# Syllabus
# for
# M. Tech. (Software Engineering)
# w.e.f. July 2017

## Master of Technology (Software Engineering)

| Sr No | Code | Course Name | Teaching Scheme | | | | Examination Scheme | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | L | P | T | CR | IA | MSE | ESE | OR | Total |
| **Semester I** | | | | | | | | | | | |
| 1 | MTSE1101 | Computer Algorithms | 3 | | 1 | 4 | 20 | 20 | 60 | | 100 |
| 2 | MTSE1102 | Machine Learning | 3 | | 1 | 4 | 20 | 20 | 60 | | 100 |
| 3 | MTSE1103 | Requirements Engineering | 3 | | 1 | 4 | 20 | 20 | 60 | | 100 |
| 4 | MTSE1104 | Elective I | 3 | | | 3 | 20 | 20 | 60 | | 100 |
| 5 | MTSE1105 | Elective II | 3 | | | 3 | 20 | 20 | 60 | | 100 |
| 6 | MTSE1106 | Communication Skill | 2 | | | 2 | 25 | | | 25 | 50 |
| 7 | MTSE1107 | Software Lab I | | 4 | | 2 | 25 | | | 25 | 50 |
| | | Total | 17 | 4 | 3 | 22 | 150 | 100 | 300 | 50 | 600 |
| **Semester II** | | | | | | | | | | | |
| 1 | MTSE1201 | Data Science | 3 | | 1 | 4 | 20 | 20 | 60 | | 100 |
| 2 | MTSE1202 | Software Architecture | 3 | | 1 | 4 | 20 | 20 | 60 | | 100 |
| 3 | MTSE1203 | Elective III | 3 | | | 3 | 20 | 20 | 60 | | 100 |
| 4 | MTSE1204 | Elective IV | 3 | | | 3 | 20 | 20 | 60 | | 100 |
| 5 | MTSE1205 | Elective V | 3 | | | 3 | 20 | 20 | 60 | | 100 |
| 7 | MTSE1207 | Software Lab II | | 4 | | 2 | 50 | | | 50 | 100 |
| 8 | MT CE1208 | Seminar I | | 4 | | 2 | 50 | | | 50 | 100 |
| | | Total | 15 | 8 | 2 | 21 | 200 | 100 | 300 | 100 | 700 |
| **Semester III** | | | | | | | | | | | |
| 1 | MTSE2101 | Project Management and Intellectual Property Rights (Self Study) | | | | 2 | 50 | | | 50 | 100 |
| 3 | MTSE2103 | Project- I | | | | 10 | 50 | | | 50 | 100 |
| | | Total | | | | 12 | 100 | | | 100 | 200 |
| **Semester IV** | | | | | | | | | | | |
| 1 | MTSE2201 | Project-II | | | | 20 | 100 | | | 100 | 200 |
| | | Total | | | | 20 | 100 | | | 100 | 200 |

## List of Electives

**Elective 1**

1. Software Modeling and Reasoning
2. Software Performance Engineering

**Elective 2**

1. Program Analysis
2. Verification of Reactive Systems
3. Mobile Computing

**Elective 3**

1. Software Testing
2. Software Language Engineering

**Elective 4**

1. Object-Oriented Systems
2. Human Computer Interaction

**Elective 5:**

**1.** Functional Programming

**2.** Empirical Software Engineering

## MTSE1101: Computer Algorithms

L:3 T:1 P:0 MSE:20 CA:20 ESE:60

**Prerequisites:** Data-structures.

### Course Contents

**Advanced Data Structures:** Red-Black Trees, B-Trees, Binomial Heap, Fibonacci Heap Data Structures for Disjoint Sets.

**Graph algorithm**: Search algorithms, computation of strongly connected components, shortest distance algorithms, minimum spanning tree algorithms.

**Network-flow algorithm**: Ford-Fulkerson method; preflow-push algorithm

**Geometric algorithm**: convex-hull computation, line-segment intersection computation, closest-pair computation.

**String matching**: Rabin Karp algorithm, Knuth-Morris-Pratt algorithm, Boyer-Moore algorithm

**Matrix algorithms**: Strassen’s multiplication algorithm, LU decomposition, inverse computation

**Polynomial computation algorithms**: multiplication using DFT, division

**Number theoretic algorithms**: division, solution of modular linear equation, primality testing.

**REFERENCES:**

1. Cormen, Leiserson, Rivest, "Introduction to Algorithms", McGraw Hill.
2. Aho, Hopcroft, Ullman, " The Design and Analysis of Computer Algorithms", Addison Wesley.

