARDS]

M.TECH. COMPUTER SCIENCE & ENGINEERING WITH SPECIALIZATION IN SOFTWARE ENGINEERING

SYLLABUS
FOR
OUTCOME BASED EDUCATION

In **Master of Technology (M.Tech.)** Degree Program
in **COMPUTER SCIENCE & ENGINEERING WITH SPECIALIZATION IN SOFTWARE ENGINEERING**

For the student admissions starting from the academic year 2021-22

Program Outcomes (PO) for the M.Tech. Program in COMPUTER SCIENCE & ENGINEERING WITH SPECIALIZATION IN SOFTWARE ENGINEERING

After the completion of M.Tech. programme, the students will be able to:

PO1: Elicit deeper and current knowledge through research/exploration leading to development work with a motivation to solve practical problems.

PO2: Communicate effectively through well-written technical documentation as well as audio-visual presentations.

PO3: Recognize the importance of entrepreneurship and innovation to create value and wealth.

PO4: Acquire mastery in the topic of study at an exceedingly higher level.

DEPARTMENT OF COMPUTER SCIENCE
PROGRAMME STRUCTURE AND SYLLABUS (2021 ADMISSIONS)

M. TECH. COMPUTER SCIENCE & ENGINEERING WITH SPECIALIZATION IN SOFTWARE ENGINEERING

Semester - I

Sl. No.	Course code	Course Title	Core/ Elective	Credits	Lec.	Lab/ Tutorial	Marks
1	21-480-0101	Mathematical Concepts for Computer Science	C	4	4	2	100
2	21-480-0102	Machine Learning Algorithms	C	4	4	3	100
3	21-480-0103	Design and Analysis of Algorithms	C	4	4	3	100
5	-	Elective I	E	3	4	1	100
6	-	Elective II	E	3	4	1	100
Total for Semester I				18	20	10	500

Electives

21-480-0104: Artificial Intelligence

21-480-0105: Human Computer Interaction

21-480-0106: Information Retrieval and Web search

21-480-0107: Functional Programming

21-480-0108: Software Quality Management

21-480-0109: Design and security of Internet of Things

21-480-0110: Quantum Computing

Semester - II

Sl. No.	Course code	Course Title	Core/ Elective	Credits	Lec.	Lab/ Tutorial	Marks
1	21-480-0201	Big Data Analytics	C	4	4	2	100
2	21-480-0202	Agile Software Engineering	C	4	4	2	100
3	21-480-0203	Seminar	C	1	0	3	50
4	-	Elective III	E	3	4	1	100
5	-	Elective IV	E	3	4	1	100
6	-	Elective V	E	3	4	1	100
Total for Semester II				18	20	10	550

Electives

21-480-0204: Software Architecture

21-480-0205: Software Agent Systems

21-480-0206: Enterprise Application Integration and Business Process Management

21-480-0207: Advanced Data Mining

21-480-0208: Fuzzy Set Theory: Foundations and Applications

21-480-0209: Complex Networks: Theory and Applications

21-480-0210: Advances in Databases

21-480-0211: Blockchain Technology

Semester - III

Sl. No.	Course code	Course Title	Core/ Elective	Credits	Lec.	Lab/ Tutorial	Marks
1	21-480-0301	Elective VI	E	2	0	10	50
2	21-480-0302	Project & Viva Voce	C	16	0	20	350
Total for Semester III				18	0	30	400

Semester - IV

Sl. No.	Course code	Course Title	Core/ Elective	Credits	Lec.	Lab/ Tutorial	Marks
1	21-480-0401	Project & Viva Voce	C	18	0	30	500

Total credits for Degree: **72**

21-480-0101: Mathematical Concepts for Computer Science

Core/Elective: **Core** Semester: **1** Credits: **4**

Course Description

This course introduces the study of mathematical structures that are fundamentally discrete in nature. The course is intended to cover the main aspects which are useful in studying, describing and modeling of objects and problems in the context of computer algorithms and programming languages.

Course Outcomes (CO)

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

CO1: Analyse the different methods for proving the correctness of the theorems and problems.
CO2: Understand the basic concepts of number theory.
CO3: Understand the basic aspects of graph theory.
CO4: Evaluate the performance of various graph-based algorithms.
CO5: Understand the fundamentals of probability theory.
CO6: Apply various probability density functions and its moments to solve problems.

Mapping with Program Outcomes

CO1: PO1
CO2: PO1
CO3: PO1
CO4: PO1, PO4
CO5: PO1
CO6: PO1, PO4

Course Content

1. Introduction – proofs – propositions – predicates and quantifiers – truth tables – first order logic – satisfiability – pattern of proof – proofs by cases – proof of an implication – proof by contradiction – proving iff – sets – proving set equations – Russell's paradox – well-ordering principle – induction – invariants – strong induction – structural induction

2. Sums – arithmetic, geometric and power sums – approximating sums – harmonic sums – products – Stirling's approximation for finding factorial-Pigeon hole principle – parity – number theory – divisibility – gcd – Euclid's algorithm – primes.

3. Graph theory – simple graphs – isomorphism – subgraphs – weighted graphs – matching problems – stable marriage problem – graph coloring – paths and walks – shortest paths – connectivity – Eulerian and Hamiltonian tours – travelling salesman problem – trees – spanning trees – planar graphs – Euler's formula – directed graphs – strong connectivity – relations – binary relations – surjective and injective relations symmetry, transitivity, reflexivity, equivalence of relations – posets and dags – topological sort.

4. Probability – events and probability spaces – conditional probability – tree diagrams for computing probability – sum and product rules of probability – A posteriori probabilities – identities of conditional probability – independence – mutual independence – birthday paradox – random variables – indicator random variables.

5. Probability distribution functions – Bernoulli, Uniform, Binomial, Poisson, Normal distributions – Expectation – linearity of expectations – sums of indicator random variables – expectation of products – variance and standard deviation of random variables – Markov's and Chebyshev's theorems – Bounds for the sums of random variables.

References

1. Eric Lehman, F Thomson Leighton, Albert R Meyer, Mathematics for Computer Science, 1e, MIT, 2010.
2. Susanna S. Epp, Discrete Mathematics with Applications, 4e, Brooks Cole, 2010.
3. Gary Chartrand, Ping Zhang, A First Course in Graph Theory, 1e, Dover Publications, 2012.
4. Michael Sipser, Introduction to Theory of Computation, 3e, Cengage, 2014.
5. Sheldon Ross, A First Course in Probability, 9e, Pearson, 2013.
6. Tom Leighton, and Marten Dijk. 6.042J Mathematics for Computer Science.Fall 2010. Massachusetts Institute of Technology: MIT OpenCourseWare, https://ocw.mit.edu.
7. John Tsitsiklis. 6.041SC Probabilistic Systems Analysis and Applied Probability. Fall 2013. Massachusetts Institute of Technology: MIT OpenCourseWare. https://ocw.mit.edu
8. Igor Pak. 18.315 Combinatorial Theory: Introduction to Graph Theory, Extremal and Enumerative Combinatorics. Spring 2005. Massachusetts Institute of Technology: MIT OpenCourseWare, https://ocw.mit.edu
9. Albert Meyer. 6.844 Computability Theory of and with Scheme. Spring 2003. Massachusetts Institute of Technology: MIT OpenCourseWare, https://ocw.mit.edu.
10. Shai Simonson , Theory of Computation, http://www.aduni.org/courses/theory/

21-480-0102: Machine Learning Algorithms

Core/Elective: **Core** Semester: **1** Credits: **4**

Course Description

Machine learning is programming computers to optimize a performance criterion using example data or past experience. This course is to discuss many methods that have their bases in different fields: statistics, pattern recognition, neural networks, artificial intelligence, signal processing, control, and data mining. Major focus of the course is on the algorithms of machine learning to help students to get a handle on the ideas, and to master the relevant mathematics and statistics as well as the necessary programming and experimentation.

Course Outcomes (CO)

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

CO1: Demonstrate strength and weakness of Machine Learning approaches.

