IV Semester MSc(Comp. Sc.) new syllabus
(w.e.f. 2017-18 and onwards)

Course Code	Subject Title	Teaching Scheme Hrs/week		Examination				Credits
				Exam. Duration (Hrs)	Marks			
		Theory	Practical		Theory/ Practical	IA	Total	
17MScCSCS 4.1	Optimization Techniques	4	--	3	80	20	100	4
17MScCSCS 4.2	Data Mining	4	--	3	80	20	100	4
17MScCSCS 4.3	Digital Image Processing	4	--	3	80	20	100	4
17MScCSCS 4.4	Computer Graphics	4	--	3	80	20	100	4
17MScCSCE 4.5	a. Software Testing b. Cloud Computing c. Theory of Computation d. Object Oriented Analysis and Design Using UML	4	--	3	80	20	100	4
17MScCSPL 4.6	Prog. Lab.: Computer Graphics Lab.		6	3	80	20	100	3
17MScCSPL 4.7	Prog. Lab.: Project Work	--	6	3	100	20	100	3
Total		**20**	**12**				**700**	**26**

CS: Core Course/Subject CE: Core Elective Course OE: Open Elective

17MScCSCS 4.1	**Optimization Techniques**	
Credits: 4	Teaching: 4Hrs/week	Max. Marks: 80 IA: 20

UNIT I 12Hrs

Introduction: Operations research model, solving the OR model, art of modeling, phases of OR study. Linear Programming: Formulation of a LP model, graphical solution to LPP, LP applications, standard and canonical form of LPP, the simplex method, big M-method, two-phase simplex method, special cases in simplex method, sensitivity analysis.

UNIT II 10Hrs

Duality and Post-Optimal analysis: Definition of dual problem, primal-dual relationships, economic interpretation of duality, dual simplex method, generalized simplex algorithm, post-optimal analysis.

UNIT III 10Hrs

Transportation models: Definition of transportation model, the transportation algorithm, the assignment model-the Hungarian method; the transshipment model.

UNIT IV 10Hrs

Network Models: Scope and definition of Network models, minimal spanning tree algorithm, shortest-route problem, maximal flow model, CPM and PERT.

UNIT V 10Hrs

Advanced Linear Programming: Simplex method fundamentals, revised simplex method, bounded-variable algorithm, duality, parametric linear programming.
Decision Analysis and Game theory: Decision making under certainty, decision making under risk, decision under uncertainty, optimal solution of two-person zero-sum games, solution of mixed strategy games.

References:

1. Hamdy A. Taha, Operations Research, 8/e, Pearson Education.
2. By Filet B. E,.Introduction to Operation Research : A Computer Oriented Algorithm Approach
3. Gillet B.E, Introduction to Operations Research, TMH.
4. Sharma J.K, Operations Research, Theory and Applications, McMillan India Ltd.

17MScCSCE 4.2	**Data Mining**	
Credits: 4	Teaching: 4Hrs/week	Max. Marks: 80 IA: 20

UNIT I 08Hrs

Introduction to Data Mining: Related technologies - Machine Learning, DBMS, OLAP, Statistics, Data Mining Goals, Stages of the Data Mining Process, Data Mining Techniques, Knowledge Representation Methods, Applications, Example: weather data

UNIT II 12Hrs

Data Warehouse and OLAP: Data Warehouse and DBMS, Multidimensional data model, OLAP operations, Example: loan data set

Data pre-processing: Data cleaning, Data transformation, Data reduction, Discretization and generating concept hierarchies, Introduction to Weka Data Mining System, Example experiments with Weka - filters, discretization.

Data mining knowledge representation: Visualization techniques, Experiments with Weka – visualization

UNIT III 10Hrs

Attribute-oriented analysis: Attribute generalization, Attribute relevance, Class comparison, Statistical measures, Experiments with Weka - using filters and statistics

Data mining algorithms -Association rules: Motivation and terminology, Example: mining weather data, Basic idea: item sets, Generating item sets and rules efficiently, Correlation analysis, Experiments with Weka - mining association rules

UNIT IV 12Hrs

Data mining algorithms- Classification: Basic learning/mining tasks, Inferring rudimentary rules: 1R algorithm, Decision trees, Covering rules, Experiments with Weka - decision trees, rules.

