COMSATS University Islamabad

Registrar Secretariat, Academic Unit (PS)

*****

No. CUI-Reg/Notif- 4518 /21/3189 November 29, 2021

NOTIFICATION

Scheme of Studies of Master of Science in Software Engineering

Academic Council in its 32nd meeting held on October 06, 2021, on the recommendation of 29th meeting of Board of Advance Studies and Research (BASAR), approved the revised Scheme of Studies of Master of Science in Software Engineering effective from Spring 2022 at CUI System:

Nomenclature: Master of Science in Software Engineering MS (SE)

1. Minimum Duration: 02 Years
2. Minimum No. of Semesters: 04

3. Course Work:	**(Min No. of Courses)**	**(Min No. of Credit Hours)**
a) Core Courses (List Attached)	03	09
b) Domain Elective Courses (List Attached)	02	06
c) General Elective Courses (List Attached)	03	09
d) Research Methodology	01	01
e) Thesis	--	06
Total No. of Courses in the Program:	**09**	**--**
Total Credit Hours of the Program:		**31**

Note:

1. The Department can offer General Elective course from the list of elective courses of other "MS Program" being offered in the Department.
2. The Regulations relating to graduate Degree Programs approved by the Competent Authority and amended from time to time shall be applicable.

Muniba Nasir

Muniba Nasir
Additional Registrar

Encl: (Total 29 pages, including this page)

Distribution:

1. All Campus Directors, CUI
2. Dean, Faculty of Information Sciences and Technology, CUI
3. Controller of Examinations, CUI
4. Chairperson, Department of Computer Science, CUI
5. Incharge QEC/CU Online, PS
6. All HoDs/Incharge of Academics/Examinations Sections, CUI Campuses
7. Internal Distribution, Registrar Office (Academic Unit), CUI

Cc:

1. PS to Rector CUI
2 PS to Registrar CUI

Core Courses

Course Code	Course Title	Credit Hours	Pre-requisite (s)
MSE602	Advanced Requirements Engineering	3(3, 0)	-
MSE606	Advanced Software System Architecture	3(3, 0)	-
MSE512	Software Testing and Quality Assurance	3(3, 0)	-

Pre-Requisite for Thesis

Course Code	Course Title	Credit Hours	Pre-requisite (s)
CSC607	Research Methodology	1(1, 0)	-

MS Thesis

Course Code	Course Title	Credit Hours	Pre-requisite (s)
MSE800	Thesis	6(0, 6)	

Domain Elective Courses

Course Code	Course Title	Credit Hours	Pre-requisite (s)
MSE503	Software Measurement and Metrics	3(3, 0)	-
MSE504	Component Based Software Engineering	3(3, 0)	-
MSE505	Software Process Improvement	3(3, 0)	-
MSE506	Agile Software Development Methods	3(3, 0)	-
MSE507	Empirical Software Engineering	3(3, 0)	-
MSE618	Advanced Formal Methods	3(3, 0)	-
MSE621	Advanced Human-Computer Interaction	3(3, 0)	-
MSE625	Advanced Software Project Management	3(3, 0)	-

General Elective Courses

Course Code	Course Title	Credit Hours	Pre-requisite (s)
MSE510	Software Risk Management	3(3, 0)	-
MSE515	Software Configuration Management	3(3, 0)	-
MSE516	Reliability Engineering	3(3, 0)	-
MSE517	Agent Based Software Engineering	3(3, 0)	-
MSE537	Knowledge Development from Software Repositories	3(3, 0)	-
MSE542	Data Science for Software Engineering	3(3, 0)	-
MSE548	Mining and Analyzing Software Repositories	3(3, 0)	-

Munibe Nagal

MSE553	Business Analytics for Global Software Development	3(3, 0)	-
MSE661	Software Architecture for Cloud and Big Data	3(3, 0)	-
MSE563	Computational intelligence in Software Engineering	3(3, 0)	-
MSE665	Fault prediction for Software Repositories	3(3, 0)	-
MSE567	Vulnerability Analysis in Software Repositories	3(3, 0)	-
MSE568	Software Engineering for Intelligent Systems	3(3, 0)	-
MSE669	Data Analytics in Software Engineering	3(3, 0)	-
MSE670	Data Science Design Patterns	3(3, 0)	-
MSE700	Advanced Topics in Software Engineering- I *	3(3, 0)	-
MSE701	Advanced Topics in Software Engineering - II *	3(3, 0)	-

*** The contents for these courses would be subject to approval from the Head of Department.**

Munibe Nafis

Course Code: **MSE602** Pre-Requisite: **None**

Course Title: **Advanced Requirements Engineering**

Credit Hours: **3(3, 0)**

Course Objectives:

- To describe the social, cultural, political, and technical issues in requirements engineering;
- To understand activities in requirements engineering (requirements elicitation, analysis and negotiation, validation, and management);
- To understand requirements traceability and requirements documentation;
- To demonstrate use-case modeling, object-oriented static modeling, functional modeling, and dynamic modeling techniques applicable in requirements engineering;
- To review research articles from well-known SE journals and conference proceedings regarding the theories and applications of requirements engineering.

