SYLLABUS

NAME OF THE DEPARTMENT : COMPUTER SCIENCE

PROGRAM NAME : MASTER OF SCIENCE IN COMPUTER SCIENCE

PROGRAM SPECIFIC OUTCOMES (PSO):

The program will enable students to:

- Utilize and implement hardware and software technologies that provide computing solutions to address the needs of an organization.
- Provide socially acceptable technical solutions to complex computer science problems with application of modern and appropriate techniques for sustainable development relevant to professional engineering practices.
- Apply the knowledge of ethical and management principles inherent in the discipline of computing to work in a team as well as to lead a team.
- Use of knowledge in various domains of computer science in lifelong learning to adopt to innovations and change in the order to be successful.

COURSE OVERVIEW:

FIRST SEMESTER										
COURSE CODE	**COURSE NAME**	**COURSE TYPE**	**Lec.**	**Tut**	**Prac.**	**Cre.**	**Contact Hours/ Week**	**Total Marks : (A) + (B)**		**Nature**
								Internal (A)	**External (B)**	
CSC1016	Advanced Concepts in OOP	**Core**	**4**	**1**	**1**	**6**	**7**	**60**	**40**	**Graded**
CSC1026	Advanced Computer Organization and Architecture	**Core**	**4**	**2**	**0**	**6**	**6**	**60**	**40**	**Graded**
CSC1036	Operating System	**Core**	**4**	**1**	**1**	**6**	**7**	**60**	**40**	**Graded**
CSC1046	Mathematical Foundations of Computer Science	**Core**	**4**	**2**	**0**	**6**	**6**	**60**	**40**	**Graded**
CSC1056	Advanced DBMS	**Core**	**4**	**1**	**1**	**6**	**7**	**60**	**40**	**Graded**
Semester Total			**20**	**7**	**3**	**30**	**33**			

SECOND SEMESTER										
COURSE CODE	**COURSE NAME**	**COURSE TYPE**	**Lect.**	**Tut**	**Prac.**	**Cre.**	**Contact Hours/ Week**	**Total Marks : (A) + (B)**		**Nature**
								Internal (A)	**External (B)**	
CSC2016	Data Communication and Computer Networks	**Core**	**4**	**1**	**1**	**6**	**7**	**60**	**40**	**Graded**
CSC2026	Algorithms and Complexity Theory	**Core**	**4**	**2**	**0**	**6**	**6**	**60**	**40**	**Graded**
CSC2036	Software Engineering	**Core**	**4**	**1**	**1**	**6**	**7**	**60**	**40**	**Graded**
CSC2046	Computer Graphics and Multimedia	**Core**	**4**	**1**	**1**	**6**	**7**	**60**	**40**	**Graded**
CSC2056	Advanced Data Structure	**Core**	**4**	**0**	**2**	**6**	**8**	**60**	**40**	**Graded**
Semester Total			**20**	**5**	**5**	**30**	**35**			

THIRD SEMESTER										
COURSE CODE	**COURSE NAME**	**COURSE TYPE**	**Lect.**	**Tut**	**Prac.**	**Cre.**	**Contact Hours/ Week**	**Total Marks : (A) + (B)**		**Nature**
								Internal (A)	**External (B)**	
CSC3016	Theory of Computations	**Core**	**4**	**2**	**0**	**6**	**6**	**60**	**40**	**Graded**
CSC3026	Distributed System	**Core**	**4**	**1**	**1**	**6**	**7**	**60**	**40**	**Graded**
CSC3036	Compiler Design	**Core**	**4**	**1**	**1**	**6**	**7**	**60**	**40**	**Graded**
CSC3046	Seminar	**Core**	**0**	**0**	**0**	**3**	**6**	**50**	**0**	**Graded**
CSC3xx6		**Elective-I / Open**	**4**	**1**	**1**	**6**	**7**	**60**	**40**	**Graded**
Semester Total			**16**	**5**	**3**	**27**	**33**			

List of Electives

Elective Subjects for 3rd semester (Elective-I)

[All the following courses carry a total of 6 credits]

1. **CSC3056** Image Processing **(open)**
2. **CSC3066** Data Mining and Warehousing **(open)**
3. **CSC3076** Web Programming Technologies

FOURTH SEMESTER										
COURSE CODE	**COURSE NAME**	**COURSE TYPE**	**Lect.**	**Tut**	**Prac.**	**Cre.**	**Contact Hours/ Week**	**Total Marks : (A) + (B)**		**Nature**
								Internal (A)	**External (B)**	
CSC4016	Programming languages	**Core**	**4**	**1**	**1**	**6**	**7**	**60**	**40**	**Graded**
CSC4026	Project Work	**Core**	**0**	**0**	**4**	**6**	**8**	**0**	**100**	**Graded**
CSC4xx6		Elective- II /Open	**4**	**1**	**1**	**6**	**7**	**60**	**40**	**Graded**
CSC4xx6		Elective-III	**4**	**1**	**1**	**6**	**7**	**60**	**40**	**Graded**
Semester Total			**16**	**5**	**3**	**24**	**29**			

List of Electives

Elective Subjects for 4th semester (Elective-II and Elective-III)

[All the following courses carry a total of 6 credits]

1. **CSC4036** Embedded System
2. **CSC4046** Artificial Intelligence **(open)**
3. **CSC4056** Speech Processing **(open)**
4. **CSC4066** Applied Graph Theory and Algorithms
5. **CSC4076** System Administration and Networking
6. **CSC4086** Wireless Communication and Networks
7. **CSC4096** Queuing theory and Operations Research

DETAILED SYLLABUS:

SEMESTER NAME: FIRST SEMESTER

COURSE CODE**: CSC1016**	L-T-P**: 4-1-0**
COURSE NAME**: ADVANCED CONCEPTS IN OOP**	CONTACT HOURS/WEEK**: 6**
COURSE TYPE**: CORE**	TOTAL MARKS**: 100** (INTERNAL**: 60,** EXTERNAL**: 40**)
NUMBER OF CREDITS**: 6**	NATURE**: GRADED**

COURSE OBJECTIVES:

1. To familiarize students with concepts of data abstraction, polymorphism, inheritance, exception handling and file handling in Object Oriented Programming
2. To provide students with concepts of different Object Oriented Design approaches, Object Oriented Modeling (OMT) tools and phases of development.
3. To make students able to implement features of Object Oriented Programming solve real world problems.

COURSE PREREQUISITE:

- Basic knowledge of C programming

COURSE OUTCOMES:

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

- Differentiate between structured programming and Object Oriented Programming methodologies.
- Develop programs using different object oriented programming features such as data abstraction, polymorphism, inheritance, exception handling etc.
- Analyze different object Oriented Design Approaches and implement it to real life problems.
- Implement different Object Modeling Techniques (OMT) techniques.