Data mining algorithms- Prediction: The prediction task, Statistical (Bayesian) classification, Bayesian networks, Instance-based methods (nearest neighbor), Linear models

UNIT IV 10Hrs

Evaluating what's been learned: Basic issues, Training and testing, Estimating classifier accuracy (holdout, cross-validation, leave-one-out), Combining multiple models (bagging, boosting, stacking), Minimum Description Length Principle (MLD), Experiments with Weka - training and testing

Clustering: Basic issues in clustering, First conceptual clustering system: Cluster/2, Partitioning methods: k-means, expectation maximization (EM), Hierarchical methods: distance-based agglomerative and divisible clustering, Conceptual clustering: Cobweb, Experiments with Weka - k-means, EM, Cobweb

References:

1. Ian H. Witten and Eibe Frank, Data Mining: Practical Machine Learning Tools and Techniques (Second Edition), Morgan Kaufmann,
2. Jiawei han, Micheline Kamber,Jian Pei, Data mining concepts and techniques, 3/e, Elsevier.
3. Margaret H. Dunham, Data Mining-Introductory and Advanced Topics, Pearson Education
4. K.P.Soman, Shyam Diwakar, and V. Ajay, Insight into Data Mining: Theory and Practice, Prentice Hall of India, 2006

17MScCSCS 4.3	**Digital Image Processing**	
Credits: 4	Teaching: 4Hrs/week	Max. Marks: 80 IA: 20

UNIT I 10Hrs

Digital Image Fundamentals: The origins of Digital Image Processing, Examples of Fields that Use Digital Image Processing, Fundamentals Steps in Image Processing, Elements of Digital Image Processing Systems, Image Sampling and Quantization, Basic relationships between pixels.

UNIT II 12Hrs

Image Enhancement in the Spatial Domain: Basic Gray Level Transformations, Histogram Processing, Enhancement Using Arithmetic and Logic operations, Basics of Spatial Filters, Smoothening and Sharpening Spatial Filters, Combining Spatial Enhancement Methods.

UNIT III 10Hrs

Image Enhancement in the Frequency Domain: Introduction to Fourier Transform and the frequency Domain, Smoothing and Sharpening Frequency Domain Filters, Homomorphic Filtering. Color Image Processing: Color models, pseudocolor image processing, color transformations, smoothening and sharpening.

UNIT IV 10Hrs

Image Restoration: A model of The Image Degradation / Restoration Process, Noise Models, Restoration in the presence of Noise Only Spatial Filtering, Periodic Noise Reduction by Frequency Domain Filtering, Linear Position-Invariant Degradations, Estimation of Degradation Function, Inverse filtering, Wiener filtering, Constrained Least Square Filtering, Geometric Mean Filter, Geometric Transformations.

UNIT V 10Hrs

Image Segmentation: Detection of Discontinuities, Edge linking and boundary detection, Thresholding, Region Oriented Segmentation, Motion based segmentation.
Morphological Processing: Some basic Morphological operations.

References:

1. Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, 3/e, Pearson Education.
2. Anil K. Jain, Fundamentals of Digital Image Processing', Pearson .
3. Kenneth R. Castleman, Digital Image Processing, Pearson.
4. Rafael C. Gonzalez, Richard E. Woods, Steven Eddins,Digital Image Processing using MATL AB, Pearson Education, Inc..
5. William K. Pratt, , Digital Image Processing, John Wiley, New York,
6. Milan Sonka et aI, Image Processing, Analysis And Machine Vision, Brookes/Cole, VikasPub lishing House.

17MScCSCS 4.4	**Computer Graphics**	
Credits: 4	Teaching: 4Hrs/week	Max. Marks: 80 IA: 20

UNIT I 10Hrs

Introduction: Application areas of Computer Graphics, overview of graphics systems, video-display devices, raster-scan systems, random scan systems, graphics monitors and work stations and input devices.

Output Primitives: Points and lines, line drawing algorithms, mid-point circle and ellipse algorithms. Filled area primitives: Scan line polygon fill algorithm, boundary-fill and flood-fill algorithms.

UNIT II 10Hrs

2-D Geometrical transforms: Translation, scaling, rotation, reflection and shear transformations, matrix representations and homogeneous coordinates, composite transforms, transformations between coordinate systems.

2-D Viewing : The viewing pipeline, viewing coordinate reference frame, window to view-port coordinate transformation, viewing functions, Cohen-Sutherland and Cyrus-beck line clipping algorithms, Sutherland –Hodgeman polygon clipping algorithm.

UNIT III 10Hrs

3-D Object representation: Polygon surfaces, quadric surfaces, spline representation, Hermite curve, Bezier curve and B-spline curves, Bezier and B-spline surfaces. Basic illumination models, polygon rendering methods.

3-D Geometric transformations: Translation, rotation, scaling, reflection and shear transformations, composite transformations, 3-D viewing: Viewing pipeline, viewing coordinates, view volume and general projection transforms and clipping.

UNIT IV 10Hrs

Visible surface detection methods: Classification, back-face detection, depth-buffer, scan-line, depth sorting, BSP-tree methods, area sub-division and octree methods.

UNIT V 10Hrs

Computer animation: Design of animation sequence, general computer animation functions, raster animation, computer animation languages, key frame systems, motion specifications.

References:

1. C, Foley, VanDam, Feiner and Hughes, Computer Graphics Principles & practice, second edition, Pearson Education
2. Donald Hearn and M. Pauline Baker, Computer Graphics C version, Pearson education.
3. Zhigand xiang, Roy Plastock, Computer Graphics Second edition, Schaum's outlines, Tata Mc Graw Hill.
4. David F Rogers, Procedural elements for Computer Graphics, Tata Mc Graw hill, 2nd edition.
5. Neuman and Sproul, Principles of Interactive Computer Graphics, Tata Mc Graw Hill,.

17MScCSCS 4.5	**a. Software Testing** **b. Cloud Computing** **c. Theory of Computation** **d. Object Oriented Analysis and Design Using UML**	
Credits: 4	Teaching: 4Hrs/week	Max. Marks: 80 IA: 20

17MScCSCS 4.6	**Prog. Lab.: Computer Graphics Lab.**	
Credits: 3	Lab. Duration: 06Hrs/week	Max. Marks: 80 IA: 20

Assignments shall be implemented using OpenGL/Java/C language.
Students shall understand C Graphics Basics Graphics programming, initializing the graphics, C Graphical functions, and write simple programs initially (draw line, circle and rectangle)

Suggested Assignments:

1. Drawing the basic primitives and sierpinsky gasket using openGL*.
2. Create a polyline using mouse interaction using openGL*.
3. Bresenham's line drawing algorithm.
4. Mid-Point ellipse drawing algorithm.
5. Implementation of Area Filling Algorithm: Boundary Fill / Flood Fill and/or Scan line Polygon Fill.
6. Program for performing Two Dimensional Transformations : Translation , Scaling , Rotation , Reflection , Shear by using a homogeneous Matrix representation, composite transformation
7. Curve Generation : Bezier for n control points , B Spline (Uniform)
8. Line clipping algorithm Cohen-Sutherland / Liang Barsky.
9. Polygon Clipping algorithm Sutherland Hodgeman.
10. Program to represent a 3D object using polygon surfaces and then perform 3D transformation
11. Fractal generation (Koch curve / Hilbert curve / peano curves using string production)
12. Program for Animation (eg. moving wheel, moving car, man walking with umbrella, flying flag, etc.)

17MScCSCS 4.6	**Prog. Lab.: PROJECT**	
Credits: 3	Lab. Duration: 06Hrs/week	Max. Marks: 80 IA: 20

Project work will be carried out in the department under the supervision of guide(s) approved by the Department of Computer Science.