**NPTEL Course**

1. Computer Algorithms – 2 by Prof. Shashank K. Mehta, IIT Kanpur.

## MTSE1102: Machine Learning

L:3 T:1 P:0 MSE:20 IA:20 ESE:60

**Prerequisites:** Basic programming skills (in Python), algorithm design, basics of probability & statistics

### Course Contents

**Introduction:** Basic definitions, types of learning, hypothesis space and inductive bias, evaluation, cross-validation.

Linear regression, Decision trees, overfitting.

Instance based learning, Feature reduction, Feature Selection, Collaborative filtering based recommendation.

Probability and Bayes learning, Evaluation Measures, Hypothesis Testing.

Logistic Regression, Linear Classification, Support Vector Machine, Kernel function and Kernel SVM.

**Neural network:** Perceptron, multilayer network, backpropagation, introduction to deep neural network.

Computational learning theory, PAC learning model, Sample complexity, VC Dimension, Ensemble learning ad methods.

**Clustering:** k-means, adaptive hierarchical clustering, Gaussian mixture model.

Expectation Maximization,  Introduction to Reinforcement Learning.

**REFERENCES:**

1. T. Hastie, R. Tibshirani, J. Friedman. The Elements of Statistical Learning, 2e, 2008.
2. Christopher Bishop. Pattern Recognition and Machine Learning. 2e.
3. Machine Learning. Tom Mitchell. First Edition, McGraw- Hill, 1997.
4. Introduction to Machine Learning Edition 2, by Ethem Alpaydin.
5. Darren Cook Practical Machine Learning with H2O Oreilly 2017

**NPTEL Courses:**

1. Introduction to Machine Learning by Dr. Balaraman Ravindran, IIT Madras.
2. Introduction to Machine Learning by Prof. S. Sarkar, IIT Kharagpur.

## MTSE1103 Requirements Engineering (Elective I)

- **Introduction:** Introduction to Requirements, Introduction to Systems Engineering , Defining Requirements Engineering .Requirements and Quality. Requirements and the Lifecycle Requirements Tracing,  Requirements and Modelling. Requirements and Testing Requirements in the Problem
- **A Generic Process for Requirements Engineering**..Introduction Developing System , Generic Process Context  Generic Process Introduction, Generic Process Information Model #Generic Process Details.
- **System Modelling for Requirements Engineering**. Introduction. Representations for Requirements Engineering, Methods..

- **Writing and Reviewing Requirements** Introduction Requirements for Requirements , Structuring Requirements Documents Key Requirements Using Attributes. Ensuring Consistency Across Requirements Value of a Requirement. The Language of Requirements Requirement Boilerplates. Granularity of Requirements. Criteria for Writing Requirements Statements
- **Requirements Engineering in the Problem Domain**What is the Problem Domain? Instantiating the Generic Process.Agree Requirements with CustomerAnalyse & Mode Identify Stakeholders. Create Use Scenarios Scoping the System.Derive Requirements.
- **Requirements Engineering in the Solution Domain** What is the Solution Domain Engineering Requirements from Stakeholder Requirements to System Requirements. Engineering Requirements from SystemRequirements to Subsystems. Design Model..
- **Advanced Traceability** Introduction. Elementary Traceability. Satisfaction Arguments Requirements Allocation. Reviewing Traceability. The Language of Satisfaction Arguments.. Rich Traceability Analysis.. Rich Traceability for Qualification. Implementing Rich Traceability Design Documents. Metrics for Traceability.
- **Management Aspects of Requirements Engineering** Introduction to Management Requirements Management Problems. Managing Requirements in an Acquisition Organisation Supplier Organisations. Product Organisations.
- **DOORS: A Tool to Manage Requirements**.. Introduction The Case for Requirements Management. DOORS Architecture. Projects, Modules and Objects. 9.5 #History and Version Control Attributes and Views. Traceabilit Import and Export UML Modelling with DOORS/Analyst.

**Reference Books**

1. Elizabeth Hull Ken JacksonJeremy Dick Requirements Engineering Springer

## MTSE1103 Software Modeling and Reasoning (Elective 1)

- **Propositional Logic**: Declarative Sentences, Natural Deduction, Propositional Logic as a formal Language, Semantics of Propositional Logic, Normal Forms SAT Solvers
- **Predicate Logic:** Predicate Logic as a formal Language, Proof theory of Predicate Logic

Semantics of Predicate Logic, Undecidability of Predicate Logic, Expressiveness of Predicate Logic, Micromodels of Softwares