Course Contents:

Software Requirements Fundamentals: Product and process requirements, Functional and non-functional requirements, Emergent properties, Quantifiable requirements, System and software requirements. Requirements Process: Process models, Process actors, Process support and management, Process quality and improvement. Requirements Analysis: Requirements sources, Elicitation techniques. Requirements Analysis: Requirements classification, Conceptual modeling, Architectural design and requirements allocation, Requirements negotiation, Formal analysis. Requirements Specification: System definition document, System requirements document, Software requirements specification. Requirements Validation: Requirements reviews

Recommended Books:

1. Software Engineering: A Practitioner's Approach, Roger S. Pressman, Bruce R. Maxim, 8th Ed, McGraw-Hill Education, 2015.
2. 2. Object-Oriented Analysis, Design and Implementation, Brahma Dathan, Sarnath Ramnath, 2nd Ed, Universities Press, India, 2014.

Course Code: **MSE606** Pre-Requisite: **None**

Course Title: **Advanced Software System Architecture**

Credit Hours: **3(3, 0)**

Course Objectives:

- To argue the importance and role of software architecture in large-scale software systems;
- To design and motivate software architecture for large-scale software systems;
- To describe a software architecture using various documentation approaches and architectural description languages;
- To discuss and evaluate the current trends and technologies such as model-driven and service-oriented architectures;
- To evaluate the coming attractions in software architecture research and practice.

Course Contents:

Quality attributes in the context of architecting. Qualitative and quantitative assessment of architectures; Architectural modeling through Architecture Description Languages; System modeling its relation to software architecting; Architecting for evolution and variability; Partitioned and layered architectures; System-of-Systems and Ultra-Large-Scale Systems; Software Product Lines and Configurable Software; Self-Adaptive Software. Architectural Description Languages; Feature Modeling; Architecture and Model-Based Testing; Current research topics in software system architecture;

Recommended Books:

1. Designing Software Architectures: A Practical Approach (SEI Series in Software Engineering), Humberto Cervantes, Rick Kazman, 1st Edition, Addison-Wesley Professional, 2016.

Course Code: **MSE512** Pre-Requisite: **None**

Course Title: **Software Testing and Quality Assurance**

Credit Hours: **3(3, 0)**

Course Objectives:

- To describe the software testing and quality assurance (T&QA) as a fundamental component of software life cycle;
- To demonstrate T&QA activities using modern software tools;
- To estimate cost of a T&QA project and manage budgets; To prepare test plans and schedules for a T&QA project.

Course Contents:

Software Testing: Levels of Testing, Regression Testing, Requirements Tracing, Requirement Traceability Matrix, V & V Standard, Software Quality Assurance, test optimization, Eleven Step Testing Process (Assess Project Management Development Estimate and Status, Develop Test Plan, Requirements Phase Testing, Design Phase Testing, Program Phase Testing, Execute Test and Record Results, Acceptance Test, Report test results, testing software installation, Test software changes, Evaluate Test Effectiveness), Testing Security. Software Reusability, Software Metrics, Software Testing Tools, Defect Tracking Tools, Defect Management Tools, Challenges, Error-Oriented Testing and Analysis, Simulation and Prototyping, Differences from testing non-OO Software, Class testing strategies, Class Modality, Message Sequence Specification

Recommended Books:

1. Software Quality Assurance: Integrating Testing, Security, and Audit (Internal Audit and IT Audit), Abu Sayed Mahfuz, Auerbach Publications, 2016.

Domain Elective Courses

Course Code: MSE503 **Pre-Requisite:** None

Course Title: Software Measurement and Metrics

Credit Hours: 3(3, 0)

Course Objectives:

- To provide a basic understanding and knowledge of the software metrics and measurement techniques;
- To describe the concepts and importance of Metrics data collection and analysis;
- To describe external product attributes, metrics for object-oriented systems, dynamic metrics and resource measurement.