- Each student shall carry out an individual project in the Lab.
- The Guide shall be concerned teacher in the department.
- The Project topic should be chosen in consultation with the guide.
- The Project topics shall be based on syllabus or beyond.
- Students shall submit the project proposal/synopsis at the beginning of the semester
- Student shall carry out the analysis and design work for the chosen problem statement and develop the s/w in the Lab.
- The students are required to submit a copy of project report(dissertation) based on the work done by him/her during the project period at the end of the semester term.
- The Evaluation scheme for the project work in the term exam shall be as follows
  - Internal assessment marks : 20

    Semester end Examination: Total Marks: 80
  - Dissertation : 35
  - S/w demo/presentation : 35
  - Viva-voce : 10

Core Electives

17MScCSCS 4.5	**a. Software Testing**	
Credits: 4	Teaching: 4Hrs/week	Max. Marks: 80 IA: 20

UNIT I 12Hrs

Basics of Software Testing and Examples: Basic definitions, Test cases, Insights from a Venn diagram, Identifying test cases, Error and fault taxonomies, Levels of testing. Examples: Generalized pseudocode, The triangle problem, The NextDate function, The commission problem, The SATM (Simple Automatic Teller Machine) problem. Decision Table-Based Testing: Decision tables, Test cases for the triangle problem, Test cases for the NextDate function, Test cases for the commission problem, Guidelines and observations. Data Flow Testing: Definition-Use testing, Slice-based testing, Guidelines and observations.

UNIT II 10Hrs

Levels of Testing: Traditional view of testing levels, Alternative life-cycle models, The SATM system, Separating integration and system testing. Integration Testing: A closer look at the SATM system, Decomposition-based, call graph- based, Path-based integrations, Case study. System Testing: Threads, Basic concepts for requirements specification, Finding threads, Structural strategies and functional strategies for thread testing, SATM test threads, System testing guidelines, ASF (Atomic System Functions) testing example.

UNIT III 10Hrs

Interaction Testing: Context of interaction, A taxonomy of interactions, Interaction, composition, and determinism, Client/Server Testing. Issues in Object-Oriented Testing: Units for object-oriented testing, Implications of composition and encapsulation, inheritance, and polymorphism, Levels of object-oriented testing, GUI testing, Dataflow testing for object-oriented software, Examples. Class Testing: Methods as units, Classes as units.

UNIT IV 10Hrs

Object-Oriented Integration Testing: UML support for integration testing, MM-paths for object-oriented software, A framework for object-oriented dataflow integration testing. GUI Testing: The currency conversion program, Unit testing, Integration Testing and System testing for the currency conversion program. Object-Oriented System Testing: Currency converter UML description, UML-based system testing, Statechart-based system testing. Exploratory Testing: The context-driven school, Exploring exploratory testing, Exploring a familiar example, Exploratory and context-driven testing observations.

UNIT V 10Hrs

Model-Based Testing: Testing based on models, Appropriate models, Use case-based testing, Commercial tool support for model-based testing. Test-Driven Development: Test-then-code cycles, Automated test execution, Java and JUnit example, Remaining questions, Pros, cons, and open questions of TDD, Retrospective on MDD versus TDD. A Closer Look at All Pairs Testing: The all-pairs technique, A closer look at NIST study, Appropriate applications for all pairs testing, Recommendations for all pairs testing. Software Testing Excellence: Craftsmanship, Best practice of software testing, Top 10 best practices for software testing excellence, Mapping best practices to diverse projects.

References:

1. Paul C. Jorgensen: Software Testing, A Craftsman's Approach, 3rd Edition, Auerbach Publications, 2012.
2. Aditya P Mathur: Foundations of Software Testing, Pearson, 2008.
3. Mauro Pezze, Michal Young: Software Testing and Analysis – Process, Principles and Techniques, 1st edition, John Wiley & Sons, 2011.
4. Srinivasan Desikan, Gopalaswamy Ramesh: Software testing Principles and Practices, 1st Edition, Pearson, 2012.
5. Brian Marrick: The Craft of Software Testing, 1st edition, Pearson, 2012

17MScCSCS 4.5	**b. Cloud Computing**	
Credits: 4	Teaching: 4Hrs/week	Max. Marks: 80 IA: 20

UNIT I 10Hrs

Introduction: Essentials, Benefits and need for Cloud Computing - Business and IT Perspective - Cloud and Virtualization - Cloud Services Requirements - Cloud and Dynamic Infrastructure - Cloud Computing Characteristics Cloud Adoption. Cloud Models: Cloud Characteristics - Measured Service - Cloud Models - Security in a Public Cloud Public versus Private Clouds.