- **Verification by Model-checking:** Linear Time Temporal Logic (LTL), Model-Checking Systems, Tool and Properties, Branching Time Logic, CTL*, Model-Checking Algorithms, The Fixed Point Characterization of CTL.
- **Program Verifcation:** A framework for Software Verification, Proof Calculus for Partial and Total Correctness. Programming by Contract
- **Modal Logics and Agents:** Modes of Truths, Basic Modal Logic, Logic Engineering, Natural Deduction, Reasoning about knowledge in a multi agent system.
- Binary Decision Diagrams: Representing Boolean Functions, Algorithms for reduced OBD, Algorithms for reduced OBDDs, Symbolic Model-checking, Relational mu-calculus

**Reference Books**

1. Michael Huth, Logics in Computer Science: Modeling and Reasoning about Systems, Prentice Hall,

## MTSE1103 Software Performance Engineering (Elective I)

- **Overview:** The Role of Performance Requirements in Performance Engineering , Business and Process Aspects of Performance Engineering, Disciplines and Techniques Used in Performance Engineering Performance Modeling, Measurement, and Testing Roles and Activities of a Performance Engineer, Interactions and Dependencies between Performance Engineering and other activities
- **Performance Metrics** General, Examples of Performance Metrics, Useful Properties of Performance Metrics, Performance Metrics in Different Domains,
- **Basic Performance Analysis** How Performance Models Inform Us about Systems Queues in Computer Systems and in Daily Life Causes of Queueing Characterizing the Performance of a Queue Basic Performance Laws: Utilization Law, Little’s Law Utilization Law Little’s Law A Single-Server Queue Networks of Queues: Introduction and Eementary Performance Properties Quantifying Device Loadings and Flow through a Computer System, 3 Upper Bounds on System Throughput, Lower Bounds on System

Response Times, Open and Closed Queueing Network Models, Simple Single-Class Open Queueing , Network Models Simple Single-Class Closed Queueing Network Mode Bottleneck Analysis for Single-Class Regularity Conditions for Computationally Tractable Queueing Network Models Multiple-Class Queueing Networks Applications of Basic Performance Laws.

- **Workload Identification and Characterization W**orkload Identification, Reference Workloads for a System in Different Environments, Time-Varying Behavior Mapping Application Domains to Computer System , Numerical Specification of the Workloads Numerical Data for an Online Securities
- **From Workloads to Business Aspects of Performance Requirements**Performance Requirements and Product Management, Sizing for Different Market Segments: Performance Requirements and the Software Lifecycle,Performance Requirements and the Mitigation of Business Risk Commercial Considerations and Performance Requirements,Performance Requirements and the Outsourcing of Software Development Guidelines for Specifying Performance Requirements Performance Requirements and Functional Requirements#
- **Qualitative and Quantitative Types of Performance Requirement** Qualitative Attributes Related to System Performance, The Concept of Sustainable Load, Formulation of Response Time Requirements Formulation of Throughput Requirements, Derived and Implicit Performance Requirements, Derived Performance Requirements, implicit Requirements Performance Requirements Related to Transaction Failure Rates, Lost Calls, and Lost Packets Performance Requirements Concerning Peak and Transient Loads
- Eliciting, Writing, and Managing Performance Requirements Elicitation and Gathering of Performance Requirements, Common Patterns and Antipatterns for Performance Requirements Response Time Pattern and Antipattern Resource Utilization Antipattern Number of Users to Be Supported Pattern Pool Size Requirement Pattern, Scalability Antipattern , The Need for Mathematically Consistent Expressing Performance Requirements in
- **System Measurement Techniques and Instrumentation**Distinguishing between Measurement and Testing Validate, Validate, Validate; Scrutinize, Scrutinize, Scrutinize, Resource Usage Measurements, Measuring Processor Usage, Processor Utilization by Individual Processes, Disk Utilization , Bandwidth Utilization Queue Lengths Utilizations and the Averaging Time Window

**Reference Books**

1. Andrei Bondi Foundations of Software and System Performance Engineering Wiley Publication

## MTSE1105 Program Analysis (Elective 2)

1. **Introduction:** Data Flaow Analysis, Constraint Based Analysis, Abstract Interpretation, Type and Effect Systems.
2. **Data-Flow Analysis:** Intraprocedural Analysis, Theoretical Properties, Monotone Frameworks, Equation Solving, Interprocedural Analysis, Shape Analysis
3. Constraint Based Analysis: Abstract0-CFA Analysis, Theoretical Properties, Syntax-Directed 0-CFA Analysis, Constraint Based 0-CFA Analysis, Adding Data Flow Analysis, Adding Context Information
4. Abstract Interpretation: A Mundane Approach to Correctness, Approximation of Fixed Points, Galois-Connections, Systematic Design of Galois Connections, Induced Operations
5. Type and Effect Systems: Control Flow Analysis, Theoretical Properties, Inference Algorithm, Effects, Behaviours
6. Algorithms: Worklist Algorithms, Iterating in Reverse Postorder, Iterating

