MTECH SYLLABUS – SOFTWARE ENGINEERING

M.Tech. Software Engineering (2015)

Semester - I

Sl. No.	Course code	Course Title	Core/Elective	Credits	Lec.	Lab	Marks
1	CSS3101	Mathematical concepts for computer science	C	4	4	2	100
2	CSS3102	Software Architecture	C	4	4	2	100
3	CSS3103	Agile Project Management	C	4	4	2	50
4		Elective I	E	3	4	0	100
5		Elective II	E	3	4	0	100
			Total for Semester I	18	20	6	500

Electives

CSS 3104: Patterns in Software Engineering
CSS 3105: Agent based Computing
CSS 3106: Human Computer Interaction
CSS 3107: Information Retrieval and Web search
CSS 3108: Functional Programming

Semester - II

Sl. No.	Course code	Course Title	Core/Elective	Credits	Lec.	Lab	Marks
1	CSS3201	Social Network Analytics	C	4	4	2	100
2	CSS3202	Model Driven Architecture	C	4	4	2	100
3	CSS3203	Seminar	C	1	0	2	50
4		Elective III	E	3	4	0	100
5		Elective IV	E	3	4	0	100
5		Elective V	E	3	4	0	100
			Total for Semester II	18	20	6	550

Electives

CSS 3204: Design of Real Time/Embedded Software
CSS 3205: Software Quality Management
CSS 3206: Network Forensics
CSS 3207: Enterprise Application Integration & Business Process Management
CSS 3208: Advanced Data Mining

Semester - III

Sl. No.	Course code	Course Title	Core/Elective	Credits	Lec.	Lab	Marks
1	CSS3301	Project & Viva Voce	C	18	0	15	400

Semester - IV

Sl. No.	Course code	Course Title	Core/Elective	Credits	Lec.	Lab	Marks
1	CSS3302	Project & Viva Voce	C	18	0	25	500

Total credits for Degree: **72**

CSS3101: 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. It introduces linear algebra, graph theory and probability. The course is intended to cover the main aspects which are useful in studying, describing and modelling of objects and problems in the context of computer algorithms and programming languages.

Course Objectives

To understand Linear systems using Linear Algebra
To get deep understanding of stochastic processes and their applications
To study graph theory and its applications using matrix formulation

Course Content

1. Linear Systems: Vector Spaces, Linear Independence and Rank, Basis, Quadratic forms and Semi Definite matrices, Eigen values and Eigen Vectors, LU Decomposition, Orthogonality, Least squares problem, QR decomposition, SVD, Basic Tensor Concepts

2. Probability axioms, Bayes theorem, Random variables and distributions, Expectation and Variance, Covariance and correlation, Moment generating functions, Inequalities: Markov, Chebyshev, Chernoff bound, Laws of large numbers, Central limit theorem

3. Multivariate Distribution, Point estimation: EM algorithm, Mean-square Error, Sufficiency-Completeness, Testing hypotheses, Stochastic Processes and Markov Chains: Markov process, Kolmogorov-Chapman equations, Parameter Estimation

4. Matrix representation of graphs, Hypergraphs, Bipartite graphs, Components, Independent paths, connectivity and cut sets, Graph Laplacian - Random walks - Measures and metrics on graphs

5. Random Graphs : Models, Ramsey theory, Random graphs with general degree distributions Connectivity and Matchings, Diameter of Random Graphs

REFERNCES

1. Fundamentals of Matrix Computations, David S. Watkins, 3rd Edn, John Wiley, (2010)
2. Introduction to Linear Algebra, Gilbert Strang, 4th Edn, Wellesley-Cambridge Press, (2009)
3. Matrix Methods in Data Mining and Pattern Recognition, Lars Elden, SIAM, (2007)
4. Probability and Computing: Randomized Algorithms and Probabilistic Analysis, Michael Mitzenmacher & Eli Upfal, Cambridge University Press, (2005)
5. Networks: An Introduction, Mark Newman, OUP Oxford, (2010)
6. Random Graphs, Bela Bollobas, 2nd Edition, Cambridge University Press, (2001)

CSS3102: SOFTWARE ARCHITECTURE

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

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 Objectives

To understand the relationships between system qualities and software architectures.
To study software architectural patterns and their relationship to system qualities
To know software architecture documentation and reuse