UNIT II 08Hrs

Cloud Infrastructure Self Service. Cloud as a Service: Gamut of Cloud Solutions - Principal Technologies - Cloud Strategy Cloud Design and Implementation using SOA - Conceptual Cloud Model - Cloud Service Defined.

UNIT III 12Hrs

Cloud Solutions: Cloud Ecosystem - Cloud Business Process Management - Cloud Service Management - Cloud Stack - Computing on Demand (CoD) – Cloud sourcing. Cloud Offerings: Information Storage, Retrieval, Archive and Protection - Cloud Analytics Testing under Cloud - Information Security - Virtual Desktop Infrastructure - Storage Cloud. Cloud Management: Resiliency – Provisioning - Asset Management - Cloud Governance - High Availability and Disaster Recovery - Charging Models, Usage Reporting, Billing and Metering.

UNIT IV 10Hrs

Cloud Virtualization Technology: Virtualization Defined - Virtualization Benefits - Server Virtualization - Virtualization for x86 Architecture - Hypervisor Management Software - Logical Partitioning (LPAR) - VIO Server - Virtual Infrastructure Requirements. Cloud Virtualization: Storage virtualization - Storage Area Networks - Network-Attached storage - Cloud Server Virtualization - Virtualized Data Center.

UNIT V 12Hrs

Cloud and SOA: SOA Journey to Infrastructure - SOA and Cloud - SOA Defined - SOA and IaaS - SOA-based Cloud Infrastructure Steps - SOA Business and IT Services. Cloud Infrastructure Benchmarking: OLTP Benchmark - Business Intelligence Benchmark - e- Business Benchmark - ISV Benchmarks - Cloud Performance Data Collection and Performance Monitoring Commands - Benchmark Tools.

Reference s

1. Dr. Kumar Saurabh, Cloud Computing – Insight into New Era Infrastructure, Wiley India, 2011.
2. Roger Jennings, Cloud Computing, Wiley India, 2009.

3. John Rhoton, Cloud Computing Explained, Recursive Press,2009.
4. Barry Sosinsky, Cloud Computing Bible, Wiley, 2011.
5. Rajkumar Buyya, James Broberg, Cloud Computing: Principles and Paradigms, Wiley, 2011.
6. Judith Hurwiz, Cloud Computing for Dummies, Wiley Publishing, 2009.
7. Rosenberg and Matheos, The Cloud at your service, Manning Publications

17MScCSCS 4.5	**c. Theory of Computation**	
Credits: 4	Teaching: 4Hrs/week	Max. Marks: 80 IA: 20

UNIT I 10Hrs

Introduction: Automata, computability, and complexity: complexity theory, computability theory, automata theory.

Mathematical Notions and Terminology: Sets, sequences and tuples, functions and relations, graphs, strings and languages, Boolean logic, definitions, theorems, and proofs, types of proof-proof by construction, proof by contradiction, proof by induction.

UNIT II 12Hrs

Automata and Languages: Regular Languages: Finite Automata- definition, examples of finite automata, definition of computation, designing finite automata, the regular operations, Nondeterminism- definition, equivalence of nfas and dfas, closure under the regular operations. Regular expressions- definition, equivalence with finite automata. Nonregular languages-the pumping lemma for regular languages

Unit III 10Hrs

Context-Free Languages: Context-free Grammars—definition, Examples of context-free grammars, designing context-free grammars, ambiguity, Chomsky normal form, Pushdown Automata—definition, examples of pushdown automata, Equivalence with context-free grammars. Non-context-free languages--The pumping lemma for context-free languages,

Unit IV 10Hrs

Computability Theory: The Church-Turing Thesis: Turing Machines—definition, examples of Turing machines, variants of Turing Machines.

Enumerators, equivalence with other models. The definition of Algorithm—Hilbert's problems, Terminology for describing Turing machines.