Course Contents:

Introduction to quality control and planning needs (Measurement Concepts, Measurement as a support process, Review Metrics Models and Standards). Measurement goals (Formulating problem and goal statement, prioritize information needs and objectives, Formalize measurement goals). Specify Measures (Identify questions and indicators, Identify data elements, Operational definitions for measures). Specify Data Collection and Storage Procedures. Sources of data. How to collect and store the measurement data? Specify Analysis Procedures. Potential data analyses. Methods and tools for measuring software. Develop software measurement reporting. Current research topics in Software Measurement and Metrics.

Recommended Books:

1. Metrics and Models in Software Quality Engineering, Stephen H. Kan, Addison Wesley, 2003
2. The Big Book of Six Sigma training Games, Chris Chen and Hadley Roth, McGraw-Hill, 2005

Course Code: MSE504 **Pre-Requisite:** None

Course Title: Component Based Software Engineering

Credit Hours: 3(3, 0)

Course Objectives:

- Describe technical platforms, conditions for and challenges with the development of larger, component-based software systems;
- Relate industrial and theoretical issues in the development of larger, component-based software systems to contemporary software development methods and techniques such as aspect-oriented programming and model-based development;
- Analyze and critically evaluate a technical platform for component-based software development and relate its properties to Software Engineering research.

Course Contents:

Introduction to Software Component (Component. Definition and Essentials, What is CBSE? Why CBSE? The Anatomy of Components: internals, application interfaces, platform interfaces, middleware, Component Characteristics: Properties of Software Component in CBSE). Basic Concepts in CBSE (Improving SW through Software Process Improvement (SPI)). Component-Based Software Development (CBSD). Approach. Component Patterns & Abstraction. Challenges of CBSE. Technical Issues and Objectives of Component Based Software Engineering. Reuse Dimensions. Software Components Types: open, closed, COTS, in house. Challenges in Software Reuse. Software Component Specification. Specification Techniques. Specifying the Semantics of Components. Specifying Extra-Functional Properties. Architecting component-based systems (Software Architecture Parts, The Roles of Software Architecture, Designing Software Architectures, Architectural Styles, Architecture-Driven Component Development, Components and Component Models, Component Model Implementation, Component Frameworks, Black-Box and White-Box Frameworks, How do we use Framework in CBSE, Component Interface Specification). Component Engineering Process: Domain Engineering, Domain Engineering pattern-based design. Domain Engineering: Component Repositories, Overview of Existing Component Techniques, Component testing in CBSE. Current research topics in Component Based Software Engineering.

Recommended Books:

1. Software Engineering: A Practitioner's Approach, Roger S. Pressman, 8th Edition, McGraw-Hill Higher Education, 2015
2. 2. Building Reliable Component Based Software Systems, Ivica Crnkovic and Magnus Larsson, Artech House Publishers; 1st edition, 2002
3. 3. Component-Based Development: Principles and Planning for Business Systems, Katharine Whitehead, Addison Wilsey, 2010

Course Code: MSE505 **Pre-Requisite:** None

Course Title: Software Process Improvement

Credit Hours: 3(3, 0)

Course Objectives:

- To introduce students to the basics of software process and process improvement;
- To teach students about the activities and issues software process engineering;
- To teach students different software process improvement approaches;
- To teach students concept of measurements and how it applies to software processes;
- To introduce students, advance and potential research topics in software process engineering.

Course Contents:

Introduction (Process Models, QA, CM, Project Planning), Process Modeling and Process Modeling Techniques (Introduction and ETVX), Process Modeling Techniques (IDEF0) Measuring and Analyzing the Current State, CMM and Other process models, CMMI– I, PSP and TSP, Process Changes using PDCA and IDEAL models, Process Assessments, Base-lining, and Benchmarking, Project Management aspects related to process management ,Process Measurement, Process Metrics i.e. Maturity, Management, and Life Cycle Metrics, Fundamentals of Measurement and Experimentation, GQM and its application to process management and improvement, Introduction to Quality Metrics, Software Engineering Measurements, Advance Topics in Software Process.

Recommended Books:

1. Introduction to Software Process Improvement, O'Regan, Gerard
2. Software Process Improvement and Management: Approaches and Tools for Practical Development, Shukor Sanim Mohd Fauzi, Mohd Hairul Nizam Md Nasir, 2011

Course Code: MSE506 **Pre-Requisite:** None

Course Title: Agile Software Development Methods

Credit Hours: 3(3, 0)

Course Objectives:

- To demonstrate the ability to participate effectively in agile practices/process for software development;
- To explain the purpose behind common agile practices;
- To be able to apply agile principles, methods and values;
- To identify developments in current research topics in Agile Software Development.