CO2: Appreciate the underlying mathematical relationships within and across algorithms and different paradigms of Machine Learning.

CO3: Utilize dimensionality reduction techniques for feature selection.

CO4: Examine methods for model building and fine tuning.

CO5: Experiment with Machine learning tools.

CO6: Apply Machine Learning algorithms to many real world problems.

Mapping with Program Outcomes

CO1: PO1

CO2: PO1

CO3: PO1, PO4

CO4: PO1, PO4

CO5: PO1, PO4

CO6: PO1, PO4

Course Content

1. Machine Learning – Examples of Machine Learning applications – Supervised Learning: Learning a class from examples – Learning multiple classes – Regression – Model selection – Bayesian Decision Theory: Classification – Discriminant functions – Association rules – Parametric methods: MLE – Bayes estimator – Parametric classification – Tuning model complexity

2. Multivariate Methods – Classification – Regression – Dimensionality reduction: LDA – PCA – Factor Analysis – ICA – Locally Linear Embedding – MDS- Probabilistic Learning: Gaussian Mixture Models- EM algorithm- Nearest Neighbor Methods – Distance Measures

3. Support Vector Machines: Optimal separation – Kernels – SVM algorithm – Extensions to SVM – Optimization and Search: Least-squares optimization – conjugate gradients – Search: Search techniques – Exploitation and exploration – Simulated annealing

4. Learning with trees: Decision trees – CART – Ensemble Learning: Boosting – Bagging – Random Forests – Unsupervised Learning: K-Means algorithm – Vector quantization – SOM algorithm – Markov Chain Monte Carlo Methods

5. Graphical Models: Bayesian Networks – Markov Random Fields – HMMS – Tracking Methods – Deep Belief Networks: Hopfield Network – Boltzmann Machine – RBM – Deep Learning

References

1. Ethem Alpaydin, Introduction to Machine Learning, 3e, MIT Press, 2014
2. Tom M. Mitchell, Machine Learning, McGraw Hill Education; 1e, 2017
3. Stephen Marsland, Machine Learning, An Algorithmic Perspective, 2e, CRC Press, 2015
4. Giuseppe Bonaccorso, Machine Learning Algorithms, 1e, Packt Publishing Limited, 2017
5. Ethem Alpaydin, Machine Learning- The New AI, MIT Press, 1e, 2016
6. Andrew Ng, Machine Learning Yearning, ATG AI (Draft version), 1e, 2018
7. Rohit Singh, Tommi Jaakkola, and Ali Mohammad. *6.867 Machine Learning.* Fall 2006. Massachusetts Institute of Technology: MIT OpenCourseWare, https://ocw.mit.edu
8. Andrew Ng, https://www.coursera.org/learn/machine-learning

21-480-0103: Design and Analysis of Algorithms

Core/Elective: **Core** Semester: **1** Credits: **4**

Course Description

The course covers the foundational algorithms in depth. The course helps in understanding the working and complexity of the fundamental algorithms and to develop the ability to design algorithms to attack new problems.

Course Outcomes (CO)

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

CO1: Understand the basic concepts of design and analysis of fundamental algorithms.
CO2: Develop the ability to design algorithms to attack new problems.
CO3: Prove the correctness of algorithms.
CO4: Develop the ability to analyze the complexity of algorithms.
CO5: Understand Complexity classes, concepts of P and NP problems

Mapping with Program Outcomes

CO1: PO1
CO2: PO1, PO4
CO3: PO2, PO4
CO4: PO1, PO2
CO5: PO1, PO4

Course Content

1. Introduction to design and analysis of algorithms, models of computation, correctness proofs, insertion sort, computational complexity, Master theorem , proof of Master theorem, merge sort, heaps, heap sort, binary search, binary search trees.

2. Graph algorithms, BFS and DFS, Dijkstra's algorithm, proof of correctness of Dijkstra's algorithm, Complexity analysis of Dijkstra's algorithm , Negative weight edges and cycles , Bellman-Ford algorithm, proof of correctness and complexity of Bellman-Ford, All pairs shortest paths, Floyd-Warshall algorithm, proof of correctness and complexity, Minimum Spanning Trees , Prim's algorithm, Cut property, Kruskal's algorithm, proof of correctness and complexity analysis of Kruskal's Algorithm, Maximum-Flow networks, Ford-Fulkerson method, proof of correctness and complexity, Edmonds-Karp algorithm

3 .Probability review, Experiments, outcomes, events, Random variables, Expectation, Linearity of Expectation, Indicator Random Variables, Hiring Problem, Quicksort , Best case and Worst case complexity, Randomized Quicksort , Average case complexity , Hashing, Chaining, Open Addressing, Universal Hashing, Perfect Hashing , Analysis of hashing operations

4. Dynamic Programming , Rod-cutting problem, Recursive formulation, Bottom-up reformulation of

recursive algorithms, Optimal Substructure Property, Matrix chain multiplication, Complexity of dynamic programming algorithms, Sequence Alignment , Longest common subsequence, Greedy algorithms, Optimal substructure and greedy-choice properties , 0-1 and fractional Knapsack problems, Huffman coding

5. P vs NP, NP Hardness, Reductions, Travelling Salesman Problem, NP-Completeness, SAT, 2-SATand 3-SAT, Vertex Cover

References

1. Thomas H. Cormen et al, Introduction to Algorithms, 3e, MIT Press, 2009.
2. Jon Kleinberg, Eva Tardos, Algorithm Design, 2e, Pearson, 2015.
3. Robert Sedgewick, Kevin Wayne, Algorithms, 4e, AW Professional, 2011.
4. Steven S. Skiena, The Algorithm Design Manual, 2e, Springer, 2011.

21-480-0104: Artificial Intelligence

Core/Elective: **Elective** Semester: **1** Credits: **3**

Course Description

Artificial Intelligence (AI) is a field that has a long history but is still constantly and actively growing and changing. In this course basics of modern AI as well as some of the representative applications of AI along with huge possibilities in the field of AI, which continues to expand human capability beyond our imagination are taught.

Course Outcomes (CO)

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

CO1: Explain what constitutes "Artificial" Intelligence and how to identify systems with Artificial Intelligence.
CO2: Explain how Artificial Intelligence enables capabilities that are beyond conventional technology
CO3: Use classical Artificial Intelligence techniques, such as search algorithms, minimax algorithms, neural networks, tracking, robot localisation.
CO4: Apply Artificial Intelligence techniques for problem solving..
CO5: Explain the limitations of current Artificial Intelligence techniques

Mapping with Program Outcomes

CO1: PO1
CO2: PO1
CO3: PO2, PO4
CO4: PO1, PO4
CO5: PO1, PO4

Course Content

1. Introduction: Overview and Historical Perspective-Intelligent Agents-Problem Solving by searching-State Space Search: Depth First Search, Breadth First Search, DFID- Informed search & exploration-Heuristic Search- Best First Search-Hill Climbing-Beam Search-Tabu Search-Randomized Search:Simulated Annealing, Genetic Algorithms- Constraint Satisfaction Problems.

2. Finding Optimal Paths: Branch and Bound, A*, IDA*, Divide and Conquer approaches-Beam Stack Search-Problem Decomposition: Goal Trees, AO*, Rule Based Systems -Game Playing: Minimax Algorithm, Alpha-Beta Algorithm, SSS*.

3. Knowledge and reasoning: Propositional Logic- First Order Logic-Soundness and Completeness-Forward and Backward chaining-Resolution-semantic networks-Handling uncertain knowledge – Probabilistic Reasoning – making simple and complex decisions.

4. Planning : Planning problems - Planning with state space search - Partial order planning - Planning Graphs – Planning with Propositional logic-Hierarchical planning - Multi agent planning.

5. Learning: Forms of learning - Inductive learning - Learning decision trees - Explanation based learning - Statistical learning - Instance based learning – Neural networks-Reinforcement learning.

References

1. Stuart Russell and Peter Norvig. Artificial Intelligence: A Modern Approach, 3e, Prentice Hall, 2009
2. Deepak Khemani. A First Course in Artificial Intelligence, 1e, McGraw Hill Education, 2017
3. Stefan Edelkamp and Stefan Schroedl. Heuristic Search: Theory and Applications, 1e, Morgan Kaufmann, 2011
4. Zbigniew Michalewicz and David B. Fogel. How to Solve It: Modern Heuristics. Springer; 2e, 2004
5. Elaine Rich and Kevin Knight. Artificial Intelligence, 3e, Tata McGraw Hill, 2017
6. Patrick Henry Winston. Artificial Intelligence, 1e, Pearson, 2002

21-480-0105: Human Computer Interaction

Core/Elective: **Elective** Semester: **1** Credits: **3**

Course Description

Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and the major phenomena surrounding them. It is often regarded as the intersection of Computer Science and behavioural science. HCI is also sometimes referred to as man–machine interaction (MMI) or computer– human interaction (CHI).

Course Outcomes (CO)

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

CO1: Explain the capabilities of both humans and computers from the viewpoint of human information processing.
CO2: Describe typical human–computer interaction (HCI) models and styles, as well as various historic HCI paradigms
CO3: Apply an interactive design process and universal design principles to designing HCI systems.
CO4: Describe and use HCI design principles, standards and guidelines
CO5: Analyze and identify user models, user support, socio-organizational issues, and stakeholder requirements of HCI systems

Mapping with Program Outcomes

CO1: PO1
CO2: PO1
CO3: PO2, PO4
CO4: PO1, PO4
CO5: PO1, PO4

Course Content

1. Overview of HCI – Mental models – Cognitive architecture – task loading and stress in HCI – Human error identification.

2. Input technologies – sensor and recognition based input – visual displays – Haptic interfaces – Non speech auditory output – network based interactions.

3. Designing human computer interaction – Visual design principles – intercultural user interface designs – Conversational speech interface – multimodal interface – adaptive interfaces and agents

– Tangible user interfaces – Information visualization – Human centered designs of DSS – Online communities – Visual environment.

4. Domain specific design – HCI in healthcare – games – older adults – kids – Physical disabilities – Perpetual Impairments – Deaf and Hard of Learning users.

5. Developments process – requirement specification – User experiences and HCI – Usability Engineering life cycle – Task analysis – prototyping tools and techniques – scenario based design – Participatory design – Testing and evaluation – Usability testing – Inspection based evaluation – Model based evaluation

REFERENCES

1. Andrew sears, Julie A Jacko, Lawrence, The human computer interaction handbook: fundamentals, evolving technologies and emerging applications, 1e, Erlbaum Associates, 2008
2. Alan Dix, Janet Finlay, Gregory D Abowd, Russell Beale, Human - Computer Interaction, 3e, Pearson, 2012
3. Helen Sharp, Yvanno Rogers and Jenny Preece, Interaction Design: Beyond human Computer Interaction, 1e, John Wiley, 2011
4. Jan Noyes, Chris Baber, User centred design of systems, 1e, Springer, 2013

21-480-0106: Information Retrieval and Web Search

Core/Elective: **Elective**   Semester: **1**   Credits: **3**

Course Description

A coherent treatment of classical and web based information retrieval that includes web search, text classification, text clustering, gathering, indexing and searching documents and methods of evaluating systems .

Course Outcomes (CO)

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

CO1: Understand advanced techniques for text-based information retrieval .
CO2: Understand Boolean and vector space retrieval models
CO3: Evaluate various text classification techniques
CO4: Understand Web search characteristics, web crawling and link analysis
CO5: Build working systems that assist users in finding useful information on the Web

Mapping with Program Outcomes

CO1: PO1
CO2: PO1
CO3: PO1,PO4
CO4: PO1
CO5: PO1,PO4

Course Content

1. Taxonomy of IR Models – Classic models- Set theoretic model- Algebraic models- Probabilistic model- Structured text retrieval models- Models for browsing- Retrieval evaluations-Reference collections

2. Query languages-query operations-text and multimedia languages-Text operations-document preprocessing- matrix decompositions and latent semantic indexing-text compression –indexing and searching-inverted files-suffix trees- Boolean queries-sequential searching-pattern matching

3. Text Classification, and Naïve bayes-vector space classification-support vector machines and machine learning on documents-flat clustering –hierarchical clustering

4. Web search basics-web characteristics-index size and estimation- near duplicates and shingling-web crawling-distributing indexes- connectivity servers-link analysis-web as a graph- PageRank-Hubs and authorities- question answering

5. Online IR systems- online public access catalogs-digital libraries-architectural issues-document models - representations and access- protocols

References

1. Ricardo Baezce Yates, BerthierRibeiro-Neto , Modern Information Retrieval: The Concepts and Technology behind Search, 3e, ACM Press, 2017
2. Christopher D. Manning, PrabhakarRaghavan and HinrichSchütze , Introduction to Information Retrieval, 1e, Cambridge University Press, 2008
3. Bruce Croft, Donald Metzler and Trevor Strohman, Search Engines: Information Retrieval in Practice, 1e, AW, 2009

21-480-0107: Functional Programming

Core/Elective: **Elective** Semester: **1** Credits: **3**

Course Description

As big data and multiple cores become ubiquitous, functional programming has become relevant as never before. The latest standards for popular programming languages like C++ and Java have included support for a large number of functional programming features. This course aims to provide a thorough introduction to functional programming. It covers both the theoretical underpinnings and practical, programming aspects.

Course Outcomes (CO)

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

CO1: Explain and appreciate the functional programming paradigm.
CO2: Identify the various methods in functional programming as different from imperative programming.
CO3: Analyze the proofs of correctness of functional programming codes.
CO4: Develop programming skills on any one frontline functional languages (e.g. Haskell, Clojure, Kotlin etc.)
CO5: Appreciate the need for imperative constructs and explain efficient methods and workarounds for the cases in functional programming languages.
CO6: Measure and appraise the recent adaptations of functional concepts into non-functional languages.

Mapping with Program Outcomes

CO1: PO1
CO2: PO1
CO3: PO1, PO4
CO4:.PO4
CO5:.PO1
CO6: PO1, PO2, PO4

Course Content

1. Introduction to Functional Programming – Motivation – Defining features of the functional Paradigm – First Class Functions – Referential Transparency – Introduction to Haskell – Data Types and Pattern Matching– Laziness – Program Correctness

2. Lambda Calculus – Alpha, beta conversions – Normal forms – Applicative order – Reductions - Church Rosser Theorems – Y combinator – Recursion – Proofs of correctness.

3. Classes for Numbers – Lists in Haskell – Basic List operations – Higher order list functions – List comprehension – Strings and Tuples – User defined data types: lists, queues, trees.

4. Proving correctness of programs – Induction – Proofs using higher order functions – Infinite Lists – Lazy Evaluation – Efficiency – Controlling Space and Time complexity – Polymorphism -

Conditional Polymorphism – Type classes

5. Programming imperatively in Haskell – The IO Monad – Why Monads are Necessary – The State Monad– ST Monad – Mutable and Immutable Arrays – Parsing using Monads – Applications – Fault-tolerant systems – Financial analysis – Comparison to other functional languages.

References

1. Richard Bird, Thinking Functionally with Haskell, 1e, Cambridge University Press, 2014.
2. Graham Hutton, Programming in Haskell, 1e, Cambridge University Press, 2007.
3. KeesDoets, Jan van Eijck, The Haskell Road to Logic, Maths and Programming, 2e, College Publications, 2004.
4. Greg Michaelson, an Introduction to Functional Programming through Lambda Calculus, 1e, Dover Publications, 2011.
5. Chris Okasaki, Purely Functional Data Structures, 1e, Cambridge University Press, 1999.

21-480-0108: Software Quality Management

Core/Elective: **Elective** Semester: **1** Credits: **3**

Course Description

This course discusses basic software project quality management principles and techniques as they relate to software project planning, monitoring and control. This course describes the basics of software verification and validation planning with an emphasis on software peer reviews and software testing. The course also covers software configuration management, technical metrics for software.

Course Outcomes (CO)

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

CO1: Understand the basics and benefits of software quality engineering
CO2: Plan, implement and audit a Software Quality Management program for their organization
CO3: Select, define, and apply software measurement and metrics to their software products, processes and services
CO4: Understand the fundamentals of the configuration management process to include configuration identification, configuration control, status accounting, and audits

Mapping with Program Outcomes

CO1: PO4
CO2: PO1, PO4
CO3: PO1, PO4
CO4: PO4

Course Content

1. Introduction to software quality: Software Quality - Hierarchical models of Boehm and McCall - Quality measurement - Metrics measurement and analysis - Gilb's approach -GQM Model

2. Tools for Quality - Ishikawa's basic tools - CASE tools - Defect prevention and removal - Reliability models - Rayleigh model - Reliability growth models for quality assessment

3. Testing for reliability measurement Software Testing - Types, White and Black Box, Operational Profiles - Difficulties, Estimating Reliability, Time/Structure based software reliability - Assumptions, Testing methods, Limits, Starvation , Coverage, Filtering, Microscopic Model of Software Risk

4. Software reliability and availability - standards and evaluation of process - ISO 9000 - SEI Capability Maturity Model (CMM) - Software configuration management -

5. Technical metrics for software - metrics for the analysis model - metrics for design model - metrics for source code - metrics for testing - metrics for maintenance - technical metrics for object oriented systems

- distinguishing characteristics - class oriented metrics -operation oriented metrics - testing metrics -project metrics

References

1. Allan C. Gillies, Software Quality: Theory and Management, 3e, Cengage, 2003
2. Ron S Kenett, E. R Baker, Software Process Quality- Management and Control, 1e, CRC, 1999
3. Stephen H. Kan , Metrics and Models in Software Quality Engineering, 1e, AW, 2014
4. Patric D. T.O connor , Practical Reliability Engineering, 5e, John Wesley & Sons, 2011
5. Roger S. Pressman, Software Engineering - A practitioner's approach, 8e, McGraw Hill,2014

21-480-0109: Design and security of Internet of Things

Core/Elective: **Elective** Semester: **1** Credits: **3**

Course Description

Internet of Things (IoT) is a specialized area of computing having an interdisciplinary nature. In this course, the basic components of IoT and its applications will be taught. IoT design requires knowledge on cloud computing, lightweight communication protocols, data visualization, etc.. We learn to create IoT solutions based on public platforms, discuss its security vulnerabilities and remedies.

Course Outcomes (CO)

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

CO1: Understand the basic concepts of design of Internet of Things. CO2:
Develop the engineering skills to build internet of things solutions.CO3:
Understand concepts of cloud computing and data encryption.
CO4: Evaluate the public cloud commercial solutions.
CO5: Examine various applications of Internet of Thigs and its security vulnerabilities.CO6: Apply security concepts to secure Internet of Things.

Mapping with Program Outcomes

CO1:PO1
CO2:PO4
CO3:PO1
CO4:PO2, PO3
CO5:PO3
CO6:PO1, PO4

Course Content

Module I: Introduction to Internet of Things: Automation and Industry 4.0. Internet of Things architecture: 4-layer architecture. Sensors and Actuators, Controllers, Devices, Gateways, PLC, SCADA. Cyber Physical Systems, M2M Communication and Networking, LAN, WAN.Connectivity for IoT: Ethernet, Wireless, Bluetooth Low Energy, Zigbee, Mobile, LoRA, RFID.

Module II: Software Engineering for IoT: Cloud computing, Computing models in cloud: IaaS, PaaS, SaaS. Scaling services on cloud. Edge computing, fog computing. MQTT: broker,publish, subscribe, Mosquitto, COAP, Websocket. Apache Kafka as Data Dispatcher, Rule Engine. Time Series data, Apache Cassandra.

Module III: Programming for IoT: Hosting IoT server in cloud. Using public Cloud IoT platforms: AWS/Azure/Google Cloud Etc. IoT Core, storing data, Analytics using public IoT platforms. Communicating from edge devices to gateway: Programming for Arduino/Raspberry Pi or Python compatible boards. Data Representation and Visualization: Building dashboards and mobile apps for IoT Analytics.

Module IV: Securing IoT: Encryption of data, symmetric and asymmetric key encryption. Lightweight protocols. Threat analysis for IoT: Types of Cyber Attacks on IoT and solutions. Cloud security: Authentication and Authorization, Tokens, API Key, Identity and Access Management in cloud.

Module V: Applications of IoT: Diagnostics, Maintenance and Predictive Analytics. Rule based and Model based IoT Analytics. Analytics on cloud and edge. Machine Learning based Analytics on cloud. Anomaly detection, Digital Twin. Applications in healthcare, smart homes,smart grid, etc.

References

1. Giacomo Veneri and Antonio Capasso, Hands-On Industrial Internet of Things: Create a powerful Industrial IoT infrastructure using Industry 4.0, 1st Edition, PacktPublishing, 2018.
2. Mayur Ramgir, Internet of Things: Architecture, Implementation and Security, 1stEdition, Pearson, 2019.
3. R. Buyya, S N. Srirama, Fog and Edge Computing: Principles and Paradigms, WileySeries on Parallel and Distributed Computing, 1st Edition, Wiley, 2019.
4. Edward A. Lee and Sanjit A. Seshia, Introduction to Embedded Systems, A Cyber-Physical Systems Approach, 2nd Edition, MIT Press, 2017.
5. Arshdeep Bahga, Vijay Madisetti, Internet Of Things: A Hands-On Approach, 1stEdition, University Press, 2015.
6. S. Misra, A. Mukherjee, and A. Roy. Introduction to IoT. 1st Edition, Cambridge University Press, 2021.

21-480-0110: Quantum Computing

Core/Elective: **Elective** Semester: **1** Credits: **3**

Course Description

This course introduces concepts of Quantum Computing. This includes representation of quantum information and use of quantum algorithms. The course will cover design of quantum circuits and provide understanding of quantum noise and quantum cryptography.

Course Outcomes (CO)

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

CO1: Understand the basic concepts Quantum Computing.
CO2: Examine quantum representation of information and encryption.
CO3: Design quantum circuits using qubit gates.
CO4: Apply quantum algorithms for computation.
CO5: Understand quantum noise and error correction for fault tolerant computation.

Mapping with Program Outcomes

CO1:PO1
CO2:PO4
CO3:PO1, PO4
CO4:PO2, PO4
CO5:PO1

Course Content

Module I: Introduction to Quantum Computation: Quantum bits, representation of a qubit and multiple qubits. Quantum mechanics, Probabilities and measurements, entanglement, density operators and correlation, Measurements in bases other than computational basis.

Module II: Quantum correlations: Bell inequalities and entanglement, Schmidt decomposition, super-dense coding, teleportation, PPT criterion. Quantum Circuits: single qubit operations, multiple qubit gates, Universal quantum gates, design of quantum circuits.

Module III: Quantum Algorithms: Classical computation on quantum computers. Relationship between quantum and classical complexity classes. Deutsch’s algorithm, Deutsch’s-Jozsa algorithm, , quantum search.

Module IV: Quantum Information and Cryptography: Comparison between classical and quantum information theory. Shannon entropy, noiseless coding, Bell states. Quantum teleportation. Quantum Cryptography, no cloning theorem.

Module V: Noise and error correction: Quantum noise and quantum operations, Flip code, The Shor code, Quantum error correction, Stabilizer codes, Hamming bound. fault-tolerant quantum computation.

References

1. Quantum Computation and Quantum Information, M. A. Nielsen & I. Chuang, Cambridge University Press, 10th Edition, 2010.
2. J. Hidary, Quantum Computing: An Applied Approach, 1st Edition, Springer Publishing, 2019.
3. David McMahon, Quantum computing explained, Wiley-Interscience, 1st Edition, John Wiley & Sons, Inc. Publication, 2008.
4. A. Peres. Quantum Theory: Concepts and Methods. 1st Edition, Springer, 1995.
5. J. Preskill, Lecture Notes on Quantum Information and Computation, California Institute of Technology, 2021.
6. Mark M. Wilde, Quantum information Theory, 1st Edition, Cambridge University Press, 2012.
7. D. A. Lidar and T. A. Brun, Quantum error correction, 1st Edition, Cambridge University Press, 2013.

21-480-0201: Big Data Analytics

Core/Elective: **Core** Semester: **2** Credits: **4**

Course Description

In the age of big data, data science (the knowledge of deriving meaningful outcomes from data) is an essential skill that should be equipped by software engineers. It can be used to predict useful information on new projects based on completed projects. This course provides a practitioner's approach to some of the key techniques and tools used in Big Data analytics. Knowledge of these methods will help the students to become active contributors to the field of Data Science and Big Data Analytics

Course Outcomes (CO)

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

CO1: Understand predictive modeling techniques for data analytics
CO2: Apply data preprocessing techniques for big data
CO3: Measure the performance of data classification and regression models
CO4: Understand the use of Classification Trees and Rule-Based Models in big data analytics projects

Mapping with Program Outcomes

CO1: PO1, PO4
CO2: PO1
CO3: PO1, PO2
CO4:.PO1, PO4

Course Content

1. Predictive Models, Process, Data Pre-processing, Data Transformations, Over-Fitting and Model Tuning, Data Splitting, Resampling Techniques.

2.Measuring Performance in Regression Models, The Variance-Bias Trade-off, Linear Regression for Solubility Data, Penalized Models, Nonlinear Regression Models, Multivariate Adaptive Regression Splines, Support Vector Machines, K-Nearest Neighbors

3.Discriminant Analysis and Other Linear Classification Models, Linear Discriminant Analysis, Partial Least Squares Discriminant Analysis, Nearest Shrunken Centroids, Nonlinear Discriminant Analysis, Flexible Discriminant Analysis

4. Measuring Performance in Classification Models, Class Predictions, Class Probabilities, Evaluating Predicted Classes, Two-Class Problems, Evaluating Class Probabilities, Receiver Operating Characteristic (ROC) Curves

5. Classification Trees and Rule-Based Models, Regression Model Trees, Bagged Trees, Random Forests, Boosting, Remedies for Severe Class Imbalance, Factors That Can Affect Model Performance

References

1. Max Kuhn and Kjell Johnson, Applied Predictive Modeling, Springer 2013,2nd printing 2018 edition
2. Ankam Venkat, Big Data Analytics, Packt Publishing Limited, Birmingham, UK, 2016
3. EMC Education Services, Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data, Wiley; 1st edition (2015)
4. Hadley Wickham, Garrett Grolemund, R for Data Science: Import, Tidy, Transform, Visualize, and Model Data, Shroff/O'Reilly; First edition (2017)
5. Joel Grus, Data Science from Scratch, Shroff , O'Reilly Media (2015)
6. James D. Miller, Statistics for Data Science, Packt Publishing Limited (2017)
7. Thomas Rahlf, Data Visualisation with R: 100 Examples, Springer, 1st ed. (2017)

21-480-0202: Agile Software Engineering

Core/Elective: **Core** Semester: **2** Credits: **4**

Course Description:

Software development is a human activity. Agile methods, whether for project management or software development, are the ideal approach for developing software products where change is a risk factor. This course discusses the important milestones in effective software development and project management in the agile way.

Course Outcomes (CO)

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

CO1: Understand the agile principle and methodologies and appreciate the need for iterative approaches to software development
CO2: Develop a software product architecture using UML
CO3: Communicate with the development team using industry standard notations, designs and documentations.
CO4: Evaluate the purpose and benefits of agile methodologies like Scrum compared to traditional methods.
CO5: Apply various techniques, metrics and strategies for testing software projects.
CO6: Analyze, Formulate, and Apply key agile project management principles to manage a practical project.
CO7: Create an ability to work as a team leader by establishing goals, planning tasks and meeting the goals.

Mapping with Program Outcomes

CO1: PO1, PO4
CO2: PO1
CO3: PO2
CO4:.PO1, PO4
CO5:.PO1, PO4
CO6: PO2, PO4
CO7:.PO1, PO3

Course Content

1. Agile product architecting using UML: Envisioning the product – product vision – desirable qualities of the vision - customer needs – techniques for creating vision – dependencies and layering

2. Agile testing and development: Testing in agile, Refactoring development artifacts, agile patterns for user interface development

3. Agile project management principles. Agile philosophy. APM frameworks – envision, speculate, explore, adapt and close. Configuring project life cycles. Deliverables – management, technical. Feature-based delivery Agile technical team: Roles and responsibilities, team empowerment, leadership collaboration

4. Agile practices: Facilitated workshops, MoSCoW approach to prioritization, iterative development methodologies – SCRUM and XP, modeling, timeboxing

5. Agile project planning and estimation: Agile requirements **-** structure and hierarchy of requirements. The Agile approach to estimating**-** Agile measurements

References

1. Gary McLean Hall, Adaptive Code: Agile coding with design patterns and SOLID principles Microsoft Press; 2 edition (2017)
2. Robert C. Martin, Clean Code: A Handbook of Agile Software Craftsmanship, PHI; First edition (2017)
3. Marcus Ries and Diana Summers, Agile Project Management: A Complete Beginner's Guide To Agile Project Management, CreateSpace Independent Publishing Platform, (2016)
4. Effective Project Management: Traditional, Agile, Extreme, (7thEd): Robert K. Wysocki; Wiley India (2014)
5. Project Management the Agile Way: Making it Work in the Enterprise (1st Ed): John C. Goodpasture, Cengage Learning India (2014)

21-480-0203: Seminar

Core/Elective: **Core** Semester: **2** Credits: **1**

Course Description

The student has to prepare and deliver a presentation on a research topic suggested by the department before the peer students and staff. They also have to prepare a comprehensive report of the seminar presented.

Course Outcomes (CO)

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

CO1: Find out relevant information for the topic.
CO2: Define clearly the topic for discussion.
CO3: Deliver the content effectively.
CO4: Organize the content with proper structure and sequencing.
CO5: Demonstrate the academic discussion skills to emphasize, argue with clarity of purpose using evidence for the claims.
CO6: Show ability to evaluate and reflect on critical questions.
CO7: Show attempts to reach across diverse disciplines and bring other schools of thoughts into the discussions.

Mapping with Program Outcomes

CO1: PO1
CO2: PO1
CO3: PO1, PO2
CO4:.PO1
CO5:.PO1
CO6: PO1, PO4
CO7:.PO1, PO4

21-480-0204: Software Architecture

Core/Elective: **Elective** Semester: **2** Credits: **3**

Course Description

This course introduces the essential concepts of software architecture. Software architecture is an abstract view of a software system distinct from the details of implementation, algorithms, and data representation. Architecture is, increasingly, a crucial part of a software organization's business strategy.

Course Outcomes (CO)

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

CO1: Understand the relationship between system qualities and software architectures
CO2: Understand various software architectural patterns
CO3: Evaluate various Architectural Styles
CO4: Evaluate various Architectural Styles
CO5: Understand architecture based development product lines.
CO6: Understand architectures specific for Big data systems
CO7: Case study of various typical software architectures

Mapping with Program Outcomes

CO1: PO1
CO2: PO1
CO3: PO1,PO4
CO4: PO1,PO4
CO5: PO1,PO4
CO6: PO1,PO3
CO7: PO2,PO3,PO4

Course Content

1. The architecture Business Cycle (ABC) – Roots of Software architecture - Software architecture definitions and importance – Architectures and quality attributes -Architectural Styles - Architectural views: Need for multiple views – Some representative views – Conceptual View – Module view – Process view – Physical view – Relating the views to each other – The Software Architecture analysis Method (SAAM).
2. Life cycle view of architecture design and analysis – Eliciting quality attributes - QAW – Design of architecture - the ADD method – Evaluating architecture -ATAM method

3.Architecture-based development Product lines – cost and benefits of product line approach – product line activities – practice areas – patterns – PLTP – phased approach for adopting product lines

4. Software Architecture for Big Data Systems - Big Data from a software Architecture Perspective-Horizontal Scaling Distributes Data - Big Data – A complex software engineering problem-Software Engineering at Scale -Enhancing Design Knowledge for Big Data Systems- QuA Base – A Knowledge Base for Big Data System Design

5. Case study of J2EE/EJB - Future of software architecture

References

1. Len Bass, Software Architecture in Practice, 3e, Pearson, 2013
2. G.Zayaraz , Quantitative approaches for Evaluating Software Architectures: Frameworks and Models, 1e, VDM Verlag, 2010
3. Klaus Pohl et.al, Software Product Line Engineering: Foundations, Principles and Techniques, 1e, Springer, 2011
4. Web resource: Ian Gorton, Software Architecture for Big DataSystems https://www.sei.cmu.edu/webinars/view_webinar.cfm?webinarid=29834

21-480-0205: Software Agent Systems

Core/Elective: **Elective** Semester: **2** Credits: **3**

Course Description:

This course provides a thorough understanding of agent related system development. Software agents are finding their way into areas such as environmental security, climate change, seismic safety, epidemic prevention, detection and response, computer emergency response and human and societal dynamics

Course Outcomes (CO)

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

CO1: Explain agent development life cycle
CO2: Contrast types of agents
CO3: Build simple agents
CO4: Identify the components of a multi-agent system

Mapping with Program Outcomes

CO1: PO1, PO4
CO2: PO1
CO3: PO1, PO2
CO4:.PO1, PO4
Course Objectives
To understand Agent development
To gain Knowledge in Multi agent and intelligent agents
To Understand Agents and security
To gain Knowledge in Agent Applications

Course Content

1. The agent landscape – The smart agent framework: Introduction – Initial concepts – Entities- Objects – Agents – Autonomy – Tropistic agent – Specification structure of SMART. – Agent relationships – An operational analysis of Agent relationships.

2. Sociological Agents - Autonomous Interaction - Contract Net as a global directed system – Computational Architecture for BDI agents – Evaluating social dependence networks – Normative agents.

3. Intelligent Agents –Deductive Reasoning Agents – Practical reasoning agents - Reactive agents – Hybrid Agents – Understanding Each other – Communicating – Methodologies

4. Modeling multi agent system with AML – JADE:Java Agent development frame work – wireless sensor networks and software Agents – Multi agent Planning Security and anonymity in agent systems.

5. Multi Agent system: Theory approaches and NASA applications – Agent based control for multi-UAV information collection- Agent based decision support system for Glider pilots – Multi agent system in E-Health Territorial Emergencies – Software Agents for computer network security- Multi-Agent Systems,

Ontologies and Negotiation for Dynamic Service Composition in Multi-Organizational Environmental Management.

References

1. Mohammad Essaidi, Maria Ganzha, and MarcinPaprzycki, Software Agents, Agent Systems and Their Applications, 1e, IOS Press, 2012.
2. Mark D . Inverno and Michael Luck, Understanding Agent Systems, 1e, Springer,2010.
3. Michael Wooldridge, An Introduction to Multi Agent Systems, 1e, John Wiley & Sons Ltd., 2009.
4.Lin Padgham, Michael Winikoff, Developing Intelligent Agent Systems: A Practical Guide, 1e, John Wiley & Sons Ltd., 2004.
5. Bradshaw, Software Agents, 1e, MIT Press, 1997.
6. Richard Murch, Tony Johnson, Intelligent Software Agents, 1e, Prentice Hall, 2000.

21-480-0206: Enterprise Application Integration and Business Process Management

Core/Elective: **Elective** Semester: **2** Credits: **3**

Course Description

The course will introduce the major design, implementation and deployment issues regarding system integration, data-oriented cross-platform integration, e-business applications implementation and the security considerations in enterprise level multi-location systems integration. Business Process Management (BPM) is the set of concepts, methods, and tools that help organizations define, implement, measure and improve their end-to-end processes.

Course Outcomes (CO)

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

CO1: Explain and appreciate the need for application integration at enterprises level.
CO2: Contrast data-oriented and service-oriented integration processes.
CO3: Explain middleware technologies.
CO4: Show how to design business processes for process-aware information systems.
CO5: Show how to implement business processes for process-aware information systems.

Mapping with Program Outcomes

CO1: PO1
CO2: PO1
CO3: PO1, PO2
CO4:.PO1, PO4
CO5:.PO1, PO4

Course Content

1. Application Integration Overview: Problems in large-scale application integration, Business & Service Oriented Integration: XSLT Processing, Enterprise Service Bus, Web services introduction, Second generation web services –messaging –security –metadata.

2. Middleware: Basics and types, Distributed Transactions, Two Phase Commit, Message-oriented Middleware (MoM), Java middleware, Integration Servers, XML and other standards. Commercial examples.

3. Data-orientated Application Integration: Loosely couples systems, Data oriented programming, Data flow architecture, Event driven architecture. Integration with Business systems: Legacy systems integration –challenges, External system integration standards –RosettaNet –ebXML – UCCNet.

4. Integration standards: SOAP, XML-RPC, REST. Vertical Application Integration. The Application Integration Process. Reliability and Fault-tolerance. Ontologies. Data integration patterns.

5. Business Process Analysis and Design: Workflows & BPMS, Introduction to BPMN, Managing Processes, Components of process models, Process Management Maturity, Rules, Integrating rules with processes, Process dashboards. Commercial solutions.

References

1. Thomas Erl, Service Oriented Architecture: A field guide to Integrating XML and Web Services, 1e, Prentice Hall, 2004
2. G.Hohpe and B. Woolf Enterprise Integration Patterns: Designing, Building and Deploying Messaging Solutions, 1e, AW Professional, 2003
3. D. Linthicum, Next Generation Application Integration: From Simple Information to Web Services, 1e, Addison Wesley, 2003
4. Michael Havey , Essential Business Process Modeling, 1e, O'Reilly Media, 2005

21-480-0207: Advanced Data Mining

Core/Elective: **Elective** Semester: **2** Credits: **3**

Course Description

Data mining is the science of extracting hidden information from large datasets. This course offers a clear and comprehensive introduction to both data mining theory and Practice. All major data mining techniques will be dealt with and how to apply these techniques in real problems are explained through case studies.

Course Outcomes (CO)

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

CO1: Understand and appreciate the need for mining large volumes of data and the importance of its applications
CO2: Understand the different techniques for analyzing data
CO3: Introduce the multidisciplinary field of data mining
CO4: Understand statistical descriptions of data and its visualization.
CO5: Evaluate different methods for mining frequent patterns, association and correlations in large data sets
CO6: Evaluate different methods for data classification and prediction.
CO7: Evaluate different methods for data clustering
CO8: Evaluate different methods for outlier detection
CO9: Apply different spatial data mining techniques for the identification of spatial patterns.

Mapping with Program Outcomes

CO1: PO1,PO2,
CO2: PO1,PO2,
CO3: PO1,PO2,
CO4: PO1,PO2,PO4
CO5: PO1,PO2,PO4
CO6: PO1,PO2,PO4
CO7: PO1,PO2,PO4
CO8: PO1,PO2,PO4
CO9: PO1,PO2,PO3,PO4

Course Content

1. Statistical descriptions of data-data visualization-measuring data similarity and dissimilarity-data pre-processing-data cleaning-data integration-data reduction-data transformation-data warehouse modeling-design-implementation-data cube technology- queries by data cube technology-multidimensional data analysis in Cube space

2. Mining frequent patterns, associations and correlations – pattern mining in multidimensional space-colossal patterns- approximate patterns- applications- Mining data streams-Mining Sequence patterns in transactional databases- mining sequence pattern in Biological Data

3. Classification and prediction- decision tree induction-Bayesian classification-rule-based classification-neural networks-support vector machines-lazy learners-genetic algorithms- model evaluation-Cluster analysis- portioning methods- hierarchical methods- density based methods-grid based-probabilistic model based clustering- clustering high dimensional data- constraint based clustering- clustering high dimensional data-graph clustering methods

4. Outlier detection- outliers and outlier analysis- outlier detection methods-statistical approaches-proximity based approaches- clustering based approaches- classification based approaches-mining contextual and collective outliers- outlier detection in High-Dimensional data

5. Time series representation and summarization methods-mining time series data -Spatial data mining-spatial data cube construction-mining spatial association and co-location patterns-spatial clustering and classification methods-spatial trend analysis- Multimedia data mining-text mining- mining world wide web- trends in Data mining

References

1. Theophano Mitsa, Temporal Data mining, 1e, CRC Press, 2018
2. Jiawei Han &Micheline Kamber, Jian Pei, Data mining concepts and techniques, 1e, Elsevier, 2014
3. A B M Showkat Ali, Saleh A Wasimi, Data mining methods and Techniques, Cengage

21-480-0208: Fuzzy Set Theory: Foundations and Applications

Core/Elective: **Elective** Semester: **2** Credits: **3**

Course Description

This course concentrates on fuzzy set theory and its application. This includes the concepts, and techniques from fuzzy sets and fuzzy logic to enhance machine learning techniques.

Course Outcomes (CO)

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

CO1: Explain the concepts of fuzzy set theory
CO2: Apply fuzzy set theory to solve problems
CO3: Use the theory in optimization problems
CO4: apply the theory to enhance machine learning
.

Mapping with Program Outcomes

CO1: PO1
CO2: PO1,PO4
CO3: PO1,PO4
CO4: PO1,PO4

Course Content

1. Crisp sets and Fuzzy sets - Introduction - crisp sets an overview-the notion of fuzzy sets-basic concepts of fuzzy sets- membership functions - methods of generating membership functions-Defuzzification methods-operations on fuzzy sets- fuzzy complement- fuzzy union- fuzzy intersection-combinations of operations-General aggregation operation

2. Fuzzy arithmetic and Fuzzy relations-Fuzzy numbers-arithmetic operations on intervals- arithmetic operations on fuzzy numbers-fuzzy equations- crisp and fuzzy relations-binary relations- binary relations on a single set - equivalence and similarity relations- compatibility or tolerance relation

3. Fuzzy measures - Fuzzy measures - belief and plausibility measure - probability measures - possibility and necessity measures- possibility distribution- relationship among classes of fuzzy measures.

4. Fuzzy Applications-Fuzzy approximate reasoning- Fuzzy Expert System-Fuzzy systems-Fuzzy controllers-Fuzzy Neural Networks- Fuzzy automata-Fuzzy Dynamic systems

5. Fuzzy Clustering-Fuzzy Pattern Recognition-Fuzzy image processing - Fuzzy databases and information retrieval-Fuzzy Decision making - Fuzzy systems and Genetic algorithms - Fuzzy regression.

References

1. George J Klir and Tina AFolger: Fuzzy Sets, Uncertainty and Information, Fuzzy Sets, Uncertainty and Information, 1e, Pearson Education, 2015

2. George J Klir and Bo Yuan, Fuzzy Sets and Fuzzy Logic: Theory and Applications, 1e, Pearson Education, 2015.
3. Timothy J Ross: Fuzzy logic with Engineering Applications, 3e, Wiley, 2011.
4. H. J. Zimmerman: Fuzzy Set theory and its Applications,4e, Springer,2001.

21-480-0209: Complex Networks: Theory and Applications

Core/Elective: **Elective** Semester: **2** Credits: **3**

Course Description

Complex networks provide a powerful abstraction of the structure and dynamics of diverse kinds of interaction viz people or people-to-technology, as it is encountered in today's inter-linked world. This course provides the necessary theory for understanding complex networks and applications built on such backgrounds.

Course Outcomes (CO)

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

CO1: Explain and appreciate complex networks and complex network systems as different from other network systems viz. computer networks, transportation networks etc.
CO2: Explain the mathematical representation of complex networks in computer programs.
CO3: Explain and compute the centrality measures in network analysis.
CO4: Demonstrate random graph generation processes and associated properties.
CO5: Discriminate various algorithms for community detection in complex networks.
CO6: Evaluate different models for complex networks.
CO7: Illustrate and explain the flow models used in complex networks for modelling social, economic and biological systems.
CO8: Identify the performance requirements related to social media systems.
CO9: Explain the techniques for predictive modelling and analytics of social media data.

Mapping with Program Outcomes

CO1: PO1
CO2: PO1, PO4
CO3: PO1, PO4
CO4:.PO1, PO4
CO5:.PO1, PO2, PO4
CO6: PO2, PO4
CO7:.PO1, PO2, PO4
CO8:.PO1
CO9: PO1, PO4

Course Content

1. Networks of information – Mathematics of networks – Measures and metrics – Large scale structure of networks – Matrix algorithms and graph partitioning
2. Network models – Random graphs – walks on graphs - Community discovery – Models of network formation – Small world model - Evolution in social networks – Assortative mixing- Real networks - Evolution of random network - Watts-Strogatz model – Clustering coefficient - Power Laws and Scale-Free Networks – Hubs - Barabasi-Albert model – measuring preferential attachment- Degree dynamics – nonlinear preferential attachment

3. Processes on networks – Percolation and network resilience – Epidemics on networks – Epidemic modelling - Cascading failures – building robustness- Dynamical systems on networks – The Bianconi-Barabási model – fitness measurement – Bose-Einstein condensation
4. Models for social influence analysis – Systems for expert location – Link prediction – privacy analysis – visualization – Data and text mining in social networks - Social tagging
5. Social media - Analytics and predictive models – Information flow – Modelling and prediction of flow - Missing data - Social media datasets – patterns of information attention – linear influence model – Rich interactions

References

1. Mark J. Newman, Networks: An introduction, 1e, Oxford University Press, 2010

2. Charu C Aggarwal (ed.), Social Network Data Analytics, 1e, Springer, 2011

3. David Easley and Jon Kleinberg, Networks, Crowds, and Markets: Reasoning about a highly connected World, 1e, Cambridge University Press, 2010

4. Albert-Laszlo Barabasi, Network Science, 1e, Cambridge University Press, 2016

21-480-0210: Advances in Databases

Core/Elective: **Elective** Semester: **2** Credits: **3**

Course Description

This is a second course in database systems which cover advanced aspects of database systems touching upon the theoretical advancements to handle the new areas and challenges related to the management of data. The course introduces the students to the frontiers of the classical database systems and takes them to the multidimensional data and the associated processing techniques. Later, a large multitude of specialty databases are introduced. This course consolidates the theory and practices pertaining to big data storages and cloud databases.

Course Outcomes (CO)

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

CO1: Explain physical database design and usage analysis.
CO2: Demonstrate the need for Denormalization.
CO3: Compare the performance of different file organizations.
CO4: Construct star schema for DW architecture.
CO5: Explain the concepts of XML data modeling.
CO6: Develop OR mappings.
CO7: Explain the basics of active databases, main memory databases and spatio-temporal databases.
CO8: Illustrate the idea of NoSQL databases.
CO9: Compare and analyze KV data stores, document data stores and wide column data stores.
CO10: Construct cloud database models.

Mapping with Program Outcomes

CO1: PO1,PO4
CO2: PO1
CO3: PO1, PO4
CO4:.PO1, PO4
CO5:.PO1, PO4
CO6: PO1
CO7:.PO1
CO8:.PO1,PO2
CO9: PO1, PO4
CO10: PO1, PO4

Course Content

1. Physical Database Design: The Physical Database Design Process - Data Volume and Usage Analysis - Controlling Data Integrity - Missing Data – Denormalization –Partitioning - File Organizations – Heap-Sequential-Indexed-Hashed – Non-unique indexing.

2. Online Analytical Processing: Recent Enhancements and Extensions to SQL - Analytical and OLAP Functions–Multidimensional Analysis - New Data Types- New Temporal Features in SQL- Other

Enhancements. Need for Data Warehousing – Architectures- Data Mart and Data Warehousing Environment - Real-Time Data Warehouse Architecture - Enterprise Data Model-Status/Event/Transient/Periodic Data - Derived Data - Star Schema and variations - Fact Tables - Dimension Tables - Normalization - Surrogate Key - Hierarchies - Unstructured Data.

3. Object-Based Databases: Complex Data Types - Structured Types and Inheritance in SQL - Table Inheritance - Array and Multiset Types in SQL - Object-Identity and Reference Types in SQL - Implementing O-R Features - Persistent Programming Languages - Object-Relational Mapping. Object-Oriented Databases: Motivation – Concepts and Features – Object Modelling – Indexing – Design Considerations- Object-Oriented versus Object-Relational. XML Databases: Motivation - Structure of XML Data - XML Document Schema - Querying and Transformation – XPath – XQuery – XSLT - Application Program Interfaces to XML - Storage of XML Data - XML Applications.

4. Spatial and Temporal Data: Motivation - Time in Databases - Spatial and Geographic Data - Multimedia Databases -Mobility and Personal Databases - Active Databases, Time series Databases. Advanced Transaction Processing : Transaction-Processing Monitors - Transactional Workflows - E-Commerce - Main-Memory Databases - Real-Time Transaction Systems - Long- Duration Transactions.

5. Classification of NoSQL Database Management Systems, Key-Value Stores- Document Stores- Wide-Column Stores - Graph-Oriented Databases–Redis, MongoDB, Cassandra, Neo4j – Hadoop data storage – Pig, Hive, HBase – Introduction to Integrated Data Architecture.

Cloud Databases: Database as a service (DBaS), Amazon SimpleDB, DynamoDB – EnterpriseDB - Google Cloud SQL, Google BigQuery – Microsoft Azure SQL.

References

1. A. Hoffer Jeffrey, V. Ramesh, Topi Heikki, Modern database management, 12e, Pearson, 2015.
2. Abraham Silberschatz, Henry F. Korth, S. Sudarshan, Database System Concepts, 6e,McGraw-Hill, 2013
3. SherifSakr, Big Data 2.0 Processing Systems: A Survey, 1e, Springer: Briefs in Computer Science, 2016
4. Lee Chao, Cloud Database Development and Management, 1e, CRC Press, 2013
5. Rini Chakrabarti and Shilbhadra Dasgupta, Advanced Database Management System, 1e, Dreamtech Press, 2011

21-480-0211: Blockchain Technology

Core/Elective: **Elective** Semester: **2** Credits : **3**

Course Description

This course intends to provide a comprehensive insight into various Blockchain techniques. The objectives are to give an insightful introduction to the basic concepts of blockchain and its applications in various domains.

Course Outcomes (CO)

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

CO1: Understand the fundamentals of blockchain technology
CO2: Understand the essentials of Bitcoin and beholding bitcoins as blockchains
CO3: Analyse and design the Ethereum Blockchain
CO4: Build factom Blockchains
CO5: Analyse the powers of blockchains and their applications in various domains
CO6: Study the impact of blockchains on industry
CO7: Execute a mini project on blockchain

Mapping with Programme Outcomes

CO1: PO4
CO2: PO4
CO3: PO4
CO4: PO4
CO5: PO1, PO4
CO6: PO3, PO4
CO7: PO1, PO3,PO4

Course Content

Module I: Introduction to blockchain: Structure of blockchains, Blockchain life cycle, working of a blockchain, picking a blockchain, exploring blockchain applications, building trust with blockchains, Blockchain in action: Use cases, introducing bitcoin blockchains.

Module II: Bitcoin & Ethereum blockchains: Understanding bitcoins, comprehending bitcoins as blockchains, analyzing Ethereum blockchains, introducing ripple and factom blockchains and their importance

Module III: Powerful blockchain platforms: Getting introduced to Hyperledger, Hyperledger vision, Hyperledger sawtooth, understanding the blockchain fabric, understanding business, and smart blockchains, IBM Blockchains, Stellar: an optimized blockchain

Module IV: Industry impacts of blockchains: Blockchains in financial technology, Blockchains in various industries such as insurance, Government, Real-estate, health care, Telecommunication, Transportation, etc..

Module V: Case Study and mini-project: Studying different blockchain projects as a case study and submit a report and present the work, Designing a blockchain application as a mini-project, and presenting the work.

References

1. Blockchain and Crypto Currency, Editors: Makoto YanoChris DaiKenichi MasudaYoshio Kishimoto, 1st Edition, Springer, 2020.
2. Blockchain or Dummies, Tiana Laurence, 1st Edition , John Wiley & Sons, Inc, , 2017.
3. Blockchain Blueprint for a new economy, Melanie Swan, 1st Edition,O'Reilly, 2017.
4. Blockchain Technology: Applications and Challenges, Panda, S.K., Jena, A.K., Swain, S.K., Satapathy, S.C. , 1st Edition, Springer, 2021
5. Blockchain and Distributed Ledgers, Alexander Lipton and Adrien Treccani, 1st Edition, World Scientific Press, 2021

21-480-0301: Elective VI

Core/Elective: **Elective** Semester: **3** Credits: **2**

Course Description

A credit-based MOOC course of minimum 12 weeks duration or three non-credit based MOOC courses of 4-weeks duration from SWAYAM/NPTEL/any other platforms approved by the Department.

21-480-0302: Dissertation & Viva Voce

Core/Elective: **Core** Semester: **3** Credits: **16**

Course Description

The dissertation work spans two semesters. Through the dissertation work, the student has to exhibit the knowledge in terms of engineering or technological innovation or research ability to solve the contemporary problem. On completion of the first part of the work, the student shall submit an interim dissertation report. The qualitative and quantitative results of the work will be evaluated through a viva-voce exam.

Course Outcomes (CO)

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

CO1: Demonstrates in depth knowledge and thoughtful application through the detailed analysis of the problem chosen for the study
CO2: Assess the gap by acquiring knowledge about the previous works, and its interpretation and application
CO3: Demonstrates the design of the proposed methodology and its merits.
CO4: Organize the interim dissertation content with proper structure and sequencing
CO5: Demonstrate the academic discussion skills to emphasize, argue with clarity of purpose using evidence for the claims.

Mapping with Program Outcomes

CO1: PO1, PO4
CO2: PO1, PO4
CO3: PO1, PO4
CO4:.PO2
CO5:.PO2

21-480-0401: Dissertation & Viva Voce

Core/Elective: **Core** Semester: **4** Credits: **18**

Course Description

The dissertation work spans two semesters. Through the dissertation work, the student has to exhibit the knowledge in terms of engineering or technological innovation or research ability to solve the contemporary problem. On completion of the work, the student shall submit a final dissertation report. The qualitative and quantitative results of the work will be evaluated through a viva-voce exam.

Course Outcomes (CO)

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

CO1: Demonstrates in depth knowledge and thoughtful application through the detailed analysis of the problem chosen for the study
CO2: Assess the gap by acquiring knowledge about the previous works, and its interpretation and application
CO3: Demonstrates the design of the proposed methodology and its merits.
CO4: Organize the interim dissertation content with proper structure and sequencing
CO5: Demonstrate the academic discussion skills to emphasize, argue with clarity of purpose using evidence for the claims.
CO6: Show ability to evaluate and reflect on critical questions.

Mapping with Program Outcomes

CO1: PO1, PO4
CO2: PO1, PO4
CO3: PO1, PO4
CO4:.PO2
CO5:.PO2
CO6: PO1, PO4

Learning Outcomes and Assessment

Each course's learning outcomes will be assessed based on one or many methods, including the internal written tests, quizzes, presentations, seminars, assignments in the form of lab exercises, and group projects. The above assessment methods will be attentively created to support the intended learning outcomes that have been set out for a particular course. The program outcome attainment is measured using the CO/PO mappings.