Unit V 10Hrs

Decidability: Decidable Languages, Decidable problems concerning regular languages, Decidable problems concerning context-free languages. The Halting Problem—The diagonalization method, The halting problem is undecidable, A Turing-unrecognizable language.

Reducibility: Undecidable Problems from Language Theory, Reductions via computation histories, A Simple Undecidable Problem, Mapping Reducibility, Computable functions, definition of mapping reducibility

References:

1. Michael Sipser, Introduction to theory of computation, 2/e, Gale Cengage learning
2. K.L.P. Mishra and N. Chandrasekaran, Theory of Computer Science, 2/e, PHI.

3. J P Hoperoft, J D Ullman, Introduction to Automata, Languages and Computation, Narosa Publications.
4. John C. Martin, Introduction to Languages and the Theory of Computation, 2nd Edition, McGraw Hill

17MScCSCS 4.5	**d. Object Oriented Analysis and Design Using UML**	
Credits: 4	Teaching: 4Hrs/week	Max. Marks: 80 IA: 20

UNIT I 10Hrs

Overview of object-oriented systems, objects, attributes, encapsulation, class hierarchy, polymorphism, inheritance, messages, history of object orientation.

UNIT II 12Hrs

Introduction to UML, basic expression of classes, attributes, and operations, Class diagrams: generalization and association constructs, composition and aggregation. Use case diagrams, Object interaction diagrams: collaboration diagrams, sequence diagrams, asynchronous messages and concurrent execution. State diagrams: basic state diagrams, nested states, concurrent states and synchronisation, transient states. Activity diagrams

UNIT III 10Hrs

Architecture diagrams : packages, deployment diagrams for hardware artifacts and software constructs . Interface diagrams: window-layout and window-navigation diagrams.

UNIT IV 10Hrs

Encapsulation structure, connascence, domains of object classes, encumbrance, class cohesion, state-spaces and behavior of classes and subclasses, class invariants, pre-conditions and post-conditions, class versus type, principle of type conformance, principle of closed behavior.

UNIT V 10Hrs

Abuses of inheritance, danger of polymorphism, mix-in classes, rings of operations, class cohesion and support of states and behavior, components and objects, design of a component, light weight and heavy weight components, advantages and disadvantages of using components.

References:

1. Page-Jones .M, Fundamentals of object-oriented design in UML, Addison Wesely
2. Booch. G, Rumbaugh J, and Jacobson. I, The Unified Modelling Language User Guide, Addison Wesely.
3. Simon Bennett, MCrobb Rayfarmar, Object Oriented Systems Analysis and Design Using UML, 2/e, Tata McGraw Hill.
4. Booch. G, Rumbaugh J, and Jacobson. I, The Unified Modelling Language Reference Manual, Addison Wesely.
5. Larman.C, Applying UML & Patterns: An Introduction to Object Oriented Analysis & Design, Addison Wesley
6. Pooley R & Stevens P, Using UML: Software Engineering with Objects & Components, Addison Wesley

Semester End Examination Question Paper Pattern

Max Marks: 80 Duration - 3 Hours.

Theory Question Paper Pattern

- There shall be eight questions of 16 marks each.
- Each question may have sub questions (a),(b) / (a),(b),(c)
- There shall be a minimum of one question from each unit
- There shall be not more than 2 questions from any unit.
- The student has to answer any five full questions for scoring full marks

Internal Assessment Scheme

Internal Assessment– 20 Marks:

IA Test : 14 marks
Attendance : 03 marks
Seminar/assignment : 03 marks

Two tests shall be conducted, one during the mid of the semester and another at the end of the semester of 1hour duration each.

First IA Marks : 14
Second IA Marks : 14

Average of the two tests shall be taken as final marks.

Attendance Marks allocation scheme

Attendance (in percentage)	**90 and above**	**80 and above but below 90**	**75 and above but below 80**	**below 75**
Marks	3	2	1	no Marks*

*not eligible for appearing semester end examination (as per Regulation 7.7)

Note: Guidelines notified by the University from time-to-time shall be followed for IA.