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 an architecture - the ADD method – Evaluating an architecture -ATAM method

3. Documenting the architecture – principles of sound documentation – view types, styles, views – refinement, context diagrams, variability, software interfaces – documenting the behaviour – seven part template

4. 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

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

REFERNCES

1.Software Architecture in Practice(3rd Ed): Len Bass, Pearson (2013)
2.Quantitative approaches for Evaluating Software Architectures:Frame Works and Models, G.Zayaraz : VDM Verlag (2010)
3. Documenting Software Architectures: Views and beyond (2nd Ed): Clements et al, AW (2010)
4.Software Product Line Engineering: Foundations, Principles and Techniques: Klaus Pohl et.al; Springer (2011)

CSS3103: AGILE PROJECT MANAGEMENT

Core/Elective: **Core** Semester: **1** 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 project management in the agile way.

Course Objectives

To provide an understanding of project management and its principles in a contemporary iterative, incremental Agile project environment

Course Content

1. Traditional project management life cycles: PMI Project Management Body of Knowledge (PMBOK) and IBM RUP: initiation (scoping), planning, execution (launching), monitoring and control, and closing of projects

2. Agile project management principles. Agile philosophy. APM frameworks – envision, speculate, explore, adapt and close. Configuring project life cycles. Deliverables – management, technical. Feature-based delivery

3. Agile technical team: Roles and responsibilities, team empowerment, leadership collaboration. Agile practices: Facilitated workshops, MoSCoW approach to prioritization, iterative development methodologies – SCRUM and XP, modeling, timeboxing.

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

5. Agile control mechanisms: 7 control parameters. Transitioning to agile. Agile approach to risk and configuration management. Quality in agile project. Agile quality management. Agile testing.

REFERNCES

1. Effective Project Management: Traditional, Agile, Extreme, (7thEd): Robert K. Wysocki; Wiley India (2014)
2. Agile Project Management (2ndEd): Jim Highsmith, Addison-Wesley Professional (2009)
3. Project Management the Agile Way: Making it Work in the Enterprise (1st Ed): John C. Goodpasture, Cengage Learning India (2014)
4. Agile Project Management: Creating Innovative Products (2ndEd): Jim Highsmith, AW (2009)

CSS3104: PATTERNS IN SOFTWARE ENGINEERING

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

Course Description

Software architecture and design requires to be warned against subtle issues that can cause major problems during implementation. Often, people only understand how to apply certain software architecture and design techniques to certain problems. Formatting and applying these techniques to a broader range of problems is, by itself, a complex problem.Patterns in the areas of software architecture and design provide general solutions, documented in a format that doesn't require specifics tied to a particular problem.
In addition, patterns give a more educated vocabulary to the software architects and designers while expressing the various scenarios of software interactions.

Course Objectives

To give a comprehensive overview of recurring patterns in software development

To impart the technical details on various patterns
To provide insight into pattern based development

Course Content

1. Patterns – Patterns category – Relationships between patterns – Patterns and Software Architecture – Architectural patterns – Idioms – Pattern systems – Documentation of Patterns

2. Analysis Patterns – Patterns in analysis – business patterns – Support patterns – Patterns for typed models – Association patterns

3. Design Patterns – Catalog of design patterns –Case study implementation– Creational – Structural Patterns –Behavioural patterns

4. Patterns of Enterprise application architecture – Object relational patterns - Web presentation patterns – Distribution patterns – Concurrency patterns

5. Business Process Improvement Patterns – Pedagogical patterns – Pattern languages –Anti-patterns – Major criticisms

REFERNCES

1. Pattern-Oriented Software Architecture Volume 4& 5 :Frank Buschmann etal.,Wiley (2007)
2. Object-oriented Software Engineering using UML, Patterns & Java (3rd Edition): Bernd Bruegge, Allen H Dutoot: Oearson 2013
3. Analysis Patterns: Reusable Object Models (1st Ed): Martin Fowler, Addison-Wesley, (1997)
4. Design Patterns: Elements of Reusable Object-Oriented Software: Eric Gamma etal., Pearson (2008)
5. Patterns of Enterprise Application Architecture (1st Ed): Martin Fowler, AW (2002)
6. Enterprise Integration Patterns , Hohpe, Gregor, Bobby Woolf, Addison-Wesley(2005).
7. Remoting Patterns: Foundations of Enterprise, Internet and Realtime Distributed Object Middleware , Kircher, Michael, Markus Völter and Uwe Zdun, John Wiley & Sons(2004).
8. J2EE Design Patterns, Kaplan, Jonathan, William C. R. Crawford, O'Reilly (2003).