Course Contents:

Agile values and principles. Agile Practices. Pair programming Refactoring. Test-driven development. Continuous integration and delivery. Automated build. Coding standards simplicity. SMART user stories and release and deployment. Applying Agile methods: Integration, XP+SCRUM, SCRUM +Kanban, Agile methods +User-Centered Design. Distributed Agile teams. Current research topics in Agile Software Development.

Recommended Books:

1. Agile Software Development, Principles, Patterns, and Practices, Robert C. Martin, Pearson, 2002.
2. Extreme Programming Explained, Kent Back & Cynthia Andres, 2nd Edition, Addison-Wesley Professional 2005.
3. Learning Agile: Understanding Scrum, XP, Lean, and Kanban, Andrew Stallman and Jennifer Greene, O'Reilly Media, 2014.

Course Code: MSE507 **Pre-Requisite:** None

Course Title: Empirical Software Engineering

Credit Hours: 3(3, 0)

Course Objectives:

- To describe the Quantitative and Qualitative study designs;
- To perform archival data analysis;
- To familiarize current research techniques in empirical software engineering.

Course Contents:

Quantitative study design. Qualitative study designs. Measurement and data collection. State-of-the practice. Archival data analysis. Human variation & impact of experience. Evidence-based software engineering. Simulation of software process. Current research techniques in Empirical Software Engineering.

Recommended Books:

1. Experimentation in Software Engineering by C. Wohlin , Kluwer, 2000.

Course Code: MSE618 **Pre-Requisite:** None

Course Title: Advanced Formal Methods

Credit Hours: 3(3, 0)

Course Objectives:

- To introduce formal methods and specification;
- To describe Transformational systems and formal development cycle;
- To introduce current research topics based on Formal Methods.

Course Contents:

Introduction to formal methods and specification. State-Based Formal Methods. Transformational systems. Traditional approaches. Z specification. Formal development cycle. Temporal Specification: reactive systems, syntax and semantics of temporal logic, temporal specification of reactive systems (safety, aliveness, fairness). Model Checking: Generating finite models, Analysis of a simple model checking algorithm. Symbolic model checking. Overview of reduction methods. Spin and Promela. Case study and practical verification of properties. Current research topics based on Formal Methods.

Recommended Books:

1. Z: An Introduction to Formal Methods by Antoni Diller, 2nd Edition, John Wiley & Sons, Inc.,1994

Course Code: MSE621 **Pre-Requisite:** None

Course Title: Advanced Human-Computer Interaction

Credit Hours: 3(3, 0)

Course Objectives:

- To teach various theories of Human-Computer Interaction;
- To apply theoretical results from HCI research to software products;
- To introduce current research topics based in Human-Computer Interaction.

Course Contents:

Introduction to HCI. Importance of usable and useful software products. The theories of HCI. How to evaluate/develop software products. How to apply theoretical results from HCI research to software products. How to conduct their own research about aspects of usability and user experience. Concepts of Human Computer Interaction. The psychology of usable things. Usability Engineering. Prototypes. Usability inspection methods. Usability testing methods. Usability in practice. User Experience (UX). Web Usability. Mobile Usability. Mobile User Experience. Site objectives and user needs. Information architecture. Information and navigation design. Implementation and optimization. Experiments and HCI guidelines. Current research topics in Human-Computer Interaction.

Recommended Books:

1. About Face: The Essentials of Interaction Design, Alan Cooper, Robert Reimann, David Cronin, Christopher Noessel, Wiley, 4th Edition, 2014.
2. Designing the User Interface, Ben Shneiderman and Catherine Plaisant, Pearson, 5th Edition, 2013.
3. Research Methods in Human-Computer Interaction, Lazar, Feng, Hochheiser, Wiley, 2010.

Course Code: MSE625 **Pre-Requisite:** None

Course Title: Advanced Software Project Management

Credit Hours: 3(3, 0)

Course Objectives:

- To discuss different software project management phases;
- To prepare a project plan for a software project that includes estimates of size and effort, a schedule, resource allocation, configuration control, change management, and project risk identification and management;
- To compare different methods and techniques used to assure the quality of a software product;
- To explain an approach to risk that will help to secure the on-time delivery of software;
- To demonstrate the use of the MS-project as a tool for software project management.

Course Contents:

Introduction to project management. Algorithmic cost estimation models. Advanced cost estimation models. Function points estimation Risk assessment. Life cycle models. Prototyping. Management of software reuse. Software maintenance. Software maturity framework. An Overview of Project Planning. Program Management and Project Evaluation. Software Effort Estimation. Activity Planning. Risk Analysis and Management. Resource Allocation. Project tracking and Control. Contract Management. Software Quality Assurance. Configuration Management. Various tools of Software Project Management. Project Cost Management. Project Human Resource Management. Project Communications Management. Project Procurement Management. Case studies, Current research topics in Software Project Management.