Through Strong Components

**Reference Books**

1. Flemming Nielson, Hanne Riis Nielson, Principles of Program Analysis, Springer Link
2.

## MTSE1105 Verification of Reactive Systems (Elective 2)

- Introduction What are Reactive Systems
- The Language of CCS
- Behavioural equivalences
- Theory of fixed points and bisimulation equivalence
- Hennessy-Milner logic
- Hennessy-Milner logic with recursive definitions
- Modelling and analysis of mutual exclusion algorithms
- CCS with time delays

- Timed automata
- Timed behavioural equivalences
- Hennessy-Milner logic with time
- Modelling and analysis of Fischer’s algorithm

**Reference Books**

1. Luca Aceto et al, Reactive Systems: Modelling, Specification and Verification, Springer

## MTSE1105 Mobile Computing (Elective 2)

L:3 T:0 P:0 MSE:20 IA:20 ESE:60

**Prerequisites:** Java Programming, Operating Systems, Basic knowledge on socket connection.

### Course Contents

Introduction to mobile computing, installing of required software and preparing the working environment, creatingyour first Android Application.

Layouts, Views, Resources.

Activities, Intents.

Background tasks, Connecting to the Internet.

Fragments, Preferences.

User Interaction – input, menu items, custom views.

User Experience – themes and styles, material design, adaptive layouts, accessibility, localization, debuggingthe UI.

Storing Data, SQLite database.

Sharing Data, content resolvers and providers, loaders to load data.

Services, background work, alarms, broadcast receivers.

Notification, widgets, transferring data efficiently, publishing app.

Multiple form factors, sensors, Google cloud messaging, monetizing your app.

**REFERENCE:**

1. Android Programming (Big Nerd Ranch Guide), by Phillips, Stewart, Hardy and Marsicano.
2. Android Programming – Pushing the limits by Hellman.

**NPTEL Course:**

1. Mobile Computing by Prof. Pushpendra Singh, IIITD.

# Semester II

## MTSE1201: Data Science

L:3 T:1 P:0 MSE:20 IA:20 ESE:60

Prerequisites:

**Course Contents**

Data Mining Patterns: Cluster Analysis, Anomaly Detection, Association Rules,

Data Mining Sequences:

Text Mining: Text mining Text Clusters

Data Analysis: Simple regression, Multiple Regression, Multivariate Regression Analysis, Robust Regression, Correlation, Clustering.

Data Viualization: R graphics, Plotting, Scatter Plots Bar Charts and Plots 3D graphics

Machine Learning: Data Partitioning Predicting events with machine learning, Supervised and Unsupervised learning.

Reference Books

1. Dan Toomey, R for Data Science, Packit First Edition Publishing 2014 NPTEL/Open Course
2. Hadley Wickham et al R for Data Science Oreilly 2016
3. Richard Cotton Learning R Oreilly 2013

## MTSE1202: Software Architecture

L:3 T:1 P:0 MSE:20 IA:20 ESE:60

Prerequisites:

Course Contents

Review of Software Engineering, Various Definitions of Software Architecture, Architecture Documentation: SEI Framework, Module View, Component and Connector View, Deployment View, Pattern-Oriented Software Architecture: Layer, MVC, Pipe-Filter, Publish/Scriber, Presentation Abstraction and Control Patterns, Software Architecture quality Attributes, Evaluating Software Architecture, Architecture Decisions, Architecture Knowledge Management, Technology Architectures.

Reference Books

1. Paul Clements, Documenting Software Architecture, Addison Wesley
2. Fran Buschman Pattern Oriented Software Architecture Vol I

## MTSE1203 Software Testing (Elective 3)

L:3 T:1 P:0 MSE:20 IA:20 ESE:60

### Course Contents

Introduction: Principles of testing, Software development life cycle models.

Types of testing: White box testing - Static testing, Structural testing, Black box testing–Requirement based testing, positive and negative testing, boundary value analysis, decision tables, equivalence partitioning, state based or graph based testing, compatibility testing, user documentation testing, domain testing.

Integration testing: top down integration, bottom up integration, bi-directional integration, system integration System and Acceptance testing–functional testing–design/architecture verification, business vertical testing, deployment testing, beta testing, certification standards and testing for compliance;

Non-functional testing: setting up the configuration, coming up with entry/exit criteria, balancing key resources, scalability testing, reliability testing, stress testing, interoperability testing;

Acceptance testing: acceptance criteria, selecting test cases for acceptance testing, executing acceptance tests.

Performance testing: collecting requirement, writing test cases, automating performance test cases, analyzing the performance test results, performance benchmarking, capacity planning.

Regression testing: erforming an initial smoke or sanity test, understanding criteria for selecting the test cases, classifying test cases, methodology for selecting test cases, resetting the test cases for regression testing Test planning, management, execution and reporting.

Test metrics and measurements.

REFERENCE:

1. Srinivasan Desikan, Gopalaswamy Ramesh, “Software Tesing Principles and Practices”, Pearson Education.
2. William Perry, “Effective Methods for Software Testing”, John Wiley & Sons, New York, 1995.

## MTSE1203 Software Language Engineering (Elective 3)

- Introduction
- Grammars and parsing
- Language processing
- Attribute grammars
- Rewriting & strategies
- Automated refactoring
- Domain-specific languages
- Domain-specific language design
- Grammar-based testing
- Linguistic architecture
- Code generation

**Reference Books**

Ralf Lammel, A Course on Software Language Engineering

## MTSE104 Functional Programming (Elective 4)

L:3 T:0 P:0 MSE:20 IA:20 ESE:60

**Prerequisites:**

**Course Contents**

Introduction to Haskell and the ghci interpreter

Defining functions: guards, pattern matching and recursion

Lists, strings and tuples 4. Types and polymorphim

Higher order functions on lists: map, filter, list comprehension

Computation as rewriting, lazy evaluation and infinite data structures

Conditional polymorphism and type classes

User defined datatypes: lists, queues, trees

Input/output and the ghc compiler

Arrays

Reference Books

## MTSE12 Empirical Software Engineering (Elective 4)

- Introduction
- Systematic Literature Reviews
- Software Metrics
- Experimental Design
- Mining Data from Software Repositories
- Data Analysis and Statistical Testing
- Model Development and Interpretation
- Validity Threats
- Reporting Results
- Mining Unstructured Data
- Demonstrating Empirical Procedures
- Tools for Analyzing Data-diction

**Reference Books**

1Ruchika Malhotra, Empirical Research in Software Engineering: Concepts, Analysis, and Applications, CRC PRess

## MTSE1205 Object-Oriented System (Elective 5)

L:3 T:0 P:0 MSE:20 IA:20 ESE:60

**Prerequisites:**

**Course Contents**

Review of programming practices and code-reuse; Object model and object-oriented concepts; Object-oriented programming languages and implementation; Object-oriented analyses and design using UML structural, behavioral and architectural modeling; Unified development process, Software reuse design patterns, components and framework; Distributed object computing, interoperability and middle ware standards COM/DCOM and CORBA; Object-oriented database system data model, object definition and query language, object-relational system.

**REFERENCE:**

1. Object Oriented System Analysis, Sally Shlaer, Prentice Hall PTR.
2. Object Oriented System Analysis and Design using UML, Simon Bennett, McGraw-Hill.

## MTSE1205 Human Computer Interface (Elective 5)

**Introduction:** Course objective and overview, Historical evolution of the field, The Human, The Computer, The Interaction.

**Design processes**: Interaction Design basics, Concept of usability – definition and elaboration, HCI in the Software Process, Design Rules.

**Implementation and Evaluation:** Implementation Support, Evaluation Techniques, Universal Design, Use Support.

**Models:** Cognitive Models, Socio – Organizational Issues and Stakeholders Requirements, Communication and Collaboration models.

**Theories:** Task Analysis Dialog notations and Design Models of the system Modeling Rich Interactions.

**Modern Systems:** Group ware, Ubiquitous Computing computing and Augmented Realities Hypertext Multimedia and World Wide web.

**Text Books:**

1. Alan Dix, Janet Finlay, Gregory Abowd, Russell Beale, *Human Computer Interaction*, 3rd Edition Pearson Education.
2. Preece J., Rogers Y., Sharp H., Baniyon D., Holland S. and Carey T., *Human Computer Interaction*, Addison-Wesley, 1994.
3. B. Shneiderman, *Designing the User Interface*, Addison Wesley 2000 (Indian Reprint).

**Reference Books:**

1. Jenny Preece, Helen Sharp, Yvonne Rogers, *Interaction Design: Beyond Human-Computer Interaction*, 4th Edition, Wiley Publication.
2. Gerard Jounghyun Kim, *Human–Computer Interaction: Fundamentals and Practice*, CRC Press.
3. Jenifer Tidwell, *Designing Interfaces 2nd Edition, Patterns for Effective Interaction Design*, O'Reilly Media