CSS3105: AGENT BASED COMPUTING

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

Course Description

A software agent is a piece of software that acts for a user or other program in a relationship of agency. Agents are not strictly invoked for a task, but activate themselves. This course deals with essentials of an agent based program and programming of agents.

Course Objectives

Introduce the concepts of Artificial intelligence required by agents
Study agent based programming languages and develop agent programs for various applications

Course Content

1. Artificial Intelligence â€“ intelligent agents â€“ Environment â€“ Structure of agents â€“ Agent types â€“ Problem solving agents â€“ Uninformed Search strategies â€“ Informed Search and Exploration â€“ Adversarial Search.

2. Knowledge and reasoning â€“ Knowledge Based Agents â€“ First order logic â€“ Reasoning â€“ Backward chaining â€“ Resolution â€“ Knowledge representation â€“ Handling uncertain knowledge â€“ Reasoning under uncertainty â€“ Statistical reasoning.

3. Planning â€“ Components of planning systems â€“ Planning with state space search â€“ Partial order planning â€“ Planning Graphs â€“ Hierarchical planning â€“ Multi agent planning.

4. Learning â€“ forms of learning â€“ Inductive learning â€“ Learning decision trees â€“ Explanation based learning â€“ Statistical learning â€“ Instantance based learning â€“ Neural networks â€“ Reinforcement learning.

5. Agent oriented programming language â€“ KQML as an agent communication language â€“ Java implementation of intelligent agents â€“ Languages supporting mobility â€“ Telescript.

REFERNCES

1. Industrial agents: Emerging applications of software agents in industry : Paulo leitao, Stamastis Kamouskos: Elsiever(2015)
2. Intelligent Software Agents: foundations & applications: Walter Brenner, Ruediger Zarnekow, Hartmut Wittig : Springer(2011)
3. Developing Multiagent systems with JADE: Fabio Luigi Bellifemine , Giovanni Caire: Wiley(2007)
4. Software Agents: Jeffrey M.Broadshaw, AAAI Press (1997)
5. Multi agent System â€“ A modern approach to distributed artificial intelligence: Gerhard Weiss, MIT Press (2000)
6. Artificial intelligence. A modern approach by Stuart Russell & Peter Norvig: Pearson(2009)

CSS3106: HUMAN-COMPUTER INTERCATION

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 manâ€“machine interaction (MMI) or computerâ€“human interaction (CHI).

Course Objectives

To understand basic HCI concepts and definitions
To understand the different types of interfaces
To study and design multimodal interfaces
To design & develop interfaces for diversified users

Course Content

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

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

3. Designing human computer interaction â€“ Visual design principles â€“ intercultural user interface designs â€“ Conversational speech interface â€“ multimodal interface â€“ adaptive interfaces and agents â€“ Tangible user interfaces â€“ Information visualization â€“ Human centered designs of DSS â€“ Online communities â€“ Visual environment.

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

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

REFERNCES

1. The human computer interaction hand book: fundamentals, evolving technologies and emerging applications: Andrew sears, Julie A Jacko, Lawrence Erlbaum Associates (2008)
2. Human â€“Computer Interaction (3rd Edition) : alan Dix, Janet Finlay, Gregory D Abowd, Russell Beale ;Pearson (2012)
3. Interaction Design : Beyond human Computer Interaction by Helen Sharp, Yvanno Rogers and Jenny preece, John Wiley (2011)
4. User centred design of systems :Jan Noyes, Chris Baber :Springer(2013)

CSS3107: 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 Objectives

Basic and advanced techniques for text-based information systems: efficient text indexing; Boolean and vector space retrieval models; evaluation and interface issues
Web search including crawling, link-based algorithms, and Web metadata
Understand the dynamics of the Web by building appropriate mathematical models.
Build working systems that assist users in finding useful information on the Web

Course Content

1. Taxonomy of IR Models â€“ 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 â€“indexing and searching-inverted files-suffix trees- Boolean queries-sequential searching-pattern matching

3. Text Classification, and NaÃ¯ve bayes-vector space classification-support vector machines

and machine learning on documents-flat clustering â€“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

REFERNCES

1. Modern Information Retrieval: The Concepts and Technology behind Search (2ndEd): Ricardo Baezce Yates, Berthier Ribeiro-Neto, AW (2010)
2. Introduction to Information Retrieval (1st Ed): Christopher D. Manning, Prabhakar Raghavan and Hinrich SchÃ¼tze, Cambridge University Press (2008)
3. Search Engines: Information Retrieval in Practice (1st Ed): Bruce Croft, Donald Metzler and Trevor Strohman, AW (2009)

CSS3108: 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 introduc- tion to functional programming. It covers both the theoretical underpinnings and practical, programming aspects.

Course Objectives

1. To have a theoretical understanding of functional programming.
2. To develop the ability to design and implement functional programs.

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 compre- hension – Strings and Tuples – User defined datatypes: 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 Polymor- phism – Type classes

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.

REFERNCES

[1] Richard Bird, Thinking Functionally with Haskell, Cambridge University Press, 2014.
[2] Graham Hutton, Programming in Haskell, Cambridge University Press, 2007.
[3] Kees Doets, Jan van Eijck, The Haskell Road to Logic, Maths and Programming, 2nd Edition, College Publications, 2004.
[4] Greg Michaelson, An Introduction to Functional Programming Through Lambda Calculus, Dover Publi- cations, 2011.
[5] Chris Okasaki, Purely Functional Data Structures, Cambridge University Press, 1999.

CSS3201: SOCIAL NETWORK ANALYTICS

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

Course Description

Social networks provide a powerful abstraction of the structure and dynamics of diverse kinds of people or people-to-technology interaction. Recent trends indicate the usage of social networks as a key feature for next generation usage and exploitation of the Web. This course provides an opportunity to learn social network analysis.

Course Objectives

Representation and analysis of social networks

Course Content

1. Networks of information – Mathematics of networks – Measures and metrics – Large scale strucuture of networks – Matrix algorithms and graph partitioning

2. Network models – Random graphs – walks on graphs - Communitiy discovery – Models of network formation – Small world model - Evolution in social networks

3. Processes on networks – Percolation and network resilience – Epidemics on networks – Dynamical systems on networks – Network search

4. Models for social influence analysis – Systems for expert location – Link prediction – privacy analysis – visulalization – Data and text mining in social networks - Social tagging

5. Social media - Analytics and predictive models – Information flow – Modeling and prediction of flow - Missing data - Social media datasets – patterns of information attention – linear influence model – Rich interactions

REFERNCES

1.Networks: An introduction: Mark Newman, Oxford University Press (2010)
2.Social Network Data Analytics: Charu C Aggarwal (ed.), Springer (2011)
3.Networks, Crowds, and Markets: Reasoning about a highly connected World, David Easley and Jon Kleinberg, Cambridge University Press (2010)
4.Understanding Social Networks: Theories, Concepts, and Findings: Charles Kadushin, OUP (2012)

CSS3202: MODEL-DRIVEN ARCHITECTURE

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

Course Description

Model-Driven Software Development (MDSD) is currently a highly regarded development paradigm among developers and researchers. With the advent of OMG's MDA and Microsoft's Software Factories, the MDSD approach has moved to the centre of the programmer's attention. MDSD is about using domain-specific languages to create models that express application structure or behaviour in an efficient and domain-specific way. These models are subsequently transformed into executable code by a sequence of model transformations.

Course Objectives

To give a comprehensive overview of MDSD and how it relates to industry standards such as MDA and Software Factories

To impart the technical details on meta modelling, DSL construction, model-to-model and model-to-code transformations and software architecture

To provide insight into engineering issues such as versioning, testing and product line engineering

Course Content

1. MDA concepts -Purpose of models- Modelling languages – Unified Modelling Language – UML Profiles – Object Constraint Language

2. MDSD -MDSD challenges- Architecture Centric MDSD-Software factories- Patterns and transformations

3. Meta modelling- MOF and UML-Meta modelling and OCL -Software architectures- Domain Architecture-DSL construction-Transformation architecture-Code generation techniques- MDSD tools- MDA standards

4. MDSD process-Target architecture development process-Product line engineering-Tests in model-driven application development-Testing the domain architecture- Versioning- Version management- Case study

5. Management-Decision support- Automation and reuse-Quality-Probability-Organizational aspects-Team structure-adoption strategies for MDSD-Risk analysis & Management

REFERNCES

1. Model Driven Software Engineering in Practice by Marco Brambilla,Jordi Cabot, Manual Vimmer, Morgan & Claypool Publishers (2012)
2. Model driven architecture with executable UML: Chris Raistrick, Paul Francis, John wright, Collin carter, Ian Wilkie Cambridge (2004)
3. Model-Driven Software Development: Technology, Engineering, Management - Markus Völter, Thomas Stahl, Jorn Bettin, Arno Haase, Simon Helsen, Krzysztof Czarnecki Wiley (2013)
4. Applying UML Patterns – An Introduction to Object Oriented Analysis & Design and the Unified Process – Craig Larman, Pearson Education, Asia (2002)
5. Domain Specific Languages- Martin Fowler with Rebecca Parsons Published by Addison Wesley Professional, (2010)

CSS3203: 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 faculty member before the peer students and staff. They also have to prepare a comprehensive report of the seminar presented

Course Objectives

Review and increase their understanding of the specific topics tested.
Inculcating presentation and leadership skills among students
Offering the presenter student an opportunity of interaction with peer students and staff

CSS3204: DESIGN OF REAL-TIME/EMBEDDED SOFTWARE

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

Course Description

This course describessoftware-engineering techniques to develop software for embedded systems. This course examines requirements analysis, the definition of object structure and behaviour, architectural and mechanistic design, and more detailed designs that encompass data structure, operations, and exceptions. The object-based Unified Modeling Language (UML) is used to describe the structural and behavioral aspects critical to real-time systems

Course Objectives

To understand the principles of software design for resource constrained devices
To understand real-time/embedded software modelling with UML
To understand and apply real-time design patterns

Course Content

1. Embedded / Real-Time Systems: Definitions and Issues - Object-Oriented Methods and the Unified Modeling Language – Basic concepts of Real-Time Systems - Safety critical systems – Object oriented process for embedded systems

2. Real Time Operating Systems: Case studies of QNX, VxWorks, Windows CE

3. Requirement Analysis of Real-time systems: Use cases – Heuristics for good Requirement analysis diagrams – Structural Object Analysis: Key strategies for object identification - Heuristics for good class diagrams – Behavioural Object analysis: UMLState charts – State chart heuristics

4. Architectural Design: Tasking Model, Component Model, Deployment Model, Safety / Reliability Model – Mechanistic Design – Detailed Design – Performance Analysis of Real Time Systems: Real Time Scheduling Theory

5. Dynamic Modelling - UML and Design Patterns – Real Time Design Patterns – Debugging and Testing – Real Time Frameworks – Design Automation Tools: Rhapsody OXF

REFERNCES

1. Designing Concurrent, Distributed, and Real-Time Applications with UML – H. Gomaa, AW

(2013)
2. Real Time UML: Advances in the UML for Real-Time Systems (3rd Ed), Bruce Powel Douglass,Pearson(2011)
3. Real-Time Design Patterns: Robust Scalable Architecture for Real-Time Systems, Bruce Powel Douglass, AW Object Technology Series(2002)
4. Modeling and Analysis of Real-Time and Embedded Systems with UML and MARTE:Bran Selic, Sebastian Gerard: Elsevier (2013)

CSS3205: SOFTWARE QUALITY MANAGEMENT

Core/Elective: **Elective** Semester: **2** 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 Objectives

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

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

REFERNCES

1. Software Quality: Theory and Management(3rd edition): Allan C. Gillies, Cengage (2003)

2. Software Process Quality- Management and Control: Ron S Kenett, E. R Baker, CRC (1999)
3. Metrics and Models in Software Quality Engineering:Stephen H. Kan, AW (2014)
4. Practical Reliability Engineering (5th Ed):Patric D. T.O connor, "", 5th Edition, John Wesley & sons, 2011
5. Software Engineering - A practitioners approach (7th Ed): Roger S. Pressman, McGraw Hill (2009)

CSS3206: NETWORK FORENSICS

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

Course Description

This course will introduce the student to the essential aspects of information security and network forensics. The student will be provided with the tools, techniques and industry accepted methodologies so that upon completion of the course the student will be able to describe key concepts of network security and forensics and how those concepts apply to themselves and their organization.

Course Objectives

Describe key principles, such as defense in depth and demilitarized zones (DMZ)
Provide an overview of the requirement for intrusion detection systems (IDS) and their implementation
Provide an overview of network security devices and infrastructures, including proxy servers and firewalls
Describe the methodologies used in network forensics
Describe data hiding and obfuscation and outline obfuscation methods

Course Content

1. Introduction to Security -CIA and AAA - protecting against Intruders - Users, Systems, and Data -Services, Role-Based Security, and Cloud Computing - Security and Forensic Computing - ISO 27002 - Risks -Risk Management/Avoidance - Security Policies -Defining the Policy - Example Risks - Defense-in-Depth - Gateways and DMZ (Demilitarized Zones) - Layered Model and Security - Encryption - Layered Approach to Defense

2. Intrusion Detection SystemsTypes of Intrusion - Attack Patterns - Host/Network-Based Intrusion Detection - Placement of the IDS - Snort - Example Rules -Running Snort -User, Machine, and Network Profiling – Honeypots - In-Line and Out-of-Line IDSs - False and True - Customized Agent-Based IDS

3. Network Security Elements Objectives - Introduction - Router (Packet Filtering) Firewalls - Network Address Translation - PIX/ASA Firewall - Proxy Servers

4. Network Forensics Key Protocols - Ethernet, IP, and TCP Headers - TCP Connection – ARP – SYN - Application Layer Analysis - FTP - ICMP - DNS - Port Scan - SYN Flood - Spoofed Addresses - Application Layer Analysis - HTTP - Network Logs on Hosts - Tripwire

5. Data Hiding and Obfuscation Obfuscation Using Encryption - Obfuscation through Tunneling - Covert Channels - Watermarking and Stenography - Hiding File Contents - File Contents

REFERNCES

1. Buchanan, William J. (2011). Introduction to Security and Network Forensics, CRC Press, ISBN: 978-0-8493-3568-6
2. Computer Forensics: Investigating Network Intrusions and Cyber Crime (Ec-Council Press Series:Computer Forensics), 2010
3. Michael E. Whitman, Herbert J. Mattord, "Principles of Information Security", 2nd Edition,, Cengage Learning Pub., 2012
4. Eoghan Casey, Digital Evidence and Computer Crime Forensic science, Computers and Internet', Elsevier Academic Press -Second Edition, 2011
5. Christof Paar, Jan Pelzl, Understanding Cryptography: A Textbook for Students and Practitioners, 2nd Edition, Springer's, 2010

CSS3207: ENTERPRISE APPLICATION INTEGRATION & 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 Objectives

The course will introduce the concepts and techniques related to the service-oriented as well as the data-oriented application integration approaches. Reasonable emphasis will be given for the middleware technologies and large-scale application integration standards. The course will also focus on the methods and techniques required to analyze, design, implement, automate, and evaluate business processes and workflows related to process-aware information systems.

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.

REFERNCES

1. Service Oriented Architecture: A field guide to Integrating XML and Web Services (1st Ed): T Erl, Prentice Hall
2. EnterpriseIntegration Patterns:Designing, Building and Deploying Messaging Solutions; G.Hohpe and B. Woolf, Addison Wesley Professional.
3. Next Generation Application Integration: From Simple Information to Web Services; D. Linthicum; Addison Wesley.
4. Essential Business Process Modeling; Michael Havey; O'Reilly Media

CSS3208: 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 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 Objectives

Introduce the fundamental concepts of data and data analysis
Case based study of specific data mining tasks like Clustering, Classification, Regression, Pattern Discovery and Retrieval by Content.
Introduce algorithms for temporal data mining and spatial data mining.

Course Content

1. Statistical descriptions of data-data visualization-measuring data similarity and dissimilarity-data preprocessing-data cleaning-data integration-data reduxtion-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 – patten 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

REFERNCES

1. Temporal Data mining –Theophano Mitsa, CRC Press 2010
2. Data mining concepts and techniques- Jiawei Han & Micheline Kamber, Jian Pei, Elsevier (2014)
3. Data mining methods and Techniques: A B M Showkat Ali, Saleh A Wasimi, Cengage Learning (2009)
4. Introduction to Data mining with case studies: G.K Gupta PHI (2008)