Recommended Books:

1. Software Project Management, Bob Hughes & Mike Cotterell, 3rd Ed., McGraw-Hill Publication, 2003
2. 2. Software Project Management in Practice, Pankaj Jalote, Addison-Wesley, 2002.

Course Code: MSE510 **Pre-Requisite**: None

Course Title: Software Risk Management

Credit Hours: 3(3, 0)

Course Objectives:

- To learn SEI's Risk Management paradigm;
- To apply methods for identifying and recording risks;
- To describe the concept of Software Risk Analysis, Software Risk Planning, and Software Risk Monitoring;
- To learn risk management in CMM.

Course Contents:

What is risk and risk management? Motivation for risk management. Reasons we don't do risk management. SEI's Risk Management paradigm. Identifying and recording software risk. Risk Taxonomy. Tools and methods for identifying and recording risks. Analyzing and classifying risks. Complex project management theory. Software Risk Identification. Software Risk Analysis. Software Risk Planning. Software Risk Monitoring. Software Qualitative Risk Analysis. Quantitative Risk Analysis. Risk management and the SDLC. Risk management in CMM. Other useful tools for successful risk management. Current research topics in Software Risk Management.

Recommended Books:

1. Applied Software Risk Management: A Guide for Software Project Managers by C. Ravindranath Pandian, 2006,
2. 3. Software Risk Management by Boehm, Barry, W. IEEE Computer Society Press.

Course Code: MSE514 **Pre-Requisite:** None

Course Title: Research Methodology in Software Engineering

Credit Hours: 3(3, 0)

Course Objectives:

- To Describe principle activities, skills and ethics associated with the research process;
- To Describe different type and components of a literature review process;
- To Describe quantitative, qualitative and mixed methods approaches to research;
- To Gain a practical understanding of the various methodological tools used for scientific research;
- To prepare the key elements of a research proposal/report.

Course Contents:

Introduction to Research. Objectives of Research. Importance of Research Methodology in Research Study. Types of Research. Steps in Conducting Research. What is Literature Review? Why need for Literature Review. Types of Literature Review. Systematic Literature Review Protocol. Problem Statement and Problem formulation. Criteria for selecting a problem. Identifying Types of variables in Research. Types of hypothesis. Identifying Target Population. Types of Sampling. Sampling Techniques. Quantitative Research Methods. Scientific Methods. Design of Quantitative Surveys. Techniques to Conduct Quantitative Methods. Introduction to Qualitative Research. Qualitative Research Methods. Data Analysis and Theory in Qualitative Research Articles. Introduction to Mixed Methods Research. Design of Mixed Methods Research. Evaluation of Mixed Methods Research. Case Study. How to Conduct a Case Study. Case Study Protocol. Importance and Benefits of Case Study. Types of Statistical Tests to Conduct Data Analysis. Data Analysis Tools. Introduction to SPSS. Hands on Practice of SPSS. How to Define variables in SPSS. How to Record Collected Data in SPSS. Types of Tests via SPSS including Regression. Correlation. Cross tabulation and others. How to write Good Research Proposal. Contents of Thesis. Important Elements of Research Thesis

Recommended Books:

1. Research design: Qualitative, quantitative and mixed methods approaches, Creswell, J. W. Thousand Oaks, CA: Sage, 2014.

Course Code: MSE515 **Pre-Requisite:** None

Course Title: Software Configuration Management

Credit Hours: 3(3, 0)

Course Objectives:

- To Plan and Run an SCM Process considering specific organizational aspects in terms of people, product, project, cross-organizational, process, and tools;
- To identify and organize configuration items to be controlled with SCM, including labeling and version control;
- To perform effective surveillance of SCM activities including software configuration auditing to evaluate the conformance of software products and processes to applicable regulations, standards, guidelines, plans, and procedures;
- To manage changes during the software life cycle including deviations and waivers in software configuration control.

Course Contents:

Management of the SCM Process. Organizational Context for SCM. Constraints and Guidance for the SCM Process. Planning for SCM. SCM Plan. Surveillance of Software Configuration Management. Software Configuration Identification. Identifying Items to Be Controlled. Software Library. Software Configuration Control. Requesting

Recommended Books:

1. Software Configuration Management Patterns: Effective Teamwork, Practical Integration by Stephen P. Berczuk, Brad Appleton, 2003

Course Code: MSE516 **Pre-Requisite:** None

Course Title: Reliability Engineering

Credit Hours: 3(3, 0)

Course Objectives:

- To introduce concepts and methods in the field of reliability engineering;
- To perform reliability analysis of a system and designing the same;
- To use of TQM (Total Quality Management) tools to measure and evaluate the quality of products;
- To apply the acquired knowledge in a practical operational problems or research projects;
- To evaluate the use of reliability engineering for industrial activities.

Course Contents:

Introduction to Reliability Engineering. The Need for Reliable Software. Software Reliability Engineering Concepts. Basic Definitions. Software Reliability and System Reliability. The Dependability Concept. Reliability Modeling. Availability Modeling. Statistical Reliability Models for Software Reliability. Best Current Practices of software Reliability Engineering. Software Metrics for Reliability Assessment. Software Testing and Reliability. Software Reliability Tools. Review of Reliability Theory. Analytical Techniques and Basic Statistics for Reliability Engineering. Current research topics in Reliability Engineering.

Recommended Books:

1. An Introduction to Reliability and Maintainability Engineering, Ebeling, C. E., Waveland Press, Inc.. 2009

Course Code: MSE517 **Pre-Requisite:** None

Course Title: Agent Based Software Engineering

Credit Hours: 3(3, 0)

Course Objectives:

- To understand terminology and development process of agent-based systems;
- To learn techniques to design agent-based system;
- To know how to modify architecture of the current software systems and re-structure them to be agent-based.

Course Contents:

Overview of agent-based software engineering. Methodologies for agent-based modeling, analysis and design: Agent-based Unified Modeling Language (AUML), Agent-based analysis and design, Other agent-based analysis and design methods. Agent communication and knowledge sharing: knowledge level communication among software agents, Knowledge Interchange Format (KIF), Agent-based System Architecture and Organization. FIPA: Foundation for Intelligent Physical Agents: FIPA specification, the application, abstract architecture, agent communication, agent management and agent message transport standards and guidelines.

Recommended Books:

1. Multi-agent Systems: A Modern Approach to Distributed Artificial Intelligence, Gerhard Weiss, Edt., 1st edition, MIT Press, 2000.
2. Agent-Oriented Methodologies, Paolo Giorgini, Idea Group Publishing, 2005.
3. Agent-Oriented Software Engineering III, Fausto Giunchiglia, James J. Odell, Gerhard Weiss, Springer Verlog - LNCS 2585 – 2002.

Course Code: MSE542 **Pre-Requisite:** None

Course Title: Data Science Methods for Software Engineering

Credit Hours: 3(3, 0)

Course Objectives:

- To provide knowledge of selected data science algorithms and their applications to selected software engineering problems;
- To acquired advanced knowledge learned via published research, on selected state-of-the-art data science applications in software engineering;
- To practice programming experience (in Java or Python), prototyping data science algorithms on software engineering datasets.

Course Contents:

Data science for software engineering, Software analytics and its application in practice, seven principles of inductive software engineering, Needs of Data Analysis Patterns, Building Remedies for Data Analysis in Software Engineering Research, Predictive Test Metrics, Resource/Schedule/Content Model, Visual analytics for software engineering data.

Recommended Books:

1. Perspectives on Data Science for Software Engineering, Tim Menzies Laurie Williams Thomas Zimmermann, Morgan Kaufmann,2016
2. The Art and Science of Analyzing Software Data, Christian Bird, Tim Menzies and Thomas Zimmermann. Morgan Kaufmann, 2015.

Course Code: MSE548 **Pre-Requisite:** None

Course Title: Mining and Analyzing Software Repositories

Credit Hours: 3(3, 0)

Course Objectives:

- To equips students with a basic understanding of Mining Software Repositories (MSR);
- To deriving scientifically or practically useful information about software systems and software development;
- To provide with an overview regarding typical and current research problems and corresponding MSR techniques;
- To learn how to plan, execute, and present MSR studies in terms of the central phases of data extraction, synthesis, and analysis.

Course Contents:

Overview and motivation, Software Repositories: Source control repositories, Bug repositories, Archived communications, Deployment logs, Code repositories Selected MSR studies, Data sources for MSR, Data extraction and modeling, Techniques for mining unstructured software repositories: bug prediction, clone detection, bug triage, feature location, code search engines, traceability link recovery, evolution and trend analysis, bug localization; Basic data synthesis, Synthesis with information retrieval, Synthesis with data mining , Analysis , Link to empirical software engineering , Link to software reverse and re-engineering, Mining Software Repositories (MSR) tools, Pathways of Software Repository Research.

Recommended Books:

1. Mining Unstructured Software Repositories, Stephen W. Thomas, Ahmed E. Hassan, Dorothea Blostein, 2018.

Course Code: MSE553 **Pre-Requisite:** None

Course Title: Business Analytics for Global software development

Credit Hours: 3(3, 0)

Course Objectives:

- To enable students to recognize, understand and apply the language, theory and models of the field of business analytics for developing global software;
- To plan and carry out a development project with participants from different cultures and with different technical competences;
- To describe theories about the trade-offs associated with central concepts and algorithms for distributed systems;
- To take a critical approach to solving open problems and making design decisions during the development process;
- To constructively deal with issues that arise in heterogeneous collaborative projects, such as personality conflicts and cultural differences;
- To plan a project in such a way that it can be done within given restrictions and using relevant collaboration tools.

Course Contents:

The topics include: Business Intelligence (BI) techniques to software development environments, BI framework tailored to software development, planning and carrying through a project, collaboration and group dynamic aspects of international team work. Concepts in distributed systems such as remote objects, clocks and clock synchronization, global states, replication, atomic transactions and concurrency control, threads and methods for locking as well as accessing shared resources, network programming, programming of embedded systems, real case studies.

Readings Material:

The course may be covered from reputed journal of the field

Course Code: MSE661 **Pre-Requisite:** None

Course Title: Software Architecture for Cloud and Big Data

Credit Hours: 3(3, 0)

Course Objectives:

- To discusses systematic and disciplined approaches to building software architectures for cloud and big data with state-of-the-art methods and techniques;
- To presents case studies involving enterprise, business, and government service deployment of big data applications;
- To shares guidance on theory, frameworks, methodologies, and architecture for cloud and big data.

Course Contents:

Cloud and Big Data Requirements Engineering, Cloud and Big Data Software Architectures and Styles, Architecture modelling and description languages for Cloud and Big Data, Evaluation, Analysis and Verification for Cloud and Big Data Architectures, Architecture-centric evolution for cloud and Big Data, Architecting for Data and Cloud Service, Self-adaptive and managed architectures for cloud and Big Data, Economics-Driven architecting for Cloud in the presence of Big Data, Architecture-level testing for Cloud and Big Data Services, Ethical and legal issues in architecting for cloud and Big Data, Data-Driven Applications on the cloud, Cloud and Big Data Simulation Tools, Architectures for Data Analytic Applications on the Cloud, Intelligent Services benefiting from Cloud and Big Data

Recommended Books:

1. Software Architecture for Big Data and the Cloud, Ivan Mistrik, Rami Bahsoon, Nour Ali, Maritta Heisel, Bruce Maxim, Morgan Kaufmann 2017.

Course Code: MSE563 **Pre-Requisite:** None

Course Title: Computational Intelligence in Software Engineering

Credit Hours: 3(3, 0)

Course Objectives:

- To formulate software engineering problems as optimization or machine learning problems;
- To demonstrate an understanding of the core techniques used in the computational intelligence approaches to solve software engineering problems;
- To build models able to support software engineering learning tasks; To use optimization algorithms to support software engineering optimization problems;
- To evaluate, analyze and critique computational intelligence approaches for software engineering; Students will learn how to: research current issues in computational intelligence for software engineering; write reports;
- To solve problems; and use computational intelligence toolboxes.

Course Contents:

Software Engineering with Computational Intelligence: Fuzzy Concepts and Formal Methods.- Trade-off Requirement Engineering.- A Generalized Object-Oriented Data Model Based on Level-2 Fuzzy Sets.- Modelling Imperfect Spatial Information in a Fuzzy Object Oriented Database.- Using Classical Object-Oriented Features to Build a Fuzzy O-O Database System.- Domain Analysis for the Engineering of Spatiotemporal Software.- Object-Oriented Framework of Fuzzy Knowledge Systems.- Fuzzy Evaluation of Domain Knowledge.- Application of Fuzzy Rule Extraction to Minimize the Costs of Misclassification on Software Quality Modeling.- Processing Software Engineering Data: Granular-based Approach, computational intelligence toolboxes, Pathways of Computational Intelligence in Software Engineering Research.

Recommended Books:

1. Software Engineering with Computational Intelligence, Jonathan Lee

Course Code: MSE665 **Pre-Requisite:** None

Course Title: Fault prediction for software repositories

Credit Hours: 3(3, 0)

Course Objectives:

- To understand the importance of software defect prediction techniques;
- To demonstrate how to combine data from the version control system in Eclipse software development repository;
- To predict software defects classification on file and packages levels individually and evaluate the performance of different models.

Course Contents:

Relationship between software defects and the line of code, Software metrics used in defects prediction: Software metrics for file-based analysis, Software metrics for packages based analysis, Eclipse defects data in version 2.0.2.1,3.0; Prediction model and model evaluation, Regression technique for predicting the quantity and density of software defects, classification model for measuring higher defect risks in software module. Latest Research Issues from Top Notch Journals from the related areas.

Readings Material:

The course may be covered from reputed journal of the field

Course Code: MSE567 **Pre-Requisite:** None

Course Title: Vulnerability Analysis in Mining Software Repositories

Credit Hours: 3(3, 0)

Course Objectives:

- To describe classifications and taxonomies for software vulnerabilities;
- To discuss Vulnerabilities Detection and analysis techniques in Mining Software Repositories;
- To introduce students, advance and potential research topics in this field.

Course Contents:

Classifications and Taxonomies for software vulnerabilities, Security Vulnerability Databases (SVDBs), Use of Vulnerability Databases in Software Engineering, Vulnerabilities Detection Techniques, Security Vulnerability Analysis Framework, Latest Research Issues from Top Notch Journals from the related areas

Readings Material:

The course may be covered from reputed journal of the field

Course Code: MSE669 **Pre-Requisite:** None

Course Title: Data Analytics in Software Engineering

Credit Hours: 3(3, 0)

Course Objectives:

- To equip students with data preparation tools necessary for Analytics;
- To familiarize integrated techniques necessary for formulating data science problem;
- To train essential skills in data science through demonstrations of how to use data to construct models, predict outcomes, and visualize data.

Course Contents:

Topics includes: Introduction to Data Preparation Tools and its various Types, Data Preparation in Analytics Architecture, Data Preparation and Analytics Life Cycle, Data Discovery, Data Exploration and Data Sourcing, Data Transformation and Data Blending, Data Governance, The Technology Landscape, Product Overviews and Selected Demonstrations, Selected conference and journal papers in the relevant field and topics.

Recommended Books:

1. Python Data Science Handbook: Tools and Techniques for Developers. Jake Vanderplas, O'Reilly. 2016.
2. Learning Spark: Lightning-Fast Big Data Analysis, Holden Karau , Andy Konwinski , Patrick Wendell , Matei Zaharia, O'Reilly. 2016
3. Data Analysis with Open Source Tools: A Hands-On Guide for Programmers and Data Scientists, Philipp K. Janert. O'Reilly.

Course Code: MSE670 **Pre-Requisite:** None

Course Title: Data Science Design Patterns

Credit Hours: 3(3, 0)

Course Objectives:

- To demonstrate how to use design patterns to address user interface design issues;
- To identify the most suitable design pattern to address a given application design problem;
- To apply design principles (e.g., open-closed, dependency inversion, least knowledge);
- To critique code by identifying and refactoring anti-patterns.

Course Contents:

Design Pattern Principles; Rational Behind Design Patterns; Design Patterns required for MapReduce, Filtering Patterns, Five different types of Organization Patterns: Structured to Hierarchical Pattern, Partitioning Pattern, Binning Pattern, Total Order Sorting Pattern and Shuffling Pattern, Four different types of Join Patterns: Reduce Side Join Pattern, Replicated Join Pattern, Composite Join Pattern, Cartesian Product Join Pattern, Types of Meta Patterns: Job Chaining ? Description, use cases, chaining with driver, basic & parallel job chaining, chaining with shell scripts, chaining with job control, Types of Input Output Patterns, Customizing Input & Output, Design Patterns for Deep Learning Architectures

Recommended Books:

1. Data Science Design Patterns, Todd Morley, Addison Wesley

Course Code: MSE568 **Pre-Requisite:** None

Course Title: Software Engineering for Intelligent Systems

Credit Hours: 3(3, 0)

Course Objectives:

- To describe various components and challenges of an AI-Enabled System;
- To discuss Software Architecture, Model Quality Test and Deployment Automation, and Security of AI-enabled Systems;
- To discuss the tradeoffs among AI Techniques.

Course Contents:

Introduction and Motivation, Components of an AI-Enabled System, Software Engineering Bootcamp, Engineering Challenges for AI-Enabled Systems, Requirements and Dealing with Mistakes, Tradeoffs among AI Techniques, Software Architecture of AI-enabled Systems, Model Quality, Test and Deployment Automation, Security, Adversarial Learning, and Feedback Loops, Scalability and Distributed Systems, Managing and Processing Large Datasets, Symbolic AI in Systems

Readings Material:

The course may be covered from reputed journal of the field