RAMAIAH
Institute of Technology

CURRICULUM

for the Academic year 2021 – 2022

INFORMATION SCIENCE AND ENGINEERING

I - IV Semester M. Tech (Software Engineering)

RAMAIAH INSTITUTE OF TECHNOLOGY
(Autonomous Institute, Affiliated to VTU)
Bangalore – 560054.

About the Institute

Dr. M. S. Ramaiah a philanthropist, founded 'Gokula Education Foundation' in 1962 with an objective of serving the society. M S Ramaiah Institute of Technology (MSRIT) was established under the aegis of this foundation in the same year, creating a landmark in technical education in India. MSRIT offers 17 UG programs and 15 PG programs. All these programs are approved by AICTE. All eligible UG and PG programs are accredited by National Board of Accreditation (NBA). The institute is accredited with **'A^{+}' grade by NAAC in March 2021** for 5 years. University Grants Commission (UGC) & Visvesvaraya Technological University (VTU) have conferred Autonomous Status to MSRIT for both UG and PG Programs since 2007. The institute is a participant to the Technical Education Quality Improvement Program (TEQIP), an initiative of the Government of India. The institute has 380 competent faculty out of which 60% are doctorates. Some of the distinguished features of MSRIT are: State of the art laboratories, individual computing facility for all faculty members, all research departments active with sponsored funded projects and more than 300 scholars pursuing Ph.D. To promote research culture, the institute has established Centre of Excellence for Imaging Technologies, Centre for Advanced Materials Technology, Centre for Antennas and Radio Frequency systems (CARFS), Center for Cyber Physical Systems & Schneider Centre of Excellence. **M S Ramaiah Institute of Technology has obtained "Scimago Institutions Rankings" All India Rank 65 & world ranking 578 for the year 2020.**

The Entrepreneurship Development Cell (EDC) and Section 8 company "Ramaiah Evolute" have been set up on campus to incubate startups. **M S Ramaiah Institute of Technology secured All India Rank 8th for the year 2020 for Atal Ranking of Institutions on Innovation Achievements (ARIIA), by MoE, Govt. of India.** MSRIT has a strong Placement and Training department with a committed team, a good Mentoring/Proctorial system, a fully equipped Sports department, large air-conditioned library with good collection of book volumes and subscription to International and National Journals. The Digital Library subscribes to online e-journals from Elsevier Science Direct, IEEE, Taylor & Francis, Springer Link, etc. MSRIT is a member of DELNET, CMTI and VTU E-Library Consortium. MSRIT has a modern auditorium and several hi-tech conference halls with video conferencing facilities. The institute has excellent hostel facilities for boys and girls. MSRIT Alumni have distinguished themselves by occupying high positions in India and abroad and are in touch with the institute through an active Alumni Association. **As per the National Institutional Ranking Framework (NIRF), MoE, Government of India, M S Ramaiah Institute of Technology has achieved 65th rank among 1143 top Engineering institutions of India for the year 2021 and is 1st amongst the Engineering colleges affiliated to VTU, Karnataka.**

About the Department

Information Science and Engineering department is established in the year 1992 with an objective of producing high-quality professionals to meet the demands of the emerging field of Information Science and Engineering. Department also started M.Tech program in Software Engineering in the year 2004 and has been recognized as R&D center by VTU in 2012. The department is accredited by the NBA in 2001, 2004, 2010, 2015 and reaccredited in 2018 under Tier-1 till 2022. Department has highly qualified and motivated faculty members and well equipped state of the art laboratories. All faculty members are involved in research and technical papers publications in reputed journals, conferences across the world. Strong collaboration with industries and high profile institutions is in place for curriculum updates, more hands on training, practical's, project based learning, EPICS, expert lectures, partial course deliveries by industry experts and student interns to enhance the skills in emerging areas to keep an inclusive and diverse academic environment. Department is regularly conducting seminars, conferences and workshops for students and academicians in the emerging areas of Information Technology. Introduced EPICS in senior projects. Some of the laboratories have also been set up in collaboration with industries such as Intel, Microsoft, Apple, SECO, Honeywell, EMC^2, NVIDIA, IBM, Green Sense Werks, Tech Machinery Labs, Sesovera Tech Pvt. Ltd., and Ramaiah Medical College (Emergency department). Also, an echo system is built to initiate start-ups at the department level along with the mentorship. All the above potential activities have led to high profile placements, motivation to become an entrepreneur, and encouragement for higher learning.

VISION OF THE INSTITUTE

To be an Institution of International Eminence, renowned for imparting quality technical education, cutting edge research and innovation to meet global socio-economic needs

MISSION OF THE INSTITUTE

MSRIT shall meet the global socio-economic needs through

- Imparting quality technical education by nurturing a conducive learning environment through continuous improvement and customization
- Establishing research clusters in emerging areas in collaboration with globally reputed organizations
- Establishing innovative skills development, techno-entrepreneurial activities and consultancy for socio-economic needs

QUALITY POLICY

We at MS Ramaiah Institute of Technology strive to deliver comprehensive, continually enhanced, global quality technical and management education through an established Quality Management System complemented by the synergistic interaction of the stake holders concerned

VISION OF THE DEPARTMENT

To evolve as an outstanding education and research center of Information Technology to create high quality Engineering Professionals for the betterment of Society

MISSION OF THE DEPARTMENT

- To provide a conducive environment that offers well balanced Information Technology education and research.
- To provide training and practical experience in fundamentals and emerging technologies.
- To nurture creativity for overall personality development.

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)

PEO1: Contribute in the area of Software Engineering development, maintenance and research in social-technical system

PEO2: Exhibit the Software Engineering skills for analysis, design and testing using modern tools and technologies within or outside discipline.

PEO3: Act according to professional ethics and communicate effectively with various stakeholders by demonstrating leadership qualities.

PROGRAMME OUTCOMES (POs)

PO1: An ability to independently carry out research/investigation and development work to solve practical problems.

PO2: An ability to write and present a substantial technical report/document.

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

PO4: An ability to analyze, design, verify, validate, implement, apply and maintain software systems

PO5: A recognition of the need for, and an ability to apply, professional and ethical responsibilities

Curriculum Course Credits Distribution

Semester	Humanities & Social Sciences (HSS)	Basic Sciences/ Lab (BS)	Engineering Sciences/ Lab (ES)	Professional Courses - Core (Hard core, soft core, Lab) (PC-C)	Professional Courses- Electives (PC-E)	Other Electives (OE)	Project Work/ Internship (PW /IN)	Extra & Co-curricular activities (EAC)	Total credits in a semester
First				14	8				22
Second				14	8				22
Third				1	4		17		22
Fourth				2			20		22
Total									**88**

SCHEME OF TEACHING

I SEMESTER

Sl. No	Subject Code	Subject	Credits				Contact Hours
			L	T	P	Total	
1	MSWE11	Advanced Software Engineering	4	0	0	04	04
2	MSWE12	Software Architecture and Design Patterns	3	1	0	04	05
3	MSWE13	Machine Learning	4	0	0	04	04
4	MSWEL1	Enterprise Application Development – I lab	0	1	1	02	04
6	MSWEAX	Elective – A	4	0	0	04	04
7	MSWEBX	Elective – B	4	0	0	04	04
Total			**20**	**1**	**1**	**22**	**25**

Elective - A

MSWEA1	Cryptography and Information Security
MSWEA2	Information Retrieval
MSWEA3	Mobile Computing

Elective - B

MSWEB1	Probability, Statistics and Queuing Theory
MSWEB2	Advanced Algorithms
MSWEB3	Web Services

SCHEME OF TEACHING

II SEMESTER

Sl. No	Subject Code	Subject	Credits				Contact Hours
			L	T	P	Total	
1	MSWE21	Software Metrics and Quality Engineering	4	0	0	4	04
2	MSWE22	Software Project Management	3	1	0	4	05
3	MSWE23	Big Data and Cloud Computing	4	0	0	4	04
4	MSWE24	Research Methodology	2	0	0	2	02
5	MSWEL2	Enterprise Application Development – II Lab	0	1	1	2	04
6	MSWECX	Elective – C	3	0	0	3	03
7	MSWEDX	Elective – D	3	0	0	3	03
Total			**20**	**1**	**1**	**22**	**25**

Elective - C

MSWEC1	Internet of Things
MSWEC2	Data Science
MSWEC3	Advances in Operating Systems

Elective - D

MSWED1	Cognitive Computing
MSWED2	System Performance and Analysis
MSWED3	Blockchain and Cyber Security

SCHEME OF TEACHING

III SEMESTER

Sl. No.	Course Code	Course Name	Category	Credits					Contact Hours
				L	T	P	S	Total	
1	MSWEEX	Elective – E	PC-E	4	0	0	0	04	04
2	MSWE31	Internship/Industrial Training	IN	0	0	4	0	04	08
3	MSWE32	Project Preliminaries	PW	0	3	10	0	13	26
4	MSWE33	Technical Seminar	PC-C	0	1	0	0	01	02
Total				**4**	**4**	**14**	**0**	**22**	**40**

Elective- E

MSWEE1	Deep Learning
MSWEE2	Bioinformatics
MSWEE3	Software mining and analysis

SCHEME OF TEACHING

IV SEMESTER

Sl. No.	Course Code	Course Name	Category	Credits				Contact Hours
				L	T	P	Total	
1	MSWE41	IPR and cyber security laws	PC-C	0	2	0	02	04
2	MSWE42	Project-2	PW	0	0	20	20	40
Total				**0**	**2**	**20**	**22**	**44**

I Semester

ADVANCED SOFTWARE ENGINEERING

Course Code: MSWE11 **Credit: 4:0:0**
Prerequisite: Software Engineering **Contact Hours: 56L**
Course Coordinator: Dr Naresh E

Course Content

Unit I

Process Models and their evolution – Waterfall Model, incremental model, Spiral Model, prototyping model, V-model, Challenges in legacy software development, Agile Manifesto, Agile Process and Principles, Extreme programming, Scrum, Test-driven development, Rational Unified Process, maturity models, Critical Analysis of Process Models.

Unit II

DevOps – Introduction, DevOps life cycle, Principles, benefits, Roles, Responsibilities and skills for a DevOps engineer, DevOps versus Agile, Continuous Integration and Deployment, Tools at various stages of DevOps. Research Issues in Software Engineering.

Requirements Engineering - Product versus service requirements engineering. Functional and Non-functional requirements, Requirements engineering process, Requirements Elicitation techniques. Requirements validation, Requirements management, Requirements engineering in modern era.

Unit III

Software Design – Concepts, Design principles, Object Oriented Design with UML- Class diagrams, Use case diagrams, sequence diagrams. Universal design applied to software engineering, Design for Reuse.

Software Architecture - Introduction, characteristics, architectural design decisions, architectural views, Architectural styles.

User Interface Design – Introduction, Golden rules, interface analysis and design process, Interface design steps, WebApp interface design.

Unit IV

Programming Paradigms – Imperative programming, Functional programming, Logical programming, Object oriented programming, Global software development – tools and practices, Coding standards, Aspect oriented software engineering, Free and Open Source Software Engineering. Test driven software development, Object Oriented Testing with C&K metrics, Software Configuration Management.

Unit V

Formal models in software engineering – Introduction, Benefits of Formal Models, Weaknesses of Formal Methods, Varieties of formal analysis, Formal specifications, Types of Formal Specifications, Applications of formal methods.

References:

1. Roger S Pressman, Software Engineering, 7th edition, TMH publication.
2. Ian Sommerville, Software Engineering, 9th edition, Pearson Education.
3. Rumbaugh, Object –Oriented Modeling and Design, Pearson Education.

Course Outcomes (COs):

At the end of the course, students will be able to-

1. Identify various software development processes and methodologies (PO1,2).
2. Gather the requirements in DevOps environment. (PO2,3).
3. Design various parts of software through UML diagrams and design patterns. (PO1,3,4).
4. Implement software modules through components. (PO1, 4).
5. Understand the software testing concepts and formal methods. (PO5).

SOFTWARE ARCHITECTURE AND DESIGN PATTERNS

Course Code: MSWE12 **Credit: 3:1:0**
Prerequisite: NIL **Contact Hours: 42L + 14T**
Course Coordinator: Dr Sumana M

Course Content

Unit I

Introduction to software architecture, Common architectural styles including Pipes and Filters, OO, Event based invocations, Layered systems, Repositories, Table driven interpreters and heterogeneous architectures. Some case studies in software architecture.

Unit II

Software design patterns, OO design principles, Creational patterns: Abstract Factory, Builder, Factory Method, Prototype, Singleton.

Unit III

Concepts and Applications of Structural Patterns: Adapter, Bridge, Composite, Decorator, Facade, Flyweight, Proxy, Case studies.

Unit IV

Chain of responsibility, Command, Interpreter, Iterator, Mediator. Overview, Applications, case studies are dealt with respect to specified design patterns.

Unit V

Behavioural Patterns: Memento, Observer, State, Strategy, Template Method, Visitor. UML modelling for different problem scenarios are illustrated.

Tutorial:

Exercises and Mini-projects based on concepts & tools will be solved in the tutorial classes. (Report submission and manual preparation)

References:

1. Len Bass, Paul Clement, Rick Kazman, Software Architectures in Practice, 3rd Edition, Pearson, 2013.
2. Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, Michael Stal, Pattern Oriented Software Architecture: A System of Patterns, John Wiley and Sons, Volume 1, Reprinted February 2001.

3. Alan Shalloway, James R Trott, Design Patterns Explained, A New Perspective on Object Oriented Design, 2nd Edition, Addison Wesley 14.
4. Mary Shaw and David Garlan, Software Architecture-Perspectives on an Emerging Discipline, PHI Learning, 2007.
5. James W Cooper, Java Design Patterns, A Tutorial, Addison Wesley.
6. Eric Freeman, Elisabeth Freeman, Head First Design Patterns, O'reilly Publications.

Course Outcomes (COs):

At the end of the course, students will be able to-

1. Summarize the need for software architecture and the principles of the classic architectural styles (PO-1, 3, 4).
2. Classify some of the challenging design issues that software engineers face and the trade-offs associated with the solutions to these. (PO-1, 2, 3, 4).
3. Describe the principles behind structural patterns and be able to apply a number of the fundamental patterns (PO-1, 2, 3, 4).
4. Outline the major approaches to integrate patterns into software design (PO-1, 2, 3, 4).
5. Demonstrate practical competence in the application and construction of software by applying appropriate architecture and patterns (PO-1, 3, 4).

MACHINE LEARNING

Course Code: MSWE13 **Credit: 4:0:0**
Prerequisite: Linear Algebra **Contact Hours: 56L**
Course Coordinator: Dr. Vijaya Kumar B P

Course Contents

Unit I

Introduction: Introduction to Machine Learning, Why Machine Learning, Introduction to Statistics and Probability **Learning Theory:** Regularization, Bias-Variance Decomposition, over fitting, under fitting and Tradeoff, Generalization and Uniform Convergence,

Unit II

Parametric and non parametric learning, *Supervised Learning:* Regression, classification, Decision Tree, Random Forest, KNN, Logistic Regression. Bayesian Learning methods, Naive Bayesian Classifier, Ensemble learning. Introduction machine learning with Julia or Python.

Unit III

Support Vector Machine: Kernel Method, Gaussian Processes, Learning Evaluation Metrics, Dimensionality Reduction Algorithms, Model selection and curse of dimensionality, PCA, Case studies.

Unit IV

ANN, Multilayer Perceptrons, Introduction to Deep Learning- CNN; Image, Text classification; *Unsupervised Learning* - K-means, KSOM-NN, Case studies for above topics.

Unit V

Reinforcement learning - Q Learning, Hidden Markov Models, Time series; Autoregressive models, Case studies in current developments, issues and future directions in Machine Learning.

Text Book:

1. Tom M Mitchell, Machine Learning, McGraw-Hill, Inc. New York.

References:

1. Christopher Bishop, Pattern Recognition and Machine Learning, CBS Publishers & Distributors-New Delhi.
2. Ethem Alpaydin, Introduction to Machine Learning, 3rd Edition, PHI Pvt. Ltd, 2010.
3. Simon Haykin, Neural networks and Learning Machines, 3rd Edition, PHI.
4. James A. Anderson, An introduction to Neural Networks, PHI.
5. Richard S S and Andrew G Barto, Reinforcement Learning, 2nd Edition, The MIT press, 2018.

Course Outcomes (COs):

At the end of the course, students will be able to-

1. Identify and interpret the concepts and issues of various learning models with lab. practices (PO-1, 3, 4).
2. Illustrate the parametric and non-parametric learning methods and compare different models for classification and regression with implementation (PO-1, 3, 4).
3. Evaluate the learning models generated from data, and interpret the dimensionality issues and their modeling. (PO-1,2, 3, 4).
4. Design and develop ANN perceptron, multilayer approaches for solving learning problems and Unsupervised Learning methods. (PO-1, 3, 4).
5. Describe the Reinforcement learning technique, and apply the algorithms to a real-world problem for optimization (PO-1, 2, 3, 4).

ENTERPRISE APPLICATION DEVELOPMENT-I LAB

Course Code: MSWEL1 **Credit: 0:1:1**
Prerequisite: NIL **Contact Hours: 28T+28P**
Course Coordinator: Dr. Krishnaraj P M

Lab Exercises:

Development of Enterprise Application using J2EE (Java 2 Enterprise Edition)/ .Net / Open Source Based Framework which includes the following activities.

1. Application of Architectural Frameworks & Design Patterns.
2. Client Tier Development.
3. Web Tier Development.
4. Business Tier Development.
5. System Integration with Enterprise Information System Tier.
6. Packaging and Deployment.

The students have to work in groups of three and develop any enterprise applications like the ones listed below:

1. HR Management
   a. Recruitment System.
   b. Staff Appraisal System.
   c. Grievance Redressal System.
2. Finance Management
   a. Billing.
   b. Payroll Processing.
3. Stock Management
   a. Sales and Purchase.
   b. Marketing of Products.
4. Production Management (Increase the QTY and QLTY)
   a. Cost Management.
   b. Quality Assurance and Delivery of Finished Products.
5. Any other problem with prior approval of the faculty.

CIE Evaluation:

Regular lab sessions based on adherence to project plan - 20 Marks
Final Demonstration - 15 Marks
Project Report - 5 Marks
Implementation of changes suggested during examination - 10 Marks
Total - 50 Marks

SEE Evaluation:

Validity of Project Plan, Software Architecture and QA Plans - 10 Marks
Final Demonstration - 15 Marks
Project Report - 5 Marks
Viva - Voce - 5 Marks
Implementation of changes suggested during examination - 15 Marks
Total - 50 Marks

References:

1. Martin Fowler, Patterns of Enterprise Application Architecture (Addison Wesley Signature Series), 1st Edition, 2010 (Reprint).
2. Inderjeet Singh, Beth Stearns, Mark Johnson, Designing Enterprise Applications with the J2EE TM Platform, Second Edition, Addison Wesley,2011 (Reprint).
3. John Kanalakis, Developing .NET Enterprise Applications, 1st Edition, Apress, 2003.
4. Yakov Fain, Enterprise Development with Flex, O'Reilly Series, 1st Edition, 2010.
5. Jansch, PHP/Architect's Guide to Enterprise PHP Development, Musketeers.me, LLC, 2008.
6. Steven Holzner, PHP: The Complete Reference, McGraw Hill Education, 1st edition (30 November 2007).
7. http://docs.oracle.com/javaee/6/firstcup/doc/gcrky.html
8. http://www.oracle.com/technetwork/developer-tools/jdev/j2eedev/084379.html
9. http://j2eetutorials.50webs.com/
10. http://www.webagesolutions.com/knowledgebase/waskb/waskb017/

Course Outcomes (COs):

At the end of the course, students will be able to-

1. Implement the full stack development framework for web based applications. (PO-1, 3, 4)
2. Develop web applications using modern IDEs. (PO-1, 3, 4)
3. Apply advanced technologies used in application development. (PO-1, 3, 4)
4. Present the outcomes of the project in written and oral forms. (PO-2)
5. Evaluate the tools used in modern web application development. (PO-1, 3, 4, 5)

CRYPTOGRAPHY AND INFORMATION SECURITY

Course Code: MSWEA1 **Credit: 4:0:0**
Prerequisite: NIL **Contact Hours: 56L**
Course Coordinator: Dr Sumana M

Course Contents

Unit I

The state of threats against computers, and networked Systems-Overview of computer security solutions and why they Fail-Vulnerability assessment, firewalls, **VPN's** –Overview of Intrusion Detection and Intrusion Prevention Network and Host-based IDS

Unit II

A General IDS model and taxonomy, Signature-based Solutions, Snort, Snort rules, Evaluation of IDS, cost sensitive IDS **Anomaly Detection Systems and Algorithms**-Network Behavior Based Anomaly Detectors (rate based)-Host-based Anomaly Detectors-Software Vulnerabilities, State transition, Immunology, Payload Anomaly Detection.

Unit III

Case Studies of Cryptography: Denial of service attacks, IP spoofing attacks, Secure inter branch payment transactions, Conventional Encryption and Message Confidentiality, Conventional Encryption Principles, Conventional Encryption Algorithms, Location of Encryption Devices, Key Distribution. Public Key Cryptography and Message Authentication: Approaches to Message Authentication, SHA-1, MD5, Public-Key Cryptography Principles, RSA, Digital, Signatures, Key Management.

Unit IV

DoS attacks. Web server and application vulnerabilities, SQL injection attacks, Vulnerability Analysis and Reverse Engineering, Buffer overflow attacks. Client-side browser exploits, Exploiting Windows Access Control Model for Local Elevation Privilege. Exploiting vulnerabilities in Mobile Application.

Unit V

Ethical hacking process, Hackers behavior & mindset, Maintaining Anonymity, Hacking Methodology, Information Gathering, Active and Passive Sniffing, Physical security vulnerabilities and countermeasures. Internal and External testing. Preparation of Ethical Hacking and Penetration Test Reports and Documents.

Reference:

1. Forouzan, B.A., Cryptography & Network Security. Tata McGraw-Hill Education, 2010.
2. Baloch, R., Ethical Hacking and Penetration Testing Guide, CRC Press, 2015.
3. Kahate, A. Cryptography and Network Security. McGraw-Hill Higher Ed., 2009.
4. Godbole, N., Information Systems Security: Security Management, Metrics, Frameworks and Best Practices. 1st Ed. John Wiley & Sons India, 2009.

Course Outcomes (COs):

At the end of the course, students will be able to-

1. Identify threats on systems and access the vulnerabilities (PO-3,4).
2. Classify the various forms of intrusion detection systems (PO-3, 4).
3. Describe the crypto graphical techniques and their usage (PO-1, 3, 4).
4. Demonstrate different types of attacks and ways to control them. (PO-1, 3, 4).
5. Understand Ethical hacking and its approaches. (PO-1, 3, 4, 5).

INFORMATION RETRIVAL

Course Code: MSWEA2 **Credit: 4:0:0**
Prerequisite: NIL **Contact Hours: 56L**
Course Coordinator: Dr Megha P Arakeri

Course Contents

UNIT-I

Introduction: Motivation, Basic concepts, Past, Present and Future, The retrieval process**. Modelling:** Taxonomy of IR models, Retrieval, Formal characterization of IR models, Classic IR, Structured text retrieval models, Models for browsing

UNIT-II

Query languages: Keyword-based querying, pattern matching, structural queries. **Scoring and Term Weighting:** Term frequency and weighting, Vector space model for scoring

UNIT-III

Text and Multimedia Properties: Metadata, Text, Multimedia. **Text Operations:** Document Preprocessing, Compression (Statistical methods). **Flat Clustering:** Clustering in information retrieval, Problem statement, Evaluation of clustering, K-means clustering

UNIT-IV

Indexing and Searching: Inverted files, Sequential searching (Brute force, Knuth-Mooris-Pratt, Boyer-Moore**. Relevance Feedback and Query Expansion:** Rocchio algorithm, when does relevance feedback work, Evaluation of relevance feedback, Global methods for query reformulation.

UNIT-V

Evaluation in information retrieval: Standard test collections, Evaluation of unranked and ranked retrieval sets, Assessing relevance, A broader perspective. **Multimedia IR:** Generic multimedia indexing approach, 1D time series, 2D color images

Reference Books:

1. Ricardo Baeza-Yates, Berthier Ribeiro-Neto, Modern Information Retrieval, Addison Wesley, 2011

2. Introduction to Information Retrieval. C.D. Manning, P. Raghavan, H. Schütze. Cambridge UP, 2008.
3. William B Frakes, Ricardo Baeza Yates: Information Retrieval Data Structures and Algorithms, Prentice Hall, 1992.

Course Outcomes (COs):

At the end of the course, students will be able to:

1. Describe fundamental concepts of information retrieval and its models (PO - 3, 4)
2. Illustrate query processing and retrieving of relevant information from information storage (PO - 1, 3, 4),
3. Apply preprocessing, compression and clustering operations on the given information (PO - 1, 3, 4)
4. Design and develop index structures and relevance feedback technique for efficient and effective information retrieval (PO - 1, 3, 4)
5. Evaluate performance of information retrieval systems and explain multimedia information retrieval. (PO - 1, 2, 3, 4)

MOBILE COMPUTING

Course Code: MSWEA3 **Credit: 4:0:0**
Prerequisite: Computer Networks, DBMS **Contact Hours: 56L**
Course Coordinator: Dr Vijaya Kumar B P

Course Content

Unit I

Introduction: Challenges in mobile computing, coping with uncertainties, resource poorness, bandwidth, etc. Cellular architecture, co-channel interference, frequency, reuse, capacity increase by cell splitting. Evolution of mobile system: CDMA, FDMA, TDMA, GSM. Wireless LAN: IEEE 802.11.

Unit II

Mobility Management: Cellular architecture, Co-channel interference, Mobility: handoff, types of handoffs; location management, HLR-VLR scheme, Mobile IP, Dynamic host configuration protocol, Mobile transport Layer-Traditional and classical TCP.

Unit III

Databases: Database Hoarding Techniques, Data Caching, Transactional Models, Query Processing. Data Dissemination and Broadcasting Systems: Communication Asymmetry, Classification of Data-Delivery Mechanisms, Data Dissemination Broadcast Models, Selective Tuning and Indexing Techniques.

Unit IV

Data Synchronization in Mobile Computing Systems: Synchronization, Synchronization software for mobile devices, Synchronization protocols, SyncML - Synchronization language for mobile computing, Sync4J (Funambol), Synchronized Multimedia Markup Language (SMIL). Mobile Devices: Server and Management: Mobile agent, Application server, Gateways, Portals, Service Discovery, Device management.

Unit V

Support for Mobility- , Mobile files systems, security, Mobile operating systems; Features, services and interfacing modules : Windows/Android/ iOS for Mobile devices.

Text Books:

1. Raj Kamal, Mobile Computing, Oxford University Press, 2nd Edition, 2012.
2. Jochen Schiller, Mobile Communications, 2nd Edition, Pearson 2003.

References:

1. Reza B, Mobile Computing Principles, Cambridge University press 2005.
2. B. P. Vijay Kumar and P. Venkataram, Prediction-based location management using multilayer neural networks, Journal of The Indian Institute of Science, Vol. 82, No. 1, 2002.
3. B. P. Vijay Kumar and P. Venkataram, A Neural Network–Based Connectivity Management for Mobile Computing Environment, International Journal of Wireless Information Networks, Vol. 10, No. 2, 2003.
4. S.Acharya, M. Franklin and S. Zdonik, Balancing Push and Pull for Data Broadcast, Proceedings of the ACM SIGMOD, Tuscon, AZ, May 1997.
5. S.Acharya, M. Franklin and S. Zdonik, Disseminating Updates on Broadcast Disks, Proceedings of the 22nd VLDB Conference, Mumbai (Bombay), India, Sept 1996.

Course Outcomes (COs):

At the end of the course, students will be able to-

1. Discuss the challenges and issues in mobile computing, and describe the basic principles and techniques, and protocol standards in wireless networks. (PO- 1, 3,4).
2. Describe the concept of network and transport layer for mobile networks in respect to mobility management. (PO-1, 4)
3. Analyze the database handling, data dissemination, synchronization in respect to Mobile data base and computing. (PO-1, 3, 4)
4. Describe and illustrate the mobile services, agents and mobility support using different file systems and platforms. (PO-1,2,3)
5. Develop mobile applications by analyzing their characteristics and Requirements by selecting the appropriate computing models and software architectures. (PO- 2, 3, 4)

PROBABILITY, STATISTICS AND QUEUEING THEORY

Course Code: MSWEB1 **Credit: 4:0:0**
Prerequisite: UG Mathematics **Contact Hours: 56L**
Course Coordinator: Dr N L Ramesh

Course Content

Unit I

Random Variables: Random Variables (Discrete and Continuous), Probability density function, Cumulative distribution function, Mean, Variance, Moment generating function.

Discrete Probability Distributions: Binomial distribution, Poisson distribution,

Unit II

Continuous probability distributions. Normal distribution, Exponential distribution, Gamma distribution and Uniform distribution. **Joint probability distribution**: Joint probability distribution (both discrete and continuous), Conditional probability, Conditional expectation, Simulation of random variable.

Unit III

Stochastic Processes: Introduction, Classification of stochastic processes, Discrete time processes, Stationary, Ergodicity, Autocorrelation, Power spectral density. **Markov Chain:** Probability Vectors, Stochastic matrices, Regular stochastic matrices, Markov chains, Higher transition probabilities, Stationary distribution of Regular Markov chains and absorbing states, Markov and Poisson processes.

Unit IV

Queuing theory: Introduction, M/M/1 With infinite and finite capacity, M/M/K systems, M/G/1 queuing system, Engineering applications

Unit V

Sampling and Statistical Inference: Sampling, Sampling distributions, Standard error, Weak law of large numbers (without proof), Central limit theorem, Basics of parametric estimation, Test of Hypothesis for means, Confidence limits for means, Z-test,Test of significance of means and difference of means for large samples, Student's t-distribution, F-distribution, Chi-square distribution as a test of goodness of fit.

Text Books:

1. B.S.Grewal , Higher Engineering Mathematics , Khanna Publishers , 44th edition-2017.
2. R.E. Walpole, R. H. Myers, R. S. L. Myers and K. Ye, Probability and Statistics for Engineers and Scientists, Pearson Education, Delhi, 9th edition, 2012.

References:

1. Sheldon M. Ross, Introduction to Probability models, Academic Press ,2009.
2. Kishor S. Trivedi, Probability & Statistics with Reliability, Queuing and Computer Science Applications, John Wiley & Sons, 2nd edition, 2008.
3. T.Veerarajan , Probability, Statistics and Random process .

Course outcomes (COs):

At the end of the course, students will be able to-

1. Analyze the given random data and their discrete probability distributions. (PO-1, 3)
2. Analyze the given random data, their continuous probability distributions and calculate the marginal & conditional distributions of bivariate random variables. (PO-1, 3)
3. Determine the parameters of stationary random processes and use Markov chain in prediction of future events. (PO-1, 3)
4. Determine the parameters involved in queuing models. (PO-1, 3)
5. Perform test of hypothesis for a population parameter. (PO-1, 3)

ADVANCED ALGORITHMS

Course Code: MSWEB2 **Credit: 4:0:0**
Prerequisite: NIL **Contact Hours: 56L**
Course Coordinator: Dr Sumana M

Course Contents

Unit I

Introduction: The Role of Algorithms in Computing – Getting started- Growth of functions – Recurrences – Probabilistic Analysis and Randomized algorithms – Heap sort – Quick Sort – Sorting in Linear Time – Elementary data structures – Binary Search Trees – Red black Trees.

Unit II

Advanced Design and Analysis Techniques: Dynamic Programming – Greedy algorithms – Amortized analysis – B-Trees – Binomial Heaps – Fibonacci Heaps. **Graph Algorithms**: Elementary Graph Algorithms – Minimum Spanning Trees – Single source Shortest path – All pairs Shortest paths.

Unit III

Sorting Networks – Matrix operations – Linear Programming – Polynomials and the FFT – Number – Theoretic Algorithms – String Matching – Computational Geometry – NP-Completeness – Approximation algorithms.

Unit IV

Parallel Algorithm Introduction – PRAM Model - Pointer Jumping - Performance evaluation of PRAM Algorithms - Comparison of PRAM Models - Sorting Machine - Relevance of the PRAM Model - **Sorting Networks:** Odd-Even Merge Sort - Sorting on a One-Dimensional Network.

Unit V

Algorithms on a Ring of Processors: Matrix-Vector Multiplication - Matrix-Matrix multiplication - A First Look at Stencil Applications - LU Factorization - A Second Look at Stencil Applications - Implementing Logical Topologies - Distributed vs. Centralized Implementations, Summary of Algorithmic Principles - **Algorithms on Grids of Processors** Logical Two-Dimensional Grid Topologies - Communication on a Grid of Processors - Matrix Multiplication on a Grid of Processors - Two-Dimensional Block Cyclic Data Distribution.

References:

1. Thomas H. Cormen, Charles E. Leiserson, Introduction to Algorithms, 2nd Edition, PHI, 2009.
2. Henri Casanova, Arnaud Legrand, Yves Robert; Parallel Algorithms, CRC press.

Course Outcomes (COs):

At the end of the course, students will be able to-

1. Demonstrate the sorting and searching algorithms as well as recurring functions learnt and analyze them. (PO -1, 3, 4)
2. Apply the advanced algorithm design and analysis techniques learnt as well as graphical algorithms studied to write simple programs. (PO-1, 4)
3. Identify and solve the computational aspects involved in matrix operations, polynomials and FFT number, NP-Completeness, computational geometry etc. (PO-1, 3, 4).
4. Demonstrate the parallel programming algorithms learnt such as PRAM, pointer jumping etc as well as identify sorting networks techniques. (PO-1, 3, 4)
5. Apply the various algorithms for ring and grid processors to write simple programs. (PO-1, 3, 4)

WEB SERVICES

Course Code: MSWEB3 **Credit: 4:0:0**
Prerequisite: NIL **Contact Hours: 56L**
Course Coordinator: Jagadish D Sai

Course Content

Unit I

What are Web Services? Why are Web Services Important; Web Services & Enterprises; Moving Forward; **Service-Oriented Architecture:** Service Orientation in daily life; Evolution of SOA; Drivers for SOA; Dimensions of SOA; Key Components of SOA; Perspectives of SOA; **Enterprise-Wide SOA:** Considerations for Enterprise-Wide SOA, Strawman Architecture for Enterprise-Wide SOA, Enterprise SOA Layers, Application Development Process, SOA Methodology for Enterprise.

Unit II

XML Fundamentals: XML: The Lingua Franca of Web Services; XML Documents; XML Namespaces; XML Schema; Processing XML; **Introduction to XML:** Document Type Definitions, Namespaces, XML Schemas, Displaying Raw XML Documents, Displaying XML Documents with CSS, XSLT Style sheets, XML Processors

Unit III

SOAP and WSDL: The SOAP Model; SOAP; SOAP messages; SOAP Encoding; SOAP RPC;
WSDL; Using SOAP and WSDL; **Mobile and Wireless:** Mobile Wireless Services, Challenges with Mobile, J2ME Web Services

Unit IV

UDDI-Universal Description, Discovery and Integration: UDDI at a Glance; The UDDI Business Registry; UDDI Under the Covers; Accessing UDDI; How UDDI is Playing Out. **Transactions:** ACID Transactions, Distributed Transactions and Two-Phase Commit, Dealing with Heuristic Outcomes, Scaling Transactions to Web Services

Unit V

Workflow: Business Process Management, Workflows & Workflow Management Systems, Workflow Management Systems Drawbacks, Web Services and Workflow; BPEL: BPEL Stack, Activities, Service Linking, Partners and Service References, Message Properties and Property Aliases, Correlating Messages, Containers and Data Handling, Workflow Example: Online Shop(Customer Web Service Not Included)

References:

1. Sandeep Chatterjee, Jeames Webber, Developing Enterprise Web Services – An Architect's Guide, Pearson Education, First Indian Reprint 2004
2. Robert W Sebesta, Programming the World Wide Web, Pearson Education, 4th Edition, Second Impression 2009
3. Shankar Kambhampaty, Service Oriented Architecture for Enterprise Applications, Wiley India Pvt Ltd, First Indian Edition 2008.
4. James McGovern et al: Java web Services Architecture, Elsevier, 2003
5. Thomas Erl, Service Oriented Architecture: Concepts, Technology and Design, Pearson Eductatin, Second Impression, 2008
6. Ben Margolis, SOA for the business developer, Shroff Publishers and Distributors Pvt Ltd (SPD), First Indian Reprint November 2007

Course Outcomes (COs):

At the end of the course, students will be able to-

1. Illustrate Service Oriented Architecture (SOA) concepts and discuss how Web Services are the implementation of SOA. (PO-1,3,4)
2. Demonstrate XML concepts of DTD, Schema, CSS etc and discuss the importance of XML as a meta-data exchange language of Web Services. (PO-1,3,4)
3. Identify the technical details associated with WSDL and SOAP, two aspects vital for Web Services implementation and discuss the implementation of web services for mobile and wireless. (PO-1,3,4)
4. Illustrate the association of Web Services and UDDI as well as discuss the complexities associated with Transactions. (PO-1,3,4)
5. Identify Business Process Workflows and illustrate the use of BPEL. (PO-1,3,4)

II Semester

SOFTWARE METRICS AND QUALITY ENGINEERING

Course Code: MSWE21 **Credit: 4:0:0**
Prerequisite: NIL **Contact Hours: 56L**
Course Coordinator: Dr Naresh E

Course Content

Unit I

Fundamentals of measurements, Measurements- what is it and why do it? – Measurement in everyday life in Software Engineering, Scope of software metrics, the basics of measurements - Representational theory of measurement, measurement and models, measurement scale and scale types, a goal framework for software measurement – Classifying software measures, determining what to measure, applying the framework, software measurement validation and in practice.

Unit II

Software metrics data collection – What is good data, how to define the data, how and when to collect data, how to store and extract data, Analysing software measurement data – Analysing the results of experiments, examples of simple analysis techniques, more advanced methods, overview of statistical tests. Measuring internal software attributes: size – Aspects of software size, length, reuse, functionality, complexity.

Unit III

Software Quality - Perspective and Expectations, A Perspective on Testing, Basic definitions, Test Scenarios, Test cases, Insights from a Venn diagram, identifying test cases, Error, fault and Failure taxonomies, testing throughout the SDLC, Levels of testing, Activities of Test engineer, Test/Debug life cycle, testing principles, the cost of bugs, what makes a good software tester? Testing Axioms. Examples: The triangle problem, The Next Date function, the commission problem, The SATM (Simple Automatic Teller Machine) problem, the currency converter.

Unit IV

Functional Testing: Boundary value analysis, Robustness testing, Worst-case testing, Special value testing, Examples, Random testing, Equivalence classes, Equivalence test cases for the triangle problem, Decision tables, Test cases for the triangle problem. Compatibility testing, Usability testing, website testing, Testing the documentation. Case studies.

Unit V

Coverage-based Testing: Statement coverage testing, Condition coverage testing, Path coverage, computing cyclomatic complexity, exploratory testing. Static Testing: Reviews, Types of reviews, Inspections, Inspection process, Inspection roles, benefits of inspection, Checklists. Test Planning, Test Management, Test Process, Test Reporting. What is Test Automation? Terms used in Automation, Skills needed for automation, what to automate.

References:

1. Norman E. Fenton and Shari Lawrence Pfleeger, Software Metrics: A Rigorous Approach, PWS; 2nd edition, 1998.
2. Paul C. Jorgensen: Software Testing, A Craftsman's Approach, 3 Edition, Auerbach Publications, 2008.
3. Srinivas Sesikan and Ramesh Gopalswamy, Principles of Software Testing, Pearson Education.
4. Stephan H. Kan, Metrics and Models in Software Quality Engineering, Second Edition, Pearson Education.

Course Outcomes (COs):

At the end of the course, students will be able to-

1. Identify the basics of measurement theory and its application to software (PO-1, 3, 4)
2. Measure the internal and external attributes of software (PO-3, 4)
3. Gain the knowledge of the basic definitions/concepts of Quality engineering and software testing. (PO-1)
4. Apply the concepts of validation and its techniques like boundary value analysis, equivalence class partitioning and decision table testing. (PO- 5)
5. Analyze the verification techniques like Reviews, Walkthroughs, checklists and Inspections in the development of software. Prepare the reports to track and monitor the defects. (PO-1, 4, 5)

SOFTWARE PROJECT MANAGEMENT

Course Code: MSWE22 **Credit: 3:1:0**
Prerequisite: NIL **Contact Hours: 42L+14T**
Course Coordinator: Dr Sumana M

Course Content

Unit I

Introduction, Contract & Technical project Management, Activities, Plans, Methods, Methodologies, objectives, business case, Success, failure, Management control, Traditional vs Modern project management, Project portfolio management, Project evaluation, Cost-benefit evaluation Techniques, Risk Evaluation, Resource allocation, Strategic management, Benefits, Step Wise Project Planning.

Unit II

Build/Buy? Methodologies, software processes, process models, prototyping, Incremental delivery, Atern, RAD, Agile methods, XP, Scrum, Selection of process model. Basis for software effort estimation, models, Expert judgment, Estimation by analogy, Albrecht FPA, FP Mark II, COSMIC FFP, COCOMO II, Cost estimation, Staffing pattern, Schedule compression, Capers Jones rules, When activity planning? Project schedules & activities, Sequencing & scheduling activities, Network Model, Time, Forward & Backward Pass, Critical path, Activity float, Shorten project duration, Critical activities, Activity on Arrow networks.

Unit III

Categories of risk, deal with risk, Risk identification, assessment, planning, Management. Evaluation of risks to the schedule, PERT, Monte Carlo, Critical chain. Nature of Resources, resource requirements, scheduling resources, creating critical paths, Counting the cost, Publish resource schedule, cost schedules, scheduling sequence.

Unit IV

Monitoring and control Framework, collect data, Review, Project termination, progress, cost monitoring, Earned Value Analysis, Prioritizing Monitoring, get project back to target, Change control, Software Configuration Management, Managing contracts, Stages, terms of contract, contract management, Acceptance.

Unit V

Managing people, understanding behavior, Organizational behavior, Selecting the right person, Best methods, Motivation, The Oldham-Hackman model, Stress, Ethical concerns. Becoming a team, Decisions, Organizational and Team structures, Coordination, Dispersed and Virtual teams, Communication genres, Communication plans, Leadership. Place and importance of quality, ISO 9126, Product and Process metrics, Product vs Process quality, Quality management systems, CMM, Enhance quality, Testing, Reliability, Quality plans.

Tutorials: Exercises and problems based on project management concepts will be solved in the tutorial classes.

References:

1. Bob Hughes, Mike Cotterell, Software Project Management, 4th Edition, Tata McGraw Hill Publications, 2006.
2. Kathy Schwalbe, Information Technology Project Management, 5th Ed, Thompson, 2006.
3. Watts S. Humphrey, Managing the Software Process, Addison-Wesley, 1989.

Course Outcomes (COs):

At the end of the course, students will be able to-

1. Classify and evaluate projects with illustrations. (PO-1, 2, 3, 4)
2. Select a project approach, estimate effort and plan activities by analysis and evaluation. (PO-1, 3, 4)
3. Analyze, evaluate and manage risks and allocate resources. (PO-1, 3, 4)
4. Analyze, evaluate, monitor and control projects and manage contracts. (PO-1, 3, 4)
5. Manage people, team work and manage software quality with illustrations. (PO-2, 3, 4, 5)

BIG DATA AND CLOUD COMPUTING

Course Code: MSWE23 **Credit: 4:0:0**
Prerequisite: NIL **Contact Hours: 56L**
Course Coordinator: Dr Siddesh G M

Course Content

Unit I

Types of Digital Data: Classification of Digital Data. **Introduction to Big Data:** Introduction to Big Data, Characteristics of Data, Definition of Big Data, Challenges with Big Data, What is Big Data?, Other Characteristics of Data Which are not Definitional Traits of Big Data, Why Big Data?, Are We Just an Information Consumer or Do We also Produce Information?, Traditional Business Intelligence (BI) versus Big Data, A Typical Data Warehouse Environment, A Typical Hadoop Environment, What is New Today?, What is Changing in the Realms of Big Data?. **Big Data Analytics:** What is Big Data Analytics?, What Big Data Analytics Isn't?, Why this Sudden Hype Around Big Data Analytics?, Classification of Analytics, Greatest Challenges that Prevent Businesses from Capitalizing on Big Data, Top Challenges Facing Big Data, Why is Big Data Analytics Important?, What Kind of Technologies are we Looking Toward to Help Meet the Challenges Posed by Big Data?, Terminologies Used in Big Data Environments, Basically Available Soft State Eventual Consistency (BASE)

Unit II

The Big Data Technology Landscape: NoSQL (Not Only SQL), Hadoop, **Introduction to Hadoop:** Introducing Hadoop, Why Hadoop?, Why not RDBMS?, RDBMS versus Hadoop, Distributed Computing Challenges, History of Hadoop, Hadoop Overview, Use Case of Hadoop, Hadoop Distributors, HDFS (Hadoop Distributed File System), Processing Data with Hadoop, Managing Resources and Applications with Hadoop YARN (Yet Another Resource Negotiator), Interacting with Hadoop Ecosystem

Unit III

Introduction to Hive: What is Hive?, Hive Architecture, Hive Data Types, Hive File Format, Hive Query Language (HQL), RCFile Implementation, SerDe, User-Defined Function (UDF). **Introduction to Pig:** What is Pig?, The Anatomy of Pig, Pig on Hadoop, Pig Philosophy, Use Case for Pig: ETL Processing, Pig Latin Overview, Data Types in Pig, Running Pig, Execution Modes of Pig, HDFS Commands, Relational Operators, Eval Function, Complex Data Types, Piggy Bank, User-Defined Functions

(UDF), Parameter Substitution, Diagnostic Operator, Word Count Example using Pig, When to use Pig?, When not to use Pig?, Pig at Yahoo!, Pig versus Hive. **Introduction to Cassandra:** Apache Cassandra – An Introduction, Features of Cassandra, CQL Data Types, CQLSH, Keyspaces, CRUD (Create, Read, Update, and Delete) Operations, Collections, Using a Counter, Time to Live (TTL), Alter Commands, Import and Export, Querying System Tables, Practice Examples.

Unit IV

Introduction Cloud Computing: an old idea, whose time has come, Cloud Computing delivery models & Services, Ethical issues, Cloud vulnerabilities, Challenges. **Cloud Infrastructure:** Amazon, Google, Azure & online services, open source private clouds. Storage diversity and vendor lock-in, Responsibility sharing.

Unit V

Cloud Computing: Applications & Paradigms: Architectural styles of cloud applications, Workflows coordination of multiple activities, Coordination based on a state machine model -the Zoo Keeper. **A case study:** Grep The Web application. Cloud Resource Virtualization: Virtualization, Layering and virtualization, Virtual machines, Full virtualization and paravirtualization, Hardware support for virtualization Case study: Xen -a VMM based on paravirtualization.

Text Book/ References:

1. Big Data Analytics, Seema Acharya and Subhashini Chellappan. Wiley India Pvt. Ltd. 2015.
2. Cloud Computing: Theory and Practice, Dan Marinescu, MK Publishers, 1st edition, 2013.
3. Network Data Analytics, Siddesh G M et.al., Springer, 2018.

Course Outcomes (COs):

At the end of the course, students will be able to-

1. Understand the requirements for and constraints in Big Data ecosystem. (PO-2, 3, 4)
2. Apply concepts of Hadoop/Map-reduce framework for solving typical Big data problems. (PO-2, 3, 4)
3. Identify and use appropriate storage / database platforms in handling Big data storage. (PO-2, 3, 4)
4. Comprehensively understand the cloud computing infrastructure. (PO-2, 3, 4)
5. Analyze different styles of implementing cloud solutions along with different virtualization techniques. (PO-2, 3, 4)

RESEARCH METHODOLOGY

Course Code: MSWE24 **Credit: 2:0:0**
Prerequisite: NIL **Contact Hours: 28L**
Course Coordinator: Dr Megha P Arakeri

Course Content

Unit I

Introduction to Research Methodology: Objectives, motivation and significance of research, Types of research, Research approaches, Research Methods versus Methodology, Research and Scientific Method, Research Process, Criteria of Good Research.

Unit II

Defining the research problem: What is a Research Problem, Selecting the problem, Necessity of defining the problem, Technique Involved in Defining a Problem, Importance of literature review in defining a problem, Survey of literature, Identifying gap areas from literature review

Unit III

Research Design: Meaning of research design, Need for research design, Features of a Good Design, Important Concepts Relating to Research Design, Different Research Designs, Basic Principles of Experimental Designs

Unit IV

Methods of Data Collection: Collection of Primary Data, Observation Method, Interview Method, Collection of Data through Questionnaires and interviews, Some Other Methods of Data Collection, Collection of Secondary Data, Selection of Appropriate Method for Data Collection, Difference between Survey and Experiment

Unit V

Interpretation and Report Writing: Techniques of interpretation, Significance of report writing, Different steps in writing report, Layout of the Research Report, Types of Reports, Mechanics of Writing a Research Report, Oral presentation**,** Tools: Latex, Mendeley, Plagiarism check tool.

Text Book/ References:

1. C. R. Kothari, Research Methodology – Methods and Techniques, New Age International Publishers, 2004.
2. William M. Trochim, James P. Donnelly, Research Methods, 2nd Edition, Cengage India, 2016.
3. Writing the literature review: A practical guide, Sara Efrat Efron, Ruth Ravid, The Guilford Press, 2019
4. Introducing Research Methodology: A Beginner's Guide to Doing a Research Project, Uwe Flick, Sage Publications, 2011

Course Outcomes (COs):

At the end of the course, students will be able to-

1. Explain basic principles, techniques and significance of research. (PO-1, 2, 3, 5)
2. Identify existing research issues through literature review and define a research problem based on the identified issues. (PO-1, 3, 4)
3. Design conceptual research structure for collection, measurement and analysis of data. (PO-1, 3, 4)
4. Describe different data collection techniques for carrying out research work. (PO-1, 3, 4, 5)
5. Develop skills related to effective technical report writing. (PO-2, 3, 5)

ENTERPRISE APPLICATION DEVELOPMENT-II LAB

Course Code: MSWEL2 **Credit: 0:1:1**
Prerequisite: NIL **Contact Hours: 28T+28P**
Course Coordinator: Dr. Krishnaraj P M

Lab Exercises:

Students have to work in groups of three to develop an application using DevOps tools and demonstrate the continuous integration, version management and change management in AWS / GCS ecosystem.

CIE Evaluation:
Regular lab sessions based on adherence to project plan - 20 Marks
Final Demonstration - 15 Marks
Project Report - 5 Marks
Implementation of changes suggested during examination - 10 Marks
Total - 50 Marks

SEE Evaluation:
Validity of Project Plan, Software Architecture and QA Plans - 10 Marks
Final Demonstration - 15 Marks
Project Report - 5 Marks
Viva - Voce - 5 Marks
Implementation of changes suggested during examination - 15 Marks
Total - 50 Marks

Course Outcomes (COs):

At the end of the course, students will be able to-

1. Implement applications using DevOps framework. (PO-1, 3, 4)
2. Develop and Integrate applications using modern IDEs. (PO-1, 3, 4)
3. Learn build and control management systems used in application development. (PO-1, 3, 4)
4. Present the outcomes of the project in written and oral forms. (PO-2)
5. Evaluate the tools used in modern application development. (PO-1, 3, 4, 5)

INTERNET OF THINGS

Course Code: MSWEC1 **Credit: 3:0:0**
Prerequisite: NIL **Contact Hours: 42L**
Course Coordinator Dr. Vijaya Kumar B P

Course Contents

Unit I

Introduction to Internet of Things Definition & Characteristics of IoT, Physical Design of IoT, Things in IoT, IoT Protocols, Logical Design of IoT, IoT Functional Blocks, IoT Communication Models, IoT Communication APIs, IoT Enabling Technologies, Wireless Sensor Networks, Cloud Computing Big Data Analytics, Communication Protocols, Embedded Systems, IoT Levels and Deployment Templates, IoT Level-1, IoT Level-2, IoT Level-3, IoT Level-4, IoT Level-5, IoT Level-6.

Unit II

IoT and M2M : Introduction,M2M, Difference between IoT and M2M, SDN and NFV for IoT, Software Defined Networking, Network Function Virtualization, IoT System Management with NETCONF-YANG, Need for IoT Systems Management, Simple Network Management Protocol (SNMP), Limitations of SNMP, Network Operator Requirements, NETCONF, YANG IoT Systems Management with NETCONF-YANG, NETOPEER .

Unit III

IoT Platforms Design Methodology: IoT Design Methodology, Purpose & Requirements Specification, Process Specification, Domain Model Specification, Information Model Specification, Service Specifications, IoT Level Specification, Functional View Specification, Operational View Specification, Device & Component Integration, Application Development, **IoT Systems** - Logical Design using Python, Functions Modules, Packages, File Handling Operations Classes, Python Packages of Interest for IoT, JSON, XML, HTTPLib & URLLib, SMTPLib.

Unit IV

IoT Physical Devices & Endpoints, Embedded Boards, Interfaces, Serial SPI, I2C, Programming with Python, Controlling LED with embedded processor (like Raspberry Pi, aurdino, etc.,) Interfacing an LED and Switch, Interfacing a Light Sensor (LDR), Other IoT Devices, pcDuino, Beagle Bone Black, Cubie board. IoT Physical Servers & Cloud Offerings, Cloud storage Models & communication API's, WAMP - AutoBahn for IoT, Xively Cloud for IoT, Python Web Application Framework – Django, Django Architecture, Starting Development with Django, Designing a RESTful Web API,Web Services for IoT, IoT Messaging Platform.

Unit V

Case Studies Illustrating IoT Design, Home Automation Smart Lighting, Home Intrusion Detection, Smart Parking, Weather Monitoring System 9.4.2 Weather Reporting, Air Pollution Monitoring, Forest Fire Detection, Agriculture, Smart Irrigation, Productivity Applications.

Data Analytics for IoT, Apache Hadoop, Map Reduce Programming Model, Hadoop Map Reduce Job Execution, Map Reduce Job Execution Workflow, Hadoop Cluster Setup, Using Hadoop Map Reduce for Batch Data Analysis, Hadoop YARN, Apache Oozie, Setting up Oozie, Oozie Workflows for IoT Data Analysis, Apache Spark, Apache Storm, Setting up a Storm Cluster, Using Apache Storm for Real-time Data Analysis, REST-based approach, Web Socket-based approach

Text Books:

1. Arshdeep Bagha, Vijay Madisetti, Internet of Things (A Hands-on-Approach) by University press Aug 2015.
2. Dirk Slama, Frank Puhlmann, Jim Morrish, Rishi M Bhatnagar, Enterprise IoT: Strategies and Best Practices for Connected Products and Services. O'Reilly Media, Sep 2015.
3. Hillar, Internet of Things with Python, Packet book.

Course Outcomes (COs):

At the end of the course, students will be able to:

1. Explain the design principles and issues involved in IoT and their Standards. (PO-3, 4, 5)
2. Identify the networking functions and the management of IoT systems.(PO-3, 4)
3. Distinguish and apply the design techniques with cloud and embedded solution for IoT. (PO-1, 3, 4)
4. Illustrate the implementation and interfacing methodologies for IoT frameworks. (PO-3, 4)
5. Design and illustrate the application specific IoT system and perform data analytics, along with the project reports. (PO-1, 2, 3, 4)

DATA SCIENCE

Course Code: MSWEC2 **Credit: 3:0:0**
Prerequisite: NIL **Contact Hours: 42L**
Course Coordinator Savita K Shetty

Course Content

Unit I

Introduction to Big Data Analytics: Big Data Overview, Data Structures, Analyst Perspective on Data Repositories, State of the Practice in Analytics, BI Versus Data Science, Current Analytical Architecture, Drivers of Big Data, Emerging Big Data Ecosystem and a New Approach to Analytics, Key Roles for the New Big Data Ecosystem, Examples of Big Data Analytics. **Data Analytics Lifecycle:** Data Analytics Lifecycle Overview, Key Roles for a Successful Analytics Project, Background and Overview of Data Analytics Lifecycle, Six phases in Data Analytics lifecycle. Case study.

Unit II

Introduction to Data Science: The data science process, the roles in a data science project, Stages of a data science project, Setting expectations. Loading data into R: Working with data from files, working with relational databases. Exploring data: Using summary statistics to spot problems, spotting problems using graphics and visualization, managing data: Cleaning data, Sampling for modeling and validation**.**

Unit III

Modeling Methods using R: Choosing and evaluating models: Mapping problems to machine learning tasks , Evaluating models ,Validating models , Memorization methods : Building single-variable models ,Building models using many variables , Linear and logistic regression: Using linear regression , Using logistic regression, Unsupervised methods :Cluster analysis, Association rules , Exploring advanced methods : Using bagging and random forests to reduce training variance ,Using bagging to improve prediction , Using random forests to further improve prediction

Unit IV

Delivering Results*:* Documentation and deployment: The buzz dataset, using knitr to produce milestone documentation, using comments and version control for running documentation, deploying models, producing effective presentations, presenting your

results to the project sponsor, presenting your model to end users, Presenting your work to other data scientists.

Unit V

Advanced Analytical Theory and Methods: Time Series Analysis: Overview, ARIMA model. Text Analysis: Steps, Example, collecting raw text, representing text, Term Frequency, Categorizing Documents by topics, Determining Sentiments, Technology and Tools: Analytics for Unstructured Data, In-Database Analytics – SQL Essentials, Text Analysis.

References:

1. Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data. John Wiley & Sons, 2015.
2. Zumel N., Mount J., Practical data science with R, 2014.

Course Outcomes (COs):

At the end of the course, students will be able to:

1. Utilize the concepts of big data, its life cycle, Business intelligence, Analytics and statistics. (PO-2, 3, 4)
2. Retrieve, organize and manipulate data using a variety of analytical tools. (PO-1, 2, 3, 4)
3. Design and evaluate modeling methods using R. (PO-1, 2, 3, 4)
4. Deliver the results using appropriate documentation, plots and deployment models. (PO-2, 3, 4)
5. Analyze and represent time series and text data. (PO-2, 3, 4)

ADVANCES IN OPERATING SYSTEMS

Course Code: MSWEC3 **Credit: 3:0:0**
Prerequisite: NIL **Contact Hours: 42L**
Course Coordinator Dr. Sumana M

Course Content

Unit I

Operating System Overview, Process description & Control: Operating System Objectives and Functions, The Evolution of Operating Systems, Major Achievements, Developments Leading to Modern Operating Systems, Microsoft Windows Overview, Traditional UNIX Systems, Modern UNIX Systems, what is a Process? Process States, Process Description, Process Control, Execution of the Operating System, Security Issues.

Unit II

Threads, SMP, and Microkernel, Virtual Memory: Processes and Threads, Symmetric Multiprocessing (SMP), Micro Kernels, Windows Vista Thread and SMP Hours Management, Linux Process and Thread Management. Hardware and Control Structures, Operating System Software, UNIX Memory Management, Windows Vista Memory Management, Summary.

Unit III

Multiprocessor and Real-Time Scheduling: Multiprocessor Scheduling, Real-Time Scheduling, Linux Scheduling, UNIX PreclsSl) Scheduling, Windows Vista Hours Scheduling, Process Migration, Distributed Global States, Distributed Mutual Exclusion, Distributed Deadlock

Unit IV

Embedded Operating Systems: Embedded Systems, Characteristics of Embedded Operating Systems, eCOS, TinyOS, Computer Security Concepts, Threats, Attacks, and Assets, Intruders, Malicious Software Overview, Viruses, Worms, and Bots, Rootkits.

Unit V

Kernel Organization: Using Kernel Services, Daemons, Starting the Kernel, Control in the Machine, Modules and Device Management, MODULE Organization, MODULE Installation and Removal, Process and Resource Management, Running Process Manager, Creating a new Task, IPC and Synchronization, The Scheduler, Memory

Manager, The Virtual Address Space, The Page Fault Handler, File Management. The windows NT/2000/XP kernel: Introduction, The NT kernel, Objects, Threads, Multiplication Synchronization, Traps, Interrupts and Exceptions, The NT executive, Object Manager, Process and Thread Manager , Virtual Memory Manager, I/o Manager.

Text Books:

1. William Stallings, Operating Systems: Internals and Design Principles, 6th Edition, Prentice Hall, 2013.
2. Gary Nutt, Operating Systems, 3rd Edition, Pearson, 2014.

References:

1. Silberschatz, Galvin, Gagne: Operating System Concepts, 8th Edition, Wiley, 2008
2. Andrew S. Tanenbaum, Albert S. Woodhull: Operating Systems, Design and Implementation, 3rd Edition, Prentice Hall, 2006.
3. Pradeep K Sinha: Distribute Operating Systems, Concept and Design, PHI, 2007.

Course Outcomes (COs):

At the end of the course, students will be able to:

1. Demonstrate the Mutual exclusion, Deadlock detection and agreement protocols of Distributed operating system. (PO-1, 3)
2. Learn the various resource management techniques for distributed systems. (PO-1, 3)
3. Identify the different features of real time and mobile operating system. (PO-1, 3)
4. Modify existing open source kernels in terms of functionality or features used. (PO-1, 3)

COGNITIVE COMPUTING

Course Code: MSWED1 **Credit: 3:0:0**
Prerequisite: NIL **Contact Hours: 42L**
Course Coordinator Dr. Vijaya Kumar B P

Course Content

Unit I

Introduction: Foundation of cognitive computing, nn platforms- machine learning, reasoning, natural language processing, speech recognition and vision (object recognition), human–computer interaction, dialog and narrative generation among other technologies. Features: Adaptive, Interactive, Iterative, Stateful, context aware.

Unit II

Design Principles for Cognitive Systems, Sentiment analysis, Natural language processing, text analysis, computational linguistics, and biometrics to systematically identify, extract, quantify, and study affective states and subjective information. Sentiment analysis to voice of the customer materials such as reviews and survey responses, online and social media, and healthcare materials for applications.

Unit III

Cognitive analytics, Relationship between Big Data and cognitive computing, Principal benefit of utilizing cognitive analytics over traditional big data analytics.
Representing knowledge in taxonomies and ontologies, Applying advanced analytics to cognitive computing, Using machine learning and deep learning neural networks to model cognition.

Unit IV

Role of cloud and distributed computing in cognitive computing, Business implications of cognitive computing, Cognitive analytics- Cognitive computing-technology platforms, Word processing documents, E-mails, videos, images, audio files, presentations, webpages, social media and many other data formats, tagging, analysis and insight generation.

Unit V

Process of building a cognitive application, Building a cognitive healthcare application, Smarter cities: cognitive computing in government, Emerging cognitive computing areas, Future applications for cognitive computing.

References:

1. Judith S. Hurwitz, Marcia Kaufman, Adrian Bowles, Cognitive Computing and Big Data Analytics, Wiley 2015.
2. Adnan Hashmi and Adnan Masood, Cognitive Computing Recipes: Artificial Intelligence Solutions Using Microsoft Cognitive Services and TensorFlow, APress, 2019.
3. Mark Watson, Introduction to Cognitive Computing, 2018, Copyright 2016-2018 Mark Watson.
4. Vijay V Raghavan, Venkat N. Gudivada, Venu Govindaraju, C.R. Rao, Cognitive Computing: Theory and Applications, Elsevier, 2016.
5. Peter Fingar, Vinton G. Cerf, Cognitive Computing: A Brief Guide for Game Changers, Meghan-Kiffer Press, 2015.

Course Outcomes (COs):

At the end of the course, students will be able to-

1. Classify and interpret the features of cognitive system and explain the necessary platforms used for cognitive computing. (PO-2, 3, 4)
2. Design the cognitive systems and illustrate the methodologies used for some of the applications using cognitive computing (PO-2, 3, 4)
3. Analyze and quantify the concepts used to model the cognition and ontologies for knowledge representation and implementation using ANN. (PO-1, 3, 4)
4. Apply and compare the cognitive computing with Cloud and distributed Computing platforms and their implications. (PO-3, 4)
5. Build the cognitive applications in some of social relevant areas. (PO-1, 2, 3, 4)

SYSTEM PERFORMANCE AND ANALYSIS

Course Code: MSWED2 **Credit: 3:0:0**
Prerequisite: NIL **Contact Hours: 42L**
Course Coordinator Dr Sumana M

Course Content

Unit I

AN OVERVIEW OF PERFORMANCE EVALUATION: Introduction, Common Mistakes in Performance Evaluation, A Systematic Approach to Performance Evaluation, selecting an Evaluation Technique, Selecting Performance Metrics, commonly used Performance Metrics, Utility Classification of Performance Metrics, Setting Performance Requirements.

Unit II

WORKLOAD SELECTION AND CHARACTERIZATION: Types of Workloads, Addition instructions, Instruction Mixes, Kernels; Synthetic programs, Application Benchmarks, Popular benchmarks, Workload Selection: Services exercised, level of detail, Representativeness, Timeliness, Other considerations in Workload selection, Workload Characterization & Techniques: Terminology, Averaging, Specifying dispersion, Single Parameter Histograms, Multi parameter histograms, Principal Component Analysis, Markov Models, Clustering.

Unit III

MEASUREMENT TECHNIQUES AND TOOLS: Monitors: Terminology and classification, software and hardware monitors, Software versus hardware monitors, firmware and hybrid monitors, Distributed System Monitors, Program-Execution Monitors and Accounting Logs: Program Execution Monitors, Techniques for Improving Program Performance, Accounting Logs, Analysis and Interpretation of Accounting log data, using accenting logs to answer commonly asked questions.

Unit IV

CAPACITY PLANNING, BENCHMARKING AND EXPERIMENTAL DESIGN: Steps in capacity planning and management, Problems in capacity planning, Common mistakes in benchmarking, Remote-Terminal Emulation, Components of an RTE, Limitations of RTEs, Experimental design and analysis: Terminology, Common mistakes in experiments, Types of Experimental designs, 2^k factorial designs, concepts,

computation of effects, Sign table method for computing effects, allocation of Variance, General 2^k factorial designs.

Unit V

SIMULATION: Introduction to Simulation, Analysis of Simulation Results: Model Verification Techniques: Top-down Modular design, Antibugging, Structured Walk-through, Deterministic Models, Run simplified cases, Trace, Online graphic displays, Continuity Tests, Degeneracy Tests, Consistency Tests, Seed Independence, Model Validation Techniques: Expert Intuition, Real System Measurements, Theoretical Results, Transient Removal: Long Runs, Proper Initialization, Truncation, Initial Date Deletion, Moving Average of Independent Replications, Batch Means, Terminating Simulation, Stopping Criteria, Variance Reduction.

References:

1. Raj Jain, The Art of Computer Systems Performance Analysis, John Wiley and sons, New York, USA, 1991.
2. Law A M and Kelton W.D, Simulation Modeling and Analysis, McGraw Hill, New York, USA, 1991.
3. Paul J Fortier, Howard E Michel, Computer Systems Performance Evaluation and Prediction, Elsevier, 2003.

Course Outcomes (COs):

At the end of the course, students will be able to:

1. Identify the concepts of performance evaluation. (PO-1,3,4)
2. Select proper workload and characterize the workload. (PO-1,3,4)
3. Analyze performance statistics, data and display results using monitors. (PO-1,3,4)
4. Design performance experiments. (PO-1,3,4)
5. Perform computer system performance analysis using simulation. (PO-1,3)

BLOCKCHAIN AND CYBER SECURITY

Course Code: MSWED3 **Credit: 3:0:0**
Prerequisite: NIL **Contact Hours: 42L**
Course Coordinator Dr. Sumana M

Course Content

Unit I

Cyber Threat Landscape and Security Challenges: Current threat landscape, Ransomware, Distributed denial-of-service (DDoS) attacks, Insider threats, Data breaches, Advanced persistence threat (APT), Defender perspectives, Live attack execution, Emerging security challenges. The security ecosystem, The zero-trust approach, The assume breach approach.

Unit II

Introducing Blockchain and Ethereum: What is blockchain? A brief history, Fundamentals of the blockchain ,Who is using blockchain and how? Internet versus blockchain, IP packet versus block ,Web app versus dApp,,How blockchain works, The building blocks of blockchain - Block, Cryptography digital signature and hashing algorithm, Consensus : the core of blockchain, Ethereum: History, What is Ethereum? Smart contract, EVM, Gas, dApp, Private versus public blockchain, Public blockchain, Private blockchain, Business adaptation.

Unit III

Hyperledger: Blockchain for Businesses: Technical requirements, Hyperledger overview, Blockchain-as-a-service (BaaS), Program goal, Architecture and core components , Hyperledger Fabric model, Hyperledeger Fabric core components, Workings of Hyperledger and transaction processing, Bitcoin versus Ethereum versus Hyperledger, Hyperledger Fabric capabilities, Tuna application.

Unit IV

Blockchain on the CIA Security Triad, What is the CIA security triad?, Understanding blockchain on confidentiality, Blockchain on integrity, Understanding blockchain on availability, Deploying PKI-Based Identity with Blockchain, Components, Architecture, Challenges of the existing PKI model, How can blockchain help? Decentralized infrastructure, Deployment method, Testing.

Unit V

Two-Factor Authentication with Blockchain, Blockchain for 2FA, Blockchain-Based DNS Security Platform: DNS, Understanding DNS components, DNS structure and hierarchy: Root name server, Current TLD structure, Registries, registrars, and registrants, DNS records ,DNS topology for large enterprises, Challenges with current DNS, Blockchain-based DNS solution, Deploying Blockchain-Based DDoS Protection: DDoS attacks, Types of DDoS attacks, Facts about Blockchain and Cyber Security.

References:

1. Rajneesh Gupta, Hands-On Cybersecurity with Blockchain: Implement DDoS protection, PKI-based identity, 2FA, and DNS security using Blockchain, Packt Publishing, 2018.
2. Imran Bashir, Mastering Blockchain: Distributed ledger technology, decentralization, and smart contracts explained, 2nd Edition, Packt Publishing, 2018.

Course Outcomes (COs):

At the end of the course, students will be able to:

1. Understand the cyber threat landscape. (PO- 1, 3)
2. Build Blockchains using Ethereum and Hyperledger. (PO-1, 4, 5)
3. Program Blockchain solutions and build Blockchain-based apps for 2FA, and DDoS protection. (PO-1, 3, 4)
4. Develop Blockchain-based PKI solutions and apps for storing DNS entries. (PO-1, 3, 4, 5)
5. Identify the challenges and the future of cybersecurity and Blockchain. (PO-1, 2, 4)

II Semester

DEEP LEARNING

Course Code: MSWEE1 **Credit: 4:0:0**
Prerequisite: NIL **Contact Hours: 56L**
Course Coordinator: Dr Sumana M

Course Content

Unit I

Review of Machine Learning: The Learning machines, Foundations of Neural network and Deep Learning, Data representations for neural networks, tensor operations and gradient based optimization. **Fundamentals of Deep Learning:** Defining Deep Learning, Common Architectural Principles of Deep Networks, Building blocks of Deep Network.

Unit II

Major Architectures of Deep Networks: Unsupervised Pretrained Networks, **Convolutional Neural Networks**- CNN Architecture Overview, input layers, Convolutional layers, common convolutional architectural patterns, configuring convolutional layers, pooling layers, configuring pooling layers, Fully Connected Layers, Applications of CNN's – Modeling handwritten images Using CNN's, Case studies.

Unit III

Recurrent Neural Networks: Modeling the Time dimension, 3D volumetric input, General Recurrent Neural Network Architecture, Network input Data and input layers, output layers and RNN output layer, Training the network, LSTM networks, debugging common issues with LSTM's, Padding and Masking, Evaluation and scoring with masking, variants of Recurrent Network Architecture Domain-specific applications and blended networks.

Unit IV

Recursive Neural Networks: Network Architecture, Varieties of Recursive Neural Networks, Applications of recursive neural networks., Modeling sequence data by using recurrent neural networks

Restricted Boltzmann Machines: Hidden units and modeling available information, Using different units, Regularization, using regularization with RBM's

Unit V

Autoencoders: Introduction, Architecture, implementation, Denoising Autoencoders, Sparse Autoencoders, Use Cases

Tuning Deep Networks: Basic concepts, Step – by – step process of building deep networks.

Deep Learning in Practice: Deep Learning for Computer vision, deep learning for text and sequences, text generation with LSTM, Generating images with variational autoencoders, Generative adversarial networks.

References:

1. Josh Patterson & Adam Gibson, Deep Learning – A Practitioners Approach, O'Reilly, 1st Edition 2017.
2. Francois Chollet, Deep Learning with Python, Manning, 2018.
3. Simon Haykin, Neural networks: A comprehensive foundation, Second Edition, Prentice Hall, New Delhi, 1999, ISBN-81-203-2373-4.
4. Ian Goodfellow, Yoshua Bengio and Aaron Courville, Deep Learning, MIT Press, 2016.

Course Outcomes (COs):

At the end of the course, students will be able to:

1. Explain the essential context and fundamentals concepts of Deep Learning (PO-3, 4)
2. Design and develop convolutional neural network for images. (PO-1,3,4)
3. Design and develop recurrent neural networks for processing sequential data (PO-1, 3,4)
4. Design and develop unsupervised deep learning recursive neural network and restricted Boltzmann machine models. (PO-1, 3,4)
5. Illustrate the use of deep learning models for various applications. (PO-1, 3,4)

BIOINFORMATICS

Course Code: MSWEE2 **Credit: 4:0:0**
Prerequisite: NIL **Contact Hours: 56L**
Course Coordinator: Mr Shashidhara H S

Course Content

Unit I

The genetic material, gene structure and information content, protein structure and function, chemical bonds, molecular biology tools.

Unit II

Dot plots, simple alignments, gaps, scoring matrices, the Needleman and Wunsch algorithm, semiglobal alignments, the Smith and Waterman algorithm, database searches – BLAST and FASTA

Unit III

Patterns of substitutions within genes, estimating substitution numbers, molecular clocks, Molecular phylogenetics, phylogenetic trees, distance matrix methods, maximum likelihood approaches

Unit IV

Parsimony, Inferred Ancestral Sequences, strategies for fast searches – branch and bound and heuristic searches, consensus trees, tree confidence, molecular phylogenies Genomics – 1: Prokaryotic genomes, prokaryotic gene structure, GC content and prokaryotic genomes, prokaryotic gene density, eukaryotic genomes

Unit V

Genomics – 2: Eukaryotic gene structure Open reading frames, GC contents in eukaryotic genomes, gene expression, transposition, repetitive elements, Amino acids, polypeptide composition, secondary structure, tertiary and quaternary structures, algorithms for modeling protein folding

References:

1. Dan E. Krane, Michael L. Raymer, Fundamental Concepts of Bioinformatics, Pearson Education, 2008
2. T K Attwood, D J Parry Smith, Introduction to Bioinformatics, Pearson Education, 2004
3. Gary B. Fogel, David W. Corne, Evolutionary Computation in Bioinformatics, Morgan Kaufmann Publishers.

Course Outcomes (COs):

At the end of the course, students will be able to:

1 Recognize the role of Genetic Material in living organisms and the ways of acquiring DNA sequence using Molecular Biology Tools (PO-2)
2 Solve sequence alignment problems using dynamic programming methods (PO-1, 4)
3 Model the pattern of substitution within homologs (PO-1, 2, 4)
4 Solve phylogenetic problems using character based and distance based phylogeny (PO-1, 4)
5 Identify different parts of prokaryotic and Eukaryotic Genes **(**PO-1,2)

SOFTWARE MINING AND ANALYSIS

Course Code: MSWEE3 **Credit: 4:0:0**
Prerequisite: NIL **Contact Hours: 56L**
Course Coordinator: Mrs Pushpalatha M N

Course content

Unit I

Introduction: Categories of Software Mining, Software engineering tasks that benefit from data mining: Development tasks, Management tasks, Research tasks, Mining software engineering data: the road from here - Targeting software tasks intelligently, lowering the barrier of entry, a word of caution.

Unit II

Specification Mining: A concise Introduction- Introduction, categorization, Mining Finite State Machines, Mining value-based Invariants, Mining Patterns and Rules, Mining Sequence Diagrams.

Unit III

Bug Report Mining: Bug report structure, Bug life cycle, Empirical studies on different types of bug reports, Studies on reproducibility of bug reports, Non-reproducible bugs- Why bugs are marked as NR, Developer's behaviour towards NR bugs, Why NR bugs get fixed, Research methodology- Subject systems, Bug type classification, Approach

Unit IV

Mining Source Code Repositories: Introduction, language models for programming languages- n-Gram Language Models, Information Theory & Language, the github java corpus, properties of a large source code corpus- Predicting Identifiers, Learnability of Identifiers, code analysis using giga-scale models- n-gram Log Probability as a Data-driven Code Complexity Metric, Log Probabilities at a Project Level, Entropy and the Rhino Project: A Case Study

Unit V

Mining Temporal Rules from Program Execution Traces: Introduction, Semantics of Mined Rules, Mining Algorithm- Challenges and Solutions, Algorithm Sketch, Case Studies- JBoss AS Transaction Component, CVS on Jakarta Commons Net.

Reference Books:

1. Mining Software Specifications: Methodologies and Applications edited by David Lo, Siau-Cheng Khoo, Jiawei Han, Chao Liu, CRC Press.
2. Manoel Mendonca, Nancy L. Sunderhaft, “Mining Software Engineering Data: A Survey”, A DACS State-of-the-Art Report.
3. Goyal, Anjali & Sardana, Neetu. (2019). An empirical study of non-reproducible bugs. International Journal of System Assurance Engineering and Management. 10. 10.1007/s13198-019-00850-5.
4. Taylor, Quinn & Giraud-Carrier, Christophe. (2010). Applications of data mining in software engineering. International Journal of Data Analysis Techniques and Strategies. 2. 243-257. 10.1504/IJDATS.2010.034058.
5. M. Allamanis and C. Sutton, "Mining source code repositories at massive scale using language modeling," 2013 10th Working Conference on Mining Software Repositories (MSR), San Francisco, CA, 2013, pp. 207-216, doi: 10.1109/MSR.2013.6624029.
6. Lo, David & Khoo, Siau-cheng & Liu, Chao. (2007). Mining temporal rules from program execution traces.

Course Outcomes (COs):

At the end of the course, students will be able to:

1. Describe the Concepts of Software Mining. (PO 1, 3)
2. Explain Specification Mining. (PO 1, 3)
3. Describe concepts of Bug Report Mining and Analyze the Bug Repositories (PO 1, 3)
4. Analyze and Mine Source code repositories. (PO 1,3)
5. Analyze program execution traces and mine temporal rules. (PO 1,3)

INTERNSHIP / INDUSTRIAL TRAINING

Course Code: MSWE31 **Credit: 0:0:4**
Prerequisite: NIL **Contact Hours: 112P**
Course Coordinator: Dr. Sumana M

Internship Work-flow

1. Students submit the initial details including broad area of work and choice of guide in a prescribed format
2. The PG Coordinator along with Head of the department finalizes the guide allocation process
3. Students are given an option to change the guide with mutual consent by applying through prescribed form
4. Students submit the Internship details to guide on the day of registration
5. Problem statement is submitted to PG Coordinator within one week of registration.
6. Students update their progress on weekly basis
7. Weekly meeting with guide is also recorded.
8. Guide evaluates the student on a regular basis according to the rubrics defined in the worksheet for total of 100 marks which constitutes the final CIE score.
9. Evaluation is based on following criteria
    - Project Management – 15 marks
    - Literature Survey – 10 marks
    - System Analysis – 15 marks
    - Software Design - 15
    - Implementation / simulation - 20
    - Testing / Validation - 15
    - Speaking and Writing Skills – 10 marks
    - Total – 100 marks

Course Outcomes (COs):

At the end of the course students will be able to:

1. Schedule milestones and deliverables using appropriate project management techniques (PO- 1)
2. Formulate the requirements for the proposed system (PO- 3,4)
3. Design, implement and validate the system according to the plan (PO- 1,3,4)
4. Select effective communication strategies within and outside the team (PO- 2)

PROJECT PRELIMINARIES

Course Code: MSWE32 **Credit: 0:3:10**
Prerequisite: NIL **Contact Hours: 84T+280L**
Course Coordinator: Dr. Sumana M

Project Work-flow:

1. Students submit the initial details including broad area of work and choice of guide in a prescribed format
2. The Project Coordinator along with Head of the department finalizes the guide allocation process.
3. Students are given an option to change the guide with mutual consent by applying through prescribed form.
4. Students submit the Project Workbook to guide on the day of registration.
5. Problem statement is submitted to Project Coordinator within one week of registration.
6. Students maintain a blog and update it on weekly basis about their work.
7. Weekly meeting with guide is recorded in the workbook.
8. Guide evaluates the student on a regular basis according to the rubrics defined in the worksheet for total of 100 marks which constitutes the final CIE score.
9. Evaluation is based on following criteria
    - Project Management – 30 marks
    - Literature Survey – 20 marks
    - System Analysis – 30 marks
    - Speaking and Writing Skills – 20 marks
    - Total – 100 marks

Course Outcomes (COs):

At the end of the course students will be able to:

1. Schedule milestones and deliverables using appropriate project management techniques (PO- 1)
2. Compare and contrast the available literature in the context of the project (PO- 3,4)
3. Formulate the requirements for the proposed system (PO-3,4)
4. Select effective communication strategies within and outside the team (PO- 2)

TECHNICAL SEMINAR

Course Code: MSWE33 **Credit: 0:1:0**
Prerequisite: NIL **Contact Hours: 28T**
Course Coordinator: Dr. Sumana M

Seminar Workflow:

1. Each student is allotted a guide by Coordinator in consultation with Head of the Department.
2. The individual guides decide the topic for seminar during the first week of the semester.
3. The student undergoes a semester long independent study of the topic.
4. During the end of the semester, the student gives an oral presentation on the topic.
5. The student also presents a written report to the guide.
6. Evaluation is based on following criteria
    - Relevance of the topic – 20 marks
    - Background Research – 20 marks
    - Quality of Presentation – 20 marks
    - Speaking Skills – 20 marks
    - Writing Skills – 20 marks
    - Total – 100 marks

Course Outcomes (COs):

At the end of the course students will be able to:

1. Study an emerging topic in software engineering and allied areas (PO - 1)
2. Demonstrate an ability to undertake a lifelong, independent study of a topic (PO – 3,4)
3. Communicate effectively among peers and general public in oral and written forms. (PO - 2)

IV SEMESTER

IPR AND CYBER LAWS

Course Code: MSWE41 **Credit: 0:2:0**
Prerequisite: NIL **Contact Hours: 56T**
Course Coordinator: Dr Naresh E

Tutorial Topics:

1. Copyright law in software
2. Application of Patent laws to software
3. Trademarks and other IPR applicable to software
4. IPR related policies by government
5. International treaties governing IPR
6. IT ACT 2000 and its amendments
7. Software related case-laws
8. Cyber crimes
9. Laws applicable to cyberspace
10. Laws related to m-commerce

References:

1. Dr. B. L. Wadhera, Law Relating to Intellectual Property, Universal law Publishing Co. Ltd. 2009.
2. Joan Ruttenberg, Paige von Mehren, Julie Yen, "Intellectual Property and Cyberlaw", Harvard Law School, 2013.
3. Cybercrime law and practice from the institute of company secretaries of India, 2016.
4. Case studies from internet sources.

Course Outcomes (COs):

At the end of the course students will be able to:

1. Describe the evolution of IPR issues in software (PO-5)
2. Infer the socially relevant issues related to software like liberty and privacy (PO-5)
3. Sketch the process of protecting the IPR issues in software (PO-5)
4. Interpret the risks and liabilities of software in context of computer crimes (PO -5)
5. Critique the ethical issues arising from new areas of software usage (PO -3,5)

PROJECT WORK

Course Code: MSWE42 **Credit: 0:0:20**
Prerequisite: Project Preliminaries **Contact Hours: 560P**
Course Coordinator: Dr. Sumana M

Project Work-flow:

1. The work done in the previous semester (MSWE32) is continued
2. Students maintain a blog and update it on weekly basis about their work
3. Weekly meeting with guide is recorded in the workbook
4. Guide evaluates the student on a regular basis according to the rubrics defined in the worksheet for total of 50 marks which constitutes the final CIE score
5. At the end of the semester, an exam is conducted with one internal and one external examiner for 50 marks which constitutes the final SEE score
6. Evaluation is based on following criteria
    - System Design – 15 marks
    - Coding – 15 marks
    - Testing – 10 marks
    - Speaking and Writing Skills – 10 marks
    - Total – 50 marks

Course Outcomes (COs):

At the end of the course students will be able to:

1. Design the software by applying the relevant guidelines. (PO- 3,4)
2. Develop the software by implementing the design. (PO- 1,3,4)
3. Evaluate the quality of software by testing using appropriate techniques. (PO- 4)
4. Demonstrate the project for final assessment and appraise its effectiveness. (PO- 5)