COURSE CONTENT:

Unit No & Name	Components of the Unit	No of contact hours	Marks
UNIT-I: **Object Oriented Programming**	• **Introduction**: Definition and Concepts of Structured Programming and Object Oriented Programming paradigms. • **Data abstraction**: Object, class, member and friend functions, memory allocation for objects, constructors and destructors, templates. • **Inheritance**: Extending a class, casting up the hierarchy, single and multiple inheritances, virtual base class.	60	60

	• **Polymorphism**: Compile time polymorphism, operator overloading, function overloading, static binding, run-time polymorphism, virtual functions, pure virtual functions, abstract class, dynamic binding. • Exception handling, File handling,		
UNIT-II: **Object Oriented Design**	• **Introduction**: Object Oriented Design Approaches • Object Modeling Techniques (OMT) tools: Object Model, Dynamic Model, and Functional Model. (Object Diagram, State Diagram, and DFD). • Phases of Object-Oriented Development: Object Analysis, System Design, Object Design.	30	40
	Total :	90	100

TEXTBOOKS/ RECOMMENDED READINGS:

- Stroustrup, B. (1995) *The C++ Programming Language*, Addison Wesley Publishing Company
- Schild Herbert , *The Complete Reference to C++,* Osborne McGraw Hill.
- Rambaugh et al., *Object Oriented Modeling and Design*, P.H.I. (EEE).

COURSE ASSESSMENT DETAILS:

Internal assessment: Class tests, Assignments, Laboratory tests, Seminar (Two/ Three mid semester examinations will be conducted. 60% of this evaluation will be added to the total marks for this course.)

External assessment: End Semester Examination (End Semester Examination will be of 100 marks covering the entire course and the exam duration will be 3 hours. 40% of the mark obtained will be added to the total marks for this course)

COURSE CODE**: CSC1026** COURSE NAME**: ADVANCED COMPUTER ORGANIZATION AND ARCHITECTURE** COURSE TYPE**: CORE** NUMBER OF CREDITS**: 6**	L-T-P**: 4-2-0** CONTACT HOURS/WEEK**: 6** TOTAL MARKS**: 100** (INTERNAL**: 60,** EXTERNAL**: 40)** NATURE**: GRADED**

COURSE OBJECTIVES:

1. To provide students the knowledge of instruction set and addressing modes and how computer system works and its basic principles
2. To provide students the knowledge of I/O devices and its working principles
3. The give students the concepts of memory system and its classifications
4. To familiarize students with concepts of advanced pipelining techniques

COURSE PREREQUISITE:

- Fundamental knowledge of computer Organization and Architecture

COURSE OUTCOMES:

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

- Explain the basic instruction set architectures of computer.
- Analyze different Input-Output systems and memory organization used in computer architecture.
- Understand control unit operations.
- Describe different parallel architectures.

COURSE CONTENT:

Unit No & Name	Components of the Unit	No of contact hours	Marks
UNIT-I: **Instruction Set Architecture**	• Instruction set design, addressing modes • representation of data (character, integral, floating point)	6	7
UNIT-II: **Computer Arithmetic**	• Serial adder, parallel adder, ripple carry adder, carry look-ahead adder • Multiplication of signed and unsigned numbers, Booth's algorithm, division of integer, floating point arithmetic.	8	8
UNIT-III: **Processor Design**	• Register transfer language, one, two and three bus data path • ALU Design, control unit, hardwired control	10	10

	unit, micro programmed control unit.		
UNIT-IV: **Memory**	• Classification and types of memory. • Cache memory, direct mapped, associative mapped and set associative mapped cache. • Cache replacement policies, write policy, unified, split and multilevel cache • Virtual memory, paging, segmentation.	10	10
UNIT-V: **Input Output System**	• I/O buses, device controller, Interrupt and DMA. • Interrupt driven I/O, Program controlled I/O and DMA transfer.	12	15
UNIT-V: **Parallel Architectures**	• Classification, SISD, SIMD, MISD, MIMD, Scalar, vector, superscalar and pipelined processor, Pipelining, Instruction pipeline, pipeline bubbles, Hazards: -resource conflicts, data dependency, branch difficulty. • Vector computing, arithmetic pipeline, vector and scalar register, chaining, scatter gather operations, vector-register processor, Memory vector processor. Array processor.	22	25
UNIT-VI: **Advanced concepts**	• Branch prediction, super pipelining, Branch delay slot, Register file, superscalar architecture, superscalar pipelines, superscalar branch prediction, out of order execution, register renaming • Pipeline scheduling, dynamic scheduling and static scheduling algorithms, reorder buffer and register renaming, Thronton technique and scoreboard. Tomasulo algorithm and reservation stations. • VLIW architecture: - EPIC architecture, Multiprocessor systems: - Interconnection types. Cache coherence problem	22	25
	Total:	90	100

TEXTBOOKS/ RECOMMENDED READINGS:

- Govindarajalu, B. *Computer Architecture and Organization*, TMH publication.
- Richard Y. Kain, *Advanced Computer Architecture A systems Design Approach*, PHI Publication
- Stallings William, *Computer Organization and Architecture Designing for Performance*, Pearson Education
- M. Morris Mano, *Computer System Architecture*, PHI Publication

COURSE ASSESSMENT DETAILS:

Internal assessment: Class tests, Assignments, Seminar (Two/ Three mid semester examinations will be conducted. 60% of this evaluation will be added to the total marks for this course.)

External assessment: End Semester Examination (End Semester Examination will be of 100 marks covering the entire course and the exam duration will be 3 hours. 40% of the mark obtained will be added to the total marks for this course)

COURSE CODE**: CSC1036**	L-T-P**: 4-1-1**
COURSE NAME**: OPERATING SYSTEM**	CONTACT HOURS/WEEK**: 7**
COURSE TYPE**: CORE**	TOTAL MARKS**: 100** (INTERNAL**: 60,** EXTERNAL**: 40)**
NUMBER OF CREDITS**: 6**	NATURE**: GRADED**

COURSE OBJECTIVES:

1. To provide students the basic concepts of operating system such as process states, I/O organization and instruction sets
2. To familiarize the students with the concepts of deadlock handling in Operating system
3. To provide students the knowledge of scheduling, multiprogramming and memory management
4. The give students the knowledge of multiprogramming system

COURSE PREREQUISITE:

- Basic concepts of computer fundamentals

COURSE OUTCOMES:

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

- Recognize the design approaches of advanced operating systems such as memory architectures, scheduling, deadlock handling etc.
- Analyze the design issues of distributed operating systems.
- Evaluate design issues of multi processor operating systems.
- Identify the requirements of database operating systems and formulate the solutions to schedule the real time applications.

COURSE CONTENT:

Unit No & Name	Components of the Unit	No of contact hours	Marks
UNIT-I: **Review of computer organization**	• Major subsystems, instruction sets • I/O organization.	8	10
UNIT-II: **Memory architecture**	• Address protection, segmentation, virtual memory, paging, page replacement algorithms, cache memory • Hierarchy of memory types, associative memory.	12	15
UNIT-III: **Support for concurrent process**	• Mutual exclusion, shared data, critical sections, busy form of waiting	8	10

	• Lock and unlock primitives, synchronization block and wakeup.		
UNIT-IV: **Scheduling**	• Process states, process scheduling queue, schedulers, virtual processors, interrupt mechanism • Scheduling algorithms: First-Come, First-Served (FCFS) Scheduling, Shortest-Job-Next (SJN) Scheduling, Priority Scheduling, Shortest Remaining Time, Round Robin(RR) Scheduling, Multiple-Level Queues Scheduling, Implementation of concurrency primitive.	32	30
UNIT-V: **System deadlock**	• Deadlock characterization, Resource Allocation Graph, Prevention, detection and avoidance of deadlock • Banker's algorithm, detection algorithm	10	10
UNIT-VI:**Multiprogramming System**	• Queue management, I/O supervisors, memory management, File system, disk and drum scheduling. • Case Study: Some real operating system– semaphores, messages, shared memory.	12	15
UNIT-VII: **Advanced Topics**	• Secondary storage management, Security, Distributed operating system	8	10
	Total:	90	100

TEXTBOOKS/ RECOMMENDED READINGS:

- Tanenbaum, A. S. and Woodhull, A. S. *Operating Systems Design and Implementation* , PHI
- Stallings, W., *UNIX Network programming*, PHI.
- Kerninghan and Pike, *The UNIX programming Environment*, PHI.
- Peterson , J. L. and Silberschatz , A., *Operating System concepts*', Addison – Wesley
- Stallings, W. , *Operating Systems*, PHI
- Silberschatz, A., and Galvin, P. , *Operating System Concepts*, Addison-Wesley

COURSE ASSESSMENT DETAILS:

Internal assessment: Class tests, Assignments, Laboratory tests, Seminar

External assessment: End Semester Examination

COURSE CODE**: CSC1046** COURSE NAME**: MATHEMATICAL FOUNDATIONS OF COMPUTER SCIENCE** COURSE TYPE**: CORE** NUMBER OF CREDITS**: 6**	L-T-P**: 4-2-0** CONTACT HOURS/WEEK**: 6** TOTAL MARKS**: 100** (INTERNAL**: 60,** EXTERNAL**: 40)** NATURE**: GRADED**

COURSE OBJECTIVES:

1. To introduce the students with the basic concepts of Mathematics which form the foundation of Computer Science.
2. To introduce topics like graph theory, mathematical logic which has tremendous applications in Computer Science.
3. To introduce the concept of Automata Theory that is used in the development of finite state machines and also in the analysis of algorithms.

COURSE PREREQUISITE:

- Basic knowledge of Programming and Discrete Mathematics.

COURSE OUTCOMES:

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

- Illustrate the definitions and theorems in basic discrete mathematics
- Formulate simple definitions, examples and proofs in discrete mathematics
- Implement the concepts of formal languages, automata and grammars, and the relation between them
- Describe basic concepts of Graph theory and Tree properties

COURSE CONTENT:

Unit No & Name	Components of the Unit	No of contact hours	Marks
UNIT-I: **Discrete mathematical structures**	• Congruence, permutation and combination with repetitions. • Basic concepts of sets. The principle of inclusion and exclusion. Fuzzy sets. • Relations, binary relations, closure of relations. • Functions; Posets and Lattices. • Boolean Algebra; Boolean functions (SOM and POM).	25	30

	• Algebraic structures- Groups, Free groups, Permutation groups. Homomorphism and Isomorphism. Vector Spaces and its properties, basis and dimension. Linear transformations and linear operators.		
UNIT-II: **Mathematical Logic**	• Connectives- statement formulae and truth tables, tautologies and tautological implications, two-state devices and statement logic; • Theory of inference- rules, consistency of premises and indirect method of proof, automatic theorem proving; • Propositional calculus • Predicate calculus- predicates, quantifiers, predicate formulas, free and bound variables, inference theory of predicate calculus; validity, soundness, completeness, compactness (definitions only). • Resolution principles; Skolemization and Herbrand domain; • Introduction to axiomatic theory.	25	25
UNIT-III **Graph theory**	• Basic concepts- finite and infinite graphs, incidence and degree, isolated and pendant vertices, null graph. • Paths and Circuits- isomorphism, subgraphs, walks, connected and disconnected graphs and components. • Euler graphs, Bi-partite graphs, Hamiltonian paths and circuits. • Trees- properties of trees, distance and centers, rooted and binary trees, counting trees, spanning trees, fundamental circuits, spanning trees in weighted graphs; Cut-sets- properties, connectivity and separability. • Network flows; Matrix representation of graphs- incidence matrix, submatrices, circuit matrix, cut-set matrix, path matrix, adjacency matrix; Coloring, Covering and Partitioning- basic concepts;	25	25

	• Directed graphs- definition, types, directed paths and connectedness, Euler digraph, tress with directed edges.		
UNIT-IV **Automata theory**	• Concept of language and grammar. • Review of DFA, NFA, NFA with empty moves and their equivalence. Minimization of FA. • Regular sets and regular expressions. Pumping lemma for regular sets, closure properties and decision algorithms for regular sets. • Context free language – definition, removal of useless symbols, removal of null productions and unit productions. • Normal forms of CFLs- CNF and GNF.	15	20
	Total	90	100

TEXTBOOKS/ RECOMMENDED READINGS:

- Tremblay, J. P., Manohar, R., *Discrete Mathematical Structures with Applications to Computer Science*, McGraw Hill.
- Liu, C. L., *Elements of Discrete Mathematics*, McGraw Hill.
- Gallier, J. H., *Logic for Computer Science*, J. Willey & Sons.
- Lewis, H. R., Papadimitriou, C. H., *Elements of the theory of computation*, PHI.
- Deo, N., *Graph Theory with applications to Engineering and Computer Science*, PHI

COURSE ASSESSMENT DETAILS:

Internal assessment: Two/ Three mid semester examinations will be conducted. 60% of this evaluation will be added to the total marks for this course.

External assessment: End Semester Examination will be of 100 marks covering the entire course and the exam duration will be 3 hours. 40% of the mark obtained will be added to the total marks for this course.

COURSE CODE**: CSC1056** COURSE NAME**: ADVANCED DATABASE MANAGEMENT SYSTEM** COURSE TYPE**: CORE** NUMBER OF CREDITS**: 6**	L-T-P**: 4-1-1** CONTACT HOURS/WEEK**: 7** TOTAL MARKS**: 100** (INTERNAL**: 60,** EXTERNAL**: 40)** NATURE**: GRADED**

COURSE OBJECTIVES:

1. To give students the concept of relational model, relational algebra and relational calculus
2. To familiarize students with the concept of normalization and ER-designing
3. To provide students the basic knowledge of distributed database systems, Query processing, concurrency control and recovery in distributed databases.
4. To provide students the concepts of spatial databases, Image and multimedia

COURSE PREREQUISITE:

- Basic concepts of database management system
- Basic knowledge of SQL query language

COURSE OUTCOMES:

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

- Distinguish different type of Relational models, databases and schemas and construction of various relational algebra and calculus.
- Analyze the concepts of ER-model, Functional dependencies and normalization techniques.
- Describe the concepts of different transactions, concurrency control techniques, database recovery techniques.
- Analyze the basic concepts and use of Object-oriented, Distributed, Image, Multimedia and Spatial databases

COURSE CONTENT:

Unit No & Name	Components of the Unit	No of contact hours	Marks
UNIT-I: **Relational model**	• Relational model concepts, relational databases and schemas • Relational algebra operations, queries in relational algebra • Overview of relational calculus; Commercial query language SQL- data definition, constraints, SQL queries, insertion, deletion, updation.	13	15
UNIT-II: **Semantic modeling**	• Introduction, E-R model, E-R diagrams, design of database with E-R model, Transformation of ER	13	15

	model to relational schema • Extended ER diagram, Generalization, Aggregation		
UNIT-III: **Normalization and functional dependencies**	• Design guidelines, functional dependencies – equivalence of sets of functional dependencies, cover, minimal cover • Normal forms- 1NF, 2NF, 3NF, BCNF , 4NF, • Dependency-preserving property , lossless join property , algorithms to ensure dependency - preserving property and lossless join property	20	20
UNIT-IV: **System implementation techniques**	• Query processing and optimization- translation between SQL queries and relational algebra • Transaction processing- transaction and system concepts, desirable properties, schedules and recoverability • Concurrency control- locking techniques, concurrency control based on timestamp ordering, multiversion concurrency control techniques • Database recovery- concepts and techniques, recovery in multidatabase systems; Security and authentication- issues, access control techniques, introduction to multilevel security.	18	20
UNIT-V: **Object oriented database systems**	• Concepts of object-oriented databases; Standards, languages and design • Object relational database systems.	9	10
UNIT-VI: **Distributed databases**	• Concepts; Data fragmentation, replication, and allocation techniques • Types of distributed database systems; Query processing in distributed databases • Overview of concurrency control and recovery in distributed databases.	9	10
UNIT-VII: **Image, multimedia, and spatial databases**	• Concepts of Image, multimedia, and spatial databases • Content-based indexing and retrieval, indexing techniques- R trees, R+ trees, KD trees.	8	10
	Total:	90	100

TEXTBOOKS/ RECOMMENDED READINGS:

- Elmasri, R., Navathe, S. B., *Fundamentals of Database Systems*, Pearson Education.
- Date, C. J., *An Introduction To Database Systems*, Pearson Education.
- Stamper, D., Price, W., *Database Design and Management- An Applied Approach*, McGraw Hill.
- Prabhu, C. S. R., *Object-Oriented Database Systems- Approaches and Architectures*, PHI.
- Ullman, J. D., *Principles of Database Systems*, Galgotia.

COURSE ASSESSMENT DETAILS:

Internal assessment: Class tests, Assignments, Laboratory tests

External assessment: End Semester Examination

SEMESTER NAME: SECOND SEMESTER

COURSE CODE**: CSC2016** COURSE NAME**: DATA COMMUNICATION AND COMPUTER NETWORKS** COURSE TYPE**: CORE** NUMBER OF CREDITS**: 6**	L-T-P**: 4-1-1** CONTACT HOURS/WEEK**: 7** TOTAL MARKS**: 100** (INTERNAL**: 60,** EXTERNAL**: 40)** NATURE**: GRADED**

COURSE OBJECTIVES:

1. To provide the knowledge of essential of data communication and networking including a study of the Open Systems Interconnection (OSI), TCP/IP and Internet models.
2. To provide knowledge of different transmission modes, different network standards and protocols.
3. To provide the basic concept of LAN and Wireless LAN, Network Security etc

COURSE PREREQUISITE:

- Basic Concept of Computer Network.

COURSE OUTCOMES:

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

- Describe different synchronous and asynchronous transmission technologies.
- Identify the different types of network topologies and protocols.
- Explain data communication system and its components.
- Implement the skill of sub-netting and routing mechanism.

COURSE CONTENT:

Unit No & Name	Components of the Unit	No of contact hours	Marks
UNIT-I: Introduction	• Data Communication concepts and terminologies: Data representation, Data transmission, Transmission channels, Signal encoding, Transmission impairments, Transmission media: Guided transmission media (*Twisted pair, Coaxial and Optical fiber);* Wireless transmission (*Terrestrial microwave, satellite microwave, Broadcast Radio and Infrared)*	10	10
UNIT-II: Transmission Modes	• Asynchronous and Synchronous transmission,	8	6

	Baseband and Broadband transmission, Modulation methods, Modems, Multiplexing.		
UNIT-III: Evolution of Computer Networks	• Evolution of computer networks: Circuit switching, Development of packet switching: 1961-1972, Proprietary networks and internetworking: 1972-1980, Proliferation of networks: 1980-1990. The internet explosion: 1990s	12	10
UNIT-IV: Network Standards and protocols	• Network standards and protocols: The IEEE standards, OSI 7 layer model, TCP/IP protocol suit. Data Link Layer: Frame design, Flow control, Error handling, HDLC, PPP, Sliding window protocol.	15	18
UNIT-V: Different Network Layers	• Network Layer: IP, X.25, Frame Relay, ATM, Routing, Queuing theory. Transport Layer: TCP, UDP, Congestion control, Flow control, Socket interface. Application Layer: SNMP, Authentication, Encryption, Web and HTTP, FTP, Email, DNS, Network File System (NFS) and File sharing, Remote Procedure Calling (RPC).	20	24
UNIT-VI: Introduction to LAN, Architecture and Technology	• Local Area Network (LAN): Needs, Architecture and Technology, Ethernet: CSMA/CD operation, parameters and specifications, Cabling: 10Base5, 10Base2, 10BaseT, 10BaseF, Hubs, patch panels and wiring closets. Bridges, Switches, 100BaseT, 100BaseVGANY,Gigabit Ethernet. FDDI, Token Ring, Wireless, ISDN, B-ISDN	15	20
UNIT-VII: Wireless LAN, Network Management and Security	• VSAT technology, Wireless LAN: Technologies, IEEE standards and protocols. • Basics of Network management and Security, Infrastructure for network management and security.	10	12
	Total :	90	100

TEXTBOOKS/ RECOMMENDED READINGS:

- Stallings, W.; *Data and Computer Communications*; Prentice Hall of India.
- Tanenbaum A.S.; *Computer Networks*; Prentice Hall of India Education
- Kurose and Ross; *Computer Networking*; Addison Wesley
- Prakash C. Gupta; *Data Communication*; Prentice Hall of India

COURSE ASSESSMENT DETAILS:

Internal assessment: Class tests, Assignments, Laboratory tests, Seminar

External assessment: End Semester Examination

COURSE CODE**: CSC2026** COURSE NAME**: ALGORITHMS AND COMPLEXITY THEORY** COURSE TYPE**: CORE** NUMBER OF CREDITS**: 6**	L-T-P**: 4-2-0** CONTACT HOURS/WEEK**: 6** TOTAL MARKS**: 100** (INTERNAL**: 60,** EXTERNAL**: 40)** NATURE**: GRADED**

COURSE OBJECTIVES:

1. To teach the students how to analyze computer algorithms.
2. To introduce some domain independent algorithm design techniques.
3. To acquaint the students with the classification of problems based on existence of efficient algorithms.

COURSE PREREQUISITE:

- Discrete Mathematics, CSC1046, CSC 1016

COURSE OUTCOMES:

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

- Determine the best, average, worst time complexity of algorithms.
- Know the big O, omega and theta notations and their usage to give asymptotic upper, Lower and tight bounds on time and space complexity of algorithms.
- Analyze major sorting and order statistics algorithms and solve problems using fundamental graph algorithms.
- Define the classes P and NP and explain the significance of NP completeness.

COURSE CONTENT:

Unit No & Name	Components of the Unit	No of contact hours	Marks
UNIT-I: **Analysis of Algorithms:**	• Concepts in algorithm analysis, time and space complexity**.** • Review of asymptotic notations (O, o, θ, ω, Ω) used for time complexity. • Common Mathematical functions-monotonicity, floors and ceilings, polynomials, exponentials, logarithms, factorials, iterated logarithmic functions. Relational properties of asymptotic notations. • Asymptotic behaviors of polynomials,	25	25

	relative asymptotic growth, ordering functions by asymptotic growth rates. • Recurrences - substitution method, iteration method (using recursion tree), using Master theorem (proof of the theorem is not included). • Introduction to the concept of amortized analysis. Use of aggregate method to some simple problems like stack operations and incrementing binary counter.		
UNIT-II: **Algorithm Design Techniques**	• Algorithm design techniques – Divide and Conquer, Dynamic programming, Greedy Algorithm, Back-tracking, Branch and Bound. Illustration of design techniques by application to some specific problems such as: sorting and searching, matrix manipulation problems, knapsack problem, internal and external sorting problem, job sequencing problem, set manipulation problem. • Dynamic storage allocation, garbage collection.	25	25
UNIT-III **Graph Algorithms**	• Representation of graphs – adjacency matrix and adjacency list. • Depth-first search and breadth-first search, topological sort. • Minimum spanning tree – Kruskal's and Prim's algorithm • Single source shortest path problem and algorithm due to Dijkstra.	20	22
UNIT-IV **Theory of NP-Completeness**	• Formal language framework, complexity classes – P, NP. co-NP. • Reducibility and NP-Completeness, NP-Hard.	10	12
Unit - V **Lower Bound Theory**	• Computing lower bounds for sorting, merging, finding maximum and second maximum, minimum and maximum	6	9

	simultaneously.		
Unit – VI **Overview of more Algorithm Design Techniques**	• Basic idea about neural network and genetic algorithm.	4	7
	Total	90	100

TEXTBOOKS/ RECOMMENDED READINGS:

- Cormen. T. H., Leiserson C. E. and Rivest. R. L., 3rd edition (2010); *Introduction to Algorithms*, Tata-Mcgraw Hill Publishers.
- Horowitz and Sahani; (2nd Edition) *Fundamentals of Computer Algorithms*, Galgotia.
- Aho.A, Hopcroft J.E. and Ullman J.D.; (2011), *Design and Analysis of Computer Algorithms*, Pearson Education.
- S. Baase and Allen Van Gelder, (3rd edition), *Computer Algorithms-Introduction to Design and Analysis*, Pearson Education, LPE.

COURSE ASSESSMENT DETAILS:

Internal assessment: Two mid semester examinations will be conducted apart from sudden class tests and home assignments. 60% of this evaluation will be added to the total marks for this course.

External assessment: End Semester Examination will be of 100 marks covering the entire course and the exam duration will be 3 hours. 40% of the mark obtained will be added to the total marks for this course.

COURSE CODE**: CSC2036**	L-T-P**: 4-1-1**
COURSE NAME**: SOFTWARE ENGINEERING**	CONTACT HOURS/WEEK**: 7**
COURSE TYPE**: CORE**	TOTAL MARKS**: 100** (INTERNAL**: 60,** EXTERNAL**: 40)**
NUMBER OF CREDITS**: 6**	NATURE**: GRADED**

COURSE OBJECTIVES:

1. To provide students the knowledge of SE challenges, Software process, S/W development process model and problem analysis.
2. The give students the concepts of role of software architecture, architecture views, software cost estimation model, quality plan, and risk management.
3. To familiarize students with concepts of module level concept, OO Analysis and OO Design, UML, Coding process, refactoring, verification, testing fundamentals.

COURSE PREREQUISITE:

- Basic knowledge of database management system.

COURSE OUTCOMES:

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

- Compare different software development processes and their challenges.
- Create software require specification and translate it into an implementable design, following a structured and organize process.
- Implement different software estimation metrics such as cost, effort size, staffing etc.
- Make effective use of UML, along with design strategies such as defining software architecture, separation of concerns and design patterns.

COURSE CONTENT:

Unit No & Name	Components of the Unit	No of contact hours	Marks
UNIT-I: **Software challenges and Software process**	• Problem domain, SE challenges • SE approach. Software process, Characteristics of SW process, SW development process model.	10	10
UNIT-II: **Software requirement and specification**	• SW requirement, problem analysis, requirement specification. • Functional specification, validation, matrices.	10	15
UNITIII:**Software architecture views and cost**	• Role of SW architecture, architecture view, component and connector view, style for C&C	30	35

estimation model.	view. • Process planning, Effort estimation, Software Cost Estimation based on COCOMO II cost model. • Scheduling and staffing. • SW configuration management plan, quality plan, risk management, project monitoring plan.		
UNIT-IV: **Design principles and Methodology**	• Design principle • Module level concept • Design notation and specification, • Structured design methodology verification. • OO Analysis and OO Design. OO Design concept, UML. OO Design methodology.	25	25
UNIT-V: **Detail Design and Testing fundamentals**	• Detail design and PDL • Verification, Metrices, Programming principles and guidelines, coding process, refactoring, verification. Testing fundamentals.	15	15
	Total	90	100

TEXTBOOKS/ RECOMMENDED READINGS:

- Jalote P. ; *An integrated Approach to Software Engineering*; Narosa Publishing House
- Patton R.; *Software Engineering;* Pearson Education.
- Agarwal K. K., Singh Y,; *Software Engineering*; New Age International Publisher.
- Sommerville I.; *Software Engineering*; Pearson Education (Addison Wesley)
- Pressman R.S.; *Software Engineering: A practitioner's Approach*; McGraw Hill.

COURSE ASSESSMENT DETAILS:

Internal assessment: Class tests, Assignments, Laboratory tests

External assessment: End Semester Examination

COURSE CODE**: CSC2046** COURSE NAME**: COMPUTER GRAPHICS AND MULTIMEDIA** COURSE TYPE**: CORE** NUMBER OF CREDITS**: 6**	L-T-P**: 4-1-1** CONTACT HOURS/WEEK**: 7** TOTAL MARKS**: 100** (INTERNAL**: 60,** EXTERNAL**: 40**) NATURE**: GRADED**

COURSE OBJECTIVES:

1. To introduce students the basic concepts of hardware, software and applications of computer graphics.
2. To familiarize students with the concepts of computer graphics such as input/output systems, line drawing algorithms, area filling algorithms, clipping algorithms etc those are essential for designing computer graphics software.
3. To provide experience of graphics programming by completing several programming assignments.

COURSE PREREQUISITE:

- Programming knowledge of C.
- Concept of basic linear algebra (matrices and vector geometry).

COURSE OUTCOMES:

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

- Explain basic knowledge of core concepts of computer graphics input/output systems.
- Analyze different techniques such as line drawing algorithms, area filling algorithms, clipping algorithms etc that are essential for designing computer graphics software.
- Solve problems 2D/3D geometric transformations and viewing techniques.
- Differentiate different Colour models, visible surface rendering method, methods

COURSE CONTENT:

Unit No & Name	Components of the Unit	No of contact hours	Marks
UNIT-I: **Introduction and overview of Graphics systems**	• **Introduction:** Computer graphics and its applications; Input devices; Output devices-display devices; Display techniques- Raster-scan display and Random-scan display; color display techniques; Direct view storage tubes; emissive & non-emissive flat-panel displays-Plasma panels, Thin-film electrostatic displays, LED, LCD; Three-dimensional viewing devices;	15	10

	display systems architecture. • **Graphics software:** classifications, graphics functions for various operations, software standards- PHIGS, PHIGS+, GKS.		
UNIT-II: **Output primitives**	• **Line-drawing algorithms**: DDA algorithm and Bresenham's algorithm. • **Midpoint algorithms for circle & ellipse generation.** • **Area-filling algorithms**: scan-line polygon-fill, nonzero-winding number rule, scan-line curve filling, boundary-fill algorithm, flood-fill algorithm. • **Character generation techniques:** generation of bitmap and outlined font.	20	20
UNIT-III: G**eometric transformations**	• **2-D geometric transformations:** Basic transformations- translation, rotation and scaling; matrix representations and Homogeneous co-ordinate representations; Composite transformations among translation, rotation and scaling; General pivot-point rotation; General fixed-point scaling; General scaling directions; Other transformations- reflection and shear; Transformation between co-ordinate systems; Definition of Affine transformations. • **3-D geometric transformations:** Translation; Rotation- rotations about co-ordinate axes, general 3-D rotation; Scaling; Reflection; Shear.	15	20
UNIT-IV: V**iewing and Clipping**	• **2-D viewing:** definition; viewing transformation pipeline; window-to-viewport co-ordinate transformation. • **2-D Clipping operations:** definition; point clipping; line clipping algorithms; polygon clipping algorithms; curve clipping, text clipping. • **3-D viewing:** viewing transformation pipeline;	15	20

	world co-ordinate to viewing co-ordinate transformation.		
UNIT-V: **3D Graphics**	• **3-D concepts:** display methods- Parallel projection, perspective projection, depth visible line & surface identification, surface rendering, exploded & cutaway views, 3-D & stereoscopic views. • **Projections:** Parallel projection techniques- orthographic & oblique projections and their transformation equations; Perspective projection and transformation equations. • **Visible surface detection:** definition; classification of algorithms- object-space methods & Image-space methods; algorithms for visible surface detection; curved-surface detection; wireframe displays. • **Illumination and Surface rendering:** definition and importance; light sources; Definition of basic illumination models. • **Color models and applications:** properties of light; standard preliminaries- XYZ model, CIE chromaticity diagram; color models- RGB, YIQ, CMY, HSV, HLS; conversion between color models.	15	20
UNIT-VI: **Multimedia Systems and Computer Animation**	• **Multimedia Systems:** Review of typical interactive multimedia systems, Aspects of multimedia systems, Multimedia design techniques, Multimedia technology; Network-based multimedia systems. • **Computer Animation:** Traditional animation techniques, 2D animation, 3D animation. • **Case Study:** Graphics API with GD or OpenGL or DirectX/3D.	10	10
	TOTAL	90	100

TEXTBOOKS/ RECOMMENDED READINGS:

- Hearn D., Baker M. P.; *Computer Graphics C Version*; PHI.
- Foly, Dam V., Feiner, Hughes; *Computer Graphics principles and practice*; Pearson Education.
- Xiang Z., Plastock R. A.; (2006); *Computer Graphics*, McGraw Hill.
- Sinha N., Udai A. D.; (2008); *Computer Graphics*, McGraw Hill

COURSE ASSESSMENT DETAILS:

Internal assessment: Class tests, Assignments, Laboratory tests

External assessment: End Semester Examination

COURSE CODE**: CSC2056** COURSE NAME**: ADVANCED DATA STRUCTURE** COURSE TYPE**: CORE** NUMBER OF CREDITS**: 6**	L-T-P**: 4-0-2** CONTACT HOURS/WEEK**: 8** TOTAL MARKS**: 100** (INTERNAL**: 60,** EXTERNAL**: 40)** NATURE**: GRADED**

COURSE OBJECTIVES:

1. To teach efficient storage mechanisms of data for an easy access.
2. To design and implementation of various basic and advanced data structures.
3. To introduce various techniques for representation of the data in the real world.
4. To develop application using data structures.

COURSE PREREQUISITE:

- Fundamentals of C/C++ programming

COURSE OUTCOMES:

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

- Analyze linear and non-linear data structures like stacks, queues, linked list etc.
- Solve problems related to data dictionary data structures such as search tree, AVL tree, Red Black trees, Splay trees and Hashing.
- Compare different Sorting and Searching techniques such as Quick sort, Heap Sort, Radix Sort, Counting Sort, BST, Median and Order Statistics and Heap data structures.
- Implement B tree, B+ tree used for external storage operation.

COURSE CONTENT:

Unit No & Name	Components of the Unit	No of contact hours	Marks
UNIT-I: **Review of basic concepts in Data Structure**	• A quick review of array versus linked list structure; binary tree, binary search tree; traversal, insertion and deletion in binary search trees.	10	10
UNIT-II: **Dictionary ADT**	• Search trees, balancing of search trees – AVL trees, Red-Black trees, multi way search trees, 2-3 trees, splay trees. Insertion and Deletion in each of the above data structures. Hashing.	20	20

UNIT-III: **Sorting and Selection Techniques**	• Quick sort, Heap sort, Shell sort, sorting in linear time – Counting sort, Radix sort. Medians and order Statistics. Selection and Adversary arguments. Lower bound on sorting	15	20
UNIT-IV**: Priority Queue ADT**	• Heaps-extended priority queue, min (max) heaps, binomial heap, fibonacci heap and its amortized analysis.	20	20
UNIT-V: **Partition ADT**	• Union-find algorithms through weighted merge and path compression.	15	15
UNIT-VI: **Data Structure for external storage operations**	• B-tree, insertion and deletion in B-trees, external sorting. B+ tree.	10	15
	TOTAL	90	100

TEXTBOOKS/ RECOMMENDED READINGS:

- Cormen T. H., Leiserson C. E., Rivest R. L.; *Introduction to Algorithms*; Tata-McGraw Hill Publishers
- Aho A., Hopcroft J. E., Ullman J. D.; *Data Structures and Algorithms*; Addison-Wesley
- Horowitz, Sahani; *Fundamentals of Data Structures in C/C++*; Computer Science Press
- Aho A., Hopcroft J. E., Ullman J. D.; *Design and Analysis of Computer Algorithms* ; Addison-Wesley

COURSE ASSESSMENT DETAILS:

Internal assessment: Class tests, Assignments, Laboratory tests

External assessment: End Semester Examination

SEMESTER NAME: THIRD SEMESTER

COURSE CODE**: CSC3016**	L-T-P**: 4-2-0**
COURSE NAME**: THEORY OF COMPUTATIONS**	CONTACT HOURS/WEEK**: 6**
COURSE TYPE**: CORE**	TOTAL MARKS**: 100** (INTERNAL**: 60,** EXTERNAL**: 40**)
NUMBER OF CREDITS**: 6**	NATURE**: GRADED**

COURSE OBJECTIVES:

1. To introduce the students with the concept of automatic procedures.
2. To introduce higher models of computations.
3. To acquaint the students with the existence of un-decidable problems.

COURSE PREREQUISITE:

- Basic knowledge of Programming, Discrete Mathematics, Finite Automata, Language and Grammar.

COURSE OUTCOMES:

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

- Analyze and design finite automata, pushdown automata, Turing machines, formal languages, and grammars.
- Demonstrate the understanding of key notions, such as algorithm, computability, decidability, and complexity through problem solving.
- Prove the basic results of the Theory of Computation.
- State and explain the relevance of the Church-Turing thesis.

COURSE CONTENT:

Unit No & Name	Components of the Unit	No of contact hours	Marks
UNIT-I: **Extending Finite-State Automata**	• Definition of Pushdown Automaton (PDA), Instantaneous description. Pushdown Automaton as language acceptor. Equivalence of acceptance with final state and empty stack. Deterministic PDA and language accepted by it. • Equivalence of PDA's and Context free languages. • Relation of Pushdown Automaton with Finite State Automata. • Pumping lemma for CFL's and its applications. Closure and decision properties of CFL's.	30	35
UNIT-II: **Turing Machines**	• The standard TM having a two way infinite	30	25

	tape and its definition. • Turing machines as language accepters function evaluators. Recursively enumerable and recursive languages. Recursive and partial recursive functions. Working out examples of each. • Other models of Turing machines. TM with Stay-option, Multi track TM, TM with one way infinite tape, Multi-tape Turing machines, Multi-dimensional Turing machines, Multi-head TM, non-deterministic Turing machines and their equivalence.		
UNIT-III **Un decidability**	• Decidable and Un-decidable problems. • Properties of recursive and recursively enumerable languages. • Universal Turing machine and universal language. Encoding of TM's. • Rice's theorem (statement only) and applications. • Church Turing hypothesis.	18	25
UNIT-IV **The Chomsky Hierarchy**	• Left and Right linear grammar and relation with Regular sets. • Unrestricted grammars and TM. • Context-sensitive languages and linear bounded automata. • Relations between classes of languages, the Chomsky hierarchy.	12	15
	Total	90	100

TEXTBOOKS/ RECOMMENDED READINGS:

- Hopcroft, J.E. and J.D. Ullman, (2008); *Introduction to Automata Theory, Languages and Computation*, Pearson.
- Linz, Peter. (2008) ; *An introduction to Formal Languages and Automata*, Narosa Publishing House.
- Michael Sipser, (2nd Edition); *Introduction to the Theory of Computation*; Thomson (India).
- Mishra, K. L. P., Chandrasekaran, N, (2006); *Theory of Computer Science (Automata, Languages and Computation)*, P. H. I.

COURSE ASSESSMENT DETAILS:

Internal assessment: Two mid semester examinations will be conducted apart from sudden class tests and home assignments. 60% of this evaluation will be added to the total marks for this course.

External assessment: End Semester Examination will be of 100 marks covering the entire course and the exam duration will be 3 hours. 40% of the mark obtained will be added to the total marks for this course.

COURSE CODE**: CSC3026**	L-T-P**: 4-1-1**
COURSE NAME**: DISTRIBUTED SYSTEMS**	CONTACT HOURS/WEEK**: 7**
COURSE TYPE**: CORE**	TOTAL MARKS**: 100** (INTERNAL**: 60,** EXTERNAL**: 40)**
NUMBER OF CREDITS**: 6**	NATURE**: GRADED**

COURSE OBJECTIVES:

1. To provide students the concepts of basic architecture and components of distributed systems
2. To familiarize the students with the concepts of various distributed algorithms.
3. To give students the concepts of concurrency controlling and distributed file system handing

COURSE PREREQUISITE:

- Basic concepts of Operating Systems and basic concepts of Computer networks

COURSE OUTCOMES:

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

- Explain the architecture and different system models of distributed systems.
- Analyze different process synchronization, Global state recording and termination detection algorithms in distributed systems.
- Compare different Mutual Exclusion, leader election algorithms, different distributed file structures
- Distinguish the Inter-process communication methods and analyze the idea of failure handling, concurrency management and Security handling issues

COURSE CONTENT:

Unit No & Name	Components of the Unit	No of contact hours	Marks
UNIT-I: **Introduction to Distributed Systems**	• Definition of a distributed system. Characteristics of distributed and centralized systems, Design issue and challenges, types of transparency issues, openness, and scalability. Hardware concepts- multiprocessors, homogeneous & heterogeneous systems, middleware, issues in distributed Operating systems, inherent limitations of distributed systems • System models: Fundamental and Architectural model, System architectures- The client-server model and its variations, application layering, client-server architectures.	18	20

UNIT-II: **Synchronization**	• Needs of clock synchronization, external and internal clock synchronization, Logical and vector clocks, Lamport's logical clock, Vector clocks, Causal Order of messages, Birman-Schiper-Stephension protocol, Schiper-Eggli-Sandoz protocol, • Global state, Chandy Lamport snapshot algorithm, termination detection, Haung's algorithm	18	20
UNIT-III: **Distributed Mutual Exclusions**	• Requirements of Mutual Exclusion algorithms, Performance measurement metrics, Classification of mutual exclusion algorithm, Token based algorithms, Non-token based algorithm, Central Server Algorithm, Lamport's timestamp algorithm, Ricart-Agrawala Algorithm, Maekawa's Voting algorithm, Ring based algorithm, Suzuki-Kasami's Broadcast algorithm, Raymond's Tree-based algorithm • Election algorithms- the Bully algorithm, Ring algorithm. Mutual exclusion- definition, algorithms.	18	20
UNIT-IV: **Distributed Scheduling and Deadlock detection**	• Distributed scheduler, issues in distributed load distribution, components of load distribution algorithm, stability, task migration • Basic conditions of deadlocks, Resource and communication deadlock, Strategies of deadlock handling, issues in deadlock detection and resolution, Deadlock detection algorithms (Centralized, Distributed , Hierarchical)	9	10
UNIT-V: **Agreement Protocols and Inter-process Communication**	• System models, classification of agreement problems (Byzantine, Consensus, Interactive), Solutions to the Byzantine agreement problem, Applications of agreement algorithms • Inter-process Communications, API for UDP/TCP, Request Reply Protocol, Remote Procedure Call- basic RPC operation, parameter passing, examples. • Remote Object Invocation- distributed objects, integrating clients and objects, static versus dynamic RMI, parameter passing, examples and case study.	9	10
UNIT-VI: **Naming**	• Naming entities- names, identifiers & addresses, name resolution, name space implementation, the Domain	4	5

	Name System.		
UNIT-VII: **Distributed Transaction Processing**	• Distributed transactions- ACID properties, flat and nested transactions, atomic commit protocols, concurrency control in distributed transactions, Introduction, reasons for replication, object replication, consistency models	7	7
UNIT-VIII: **Distributed File Systems**	• Introduction: characteristics of file systems, distributed file system requirements, File service architecture, file accessing models, detailed case study of Sun Network File System (NFS).	7	8
	Total:	90	100

TEXTBOOKS/ RECOMMENDED READINGS:

- Tanenbaum & Steen; (2004); *Distributed Systems Principles and Paradigms*; Pearson Education
- Coulouris, Dollimore & Kindberg; (2006); *Distributed Systems Concepts and Design*; Pearson Education

COURSE ASSESSMENT DETAILS:

Internal assessment: Class tests, Assignments, Laboratory tests

External assessment: End Semester Examination

COURSE CODE**: CSC3036**	L-T-P**: 4-1-1**
COURSE NAME**: COMPILER DESIGN**	CONTACT HOURS/WEEK**: 7**
COURSE TYPE**: CORE**	TOTAL MARKS**: 100** (INTERNAL**: 60,** EXTERNAL**: 40**)
NUMBER OF CREDITS**: 6**	NATURE**: GRADED**

COURSE OBJECTIVES:

1. To provide the students the concepts of various of phases of compiler design
2. To familiarize the students the concepts parsing, code generation and code optimization
3. To enable students to design a basic compile systems

COURSE PREREQUISITE:

- Basic concepts of theory of computation (**CSC3016**)

COURSE OUTCOMES:

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

- Implement Lexical analyzer such as NFA, DFA, Regular Expressions and parser generator tools.
- Derive different syntax analysis such as LR, SLR and LALR parsing
- Build symbol tables and generating intermediate code.
- Explain DAG and formulate code optimization solution for a give problem

COURSE CONTENT:

Unit No & Name	Components of the Unit	No of contact hours	Marks
UNIT-I: **Introduction**	• What is a compiler? Phases of compiler. Overview of working of a compiler	8	10
UNIT-II: **Lexical Analysis**	• NFA, DFA, conversion from NFA to DFA. Regular expression. • Regular expression to NFA conversion. Minimisation of DFA. Writing a lexical analyser for C using Lex	12	15
UNIT-III: **Syntax analysis**	• Grammar representation. Derivation and parse tree. Ambiguity and possible elimination. • Top down parsing. Recursive descent and predictive top down parsing. Elimination of Left recursion. • Bottom up parsing. Operator precedence parsing, LR parsing (including SLR and LALR). Error detection	23	25

	and recovery. Parser table construction. Writing a parser for a subset of C using yacc.		
UNIT-IV: **Code generation**	• Symbol table contents, implementation. Type checking. Syntax directed translation. Forms of intermediate codes. Abstract Syntax Trees, Directed Acyclic Graph, Three address code. • Intermediate code generation for different language constructs like arrays, boolean expressions, if, if-else, while, case or switch, function calls. Writing a intermediate code generator and an interpreter for the intermediate code for the parser developed in 3 above. Target code generation issues. Runtime storage management.	23	25
UNIT-V: **Code Optimisation**	• DAG, basic blocks, Common sub-expression elimination, variable propagation, code motion, strength reduction, elimination of dead code, loop optimization. • Data flow analysis, objects, integrating clients and objects, static versus dynamic RMI, parameter passing, examples and case study.	22	25
	Total:	90	100

TEXTBOOKS/ RECOMMENDED READINGS:

- Aho, Sethi, Ullman; *Compilers, Principles, Techniques*, Tools; Pearson Education
- Aho A.V., Sethi R., Ullman J.D.*; Introduction to Compiler Construction*; Pearson Education.
- Holub.; *Compiler Design in C*; P.H.I.
- Chattopadhyay S.*; Compiler Design*; P.H.I.
- Hunter; *The Essence of Compilers*; Pearson Education.

COURSE ASSESSMENT DETAILS:

Internal assessment: Class tests, Assignments, Laboratory tests

External assessment: End Semester Examination

COURSE CODE**: CSC3046**	L-T-P**: 0-0-3**
COURSE NAME**: SEMINAR**	CONTACT HOURS/WEEK**: 6**
COURSE TYPE**: CORE**	TOTAL MARKS**: 50** (INTERNAL**: 50,** EXTERNAL**: 0**)
NUMBER OF CREDITS**: 6**	NATURE**: GRADED**

COURSE OBJECTIVES:

1. The objective of the project is to train the student to independently search, identify and study real-life important topics in CS/IT,
2. To develop skills among students in a particular field of CS/IT; and to expose students to the world of technology, innovation, and research

COURSE PREREQUISITE:

- Basic knowledge of computer fundamental, hardware & software, concepts of basic programming such as C/C++ /Java etc.

COURSE OUTCOMES:

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

- Analyze their ideas on selected topics on recent technologies
- Perform their communication and presentation skill.
- Write report on a given topic.

COURSE CONTENT:

Student should take a particular domain of research such as database, data mining, computer networking, image processing, speech and signal processing, algorithm, embedded system, cloud computing, soft computing etc. It is expected that at-least one standard methodology/algorithm should apply on a particular domain. The accuracy of the output should be tested using some standard accuracy measuring tools or statistical tools.

COURSE ASSESSMENT DETAILS:

Internal assessment: Each student will have to deliver at least 3 seminars talks on topics assigned by the department.

COURSE CODE**: CSC3056**	L-T-P**: 4-1-1**
COURSE NAME**: IMAGE PROCESSING**	CONTACT HOURS/WEEK**: 7**
COURSE TYPE**: ELECTIVE/OPEN**	TOTAL MARKS**: 100** (INTERNAL**: 60,** EXTERNAL**: 40**)
NUMBER OF CREDITS**: 6**	NATURE**: GRADED**

COURSE OBJECTIVES:

1. To learn the fundamental concepts of Digital Image Processing.
2. To study basic Image Processing operations and image analysis algorithms.
3. To expose students to current applications in the field of Digital Image Processing.

COURSE PREREQUISITE:

- Discrete Mathematics, Knowledge of C/C++ programming

COURSE OUTCOMES:

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

- Apply principles and techniques of digital image processing in applications related to digital imaging system design and analysis.
- Analyze and implement image different transformation and enhancement technique such as DFT, FFT, Filtering, Histogram processing
- Differentiate different image compression techniques.
- Compare different binary image processing techniques.

COURSE CONTENT:

Unit No & Name	Components of the Unit	No of contact hours	Marks
UNIT-I: **Digital image processing systems**	• Image acquisition, storage, processing, communication, display.	6	6
UNIT-II: **Visual Perception**	• Structure of the human eye, image formation in the human eye, brightness, adaptation and discrimination	8	8
UNIT-III: **Image Model**	• Uniform and non-uniform sampling, quantization	8	10
UNIT-IV: **Image Transforms**	• Introduction to Fourier transform, DFT and two dimensional DFT, some properties of DFT, separability, translation, periodicity, conjugate symmetry, rotation, scaling, average value,	12	10

	convolution theorem, correlation, FFT algorithms, inverse FFT, filter implementation through FFT. Other transforms: Other separable image transforms and their algorithms.		
UNIT-V: **Image Enhancement**	• Image enhancement in spatial domain and frequency domain, Histogram processing. Spatial Filtering, Frequency Domain Filtering.	10	12
UNIT-VI: **Image Restoration**	• Restoration/Degradation Model, Inverse Filtering, Wiener Filtering	8	10
UNIT-VII: **Edge Detection and Segmentation**	• Edge detection, Line detection, Segmentation, Texture Analysis and Classification.	10	12
UNIT-VIII: **Binary Image Processing**	• Binarisation, Morphological Image Processing, Distance Transform.	8	8
UNIT-IX**: Color Image Processing**	• Color model, Color Image Quantisation, Histogram of a colour image.	10	12
UNIT-X **Image Compression**	• Lossy Compression, Loss-less compression, Run-length and Huffman Coding, Transform Coding, Image Compression Standards. Other Advanced Web Technologies- AJAX, ISAPI, .NET.	10	12
	Total:	90	100

TEXTBOOKS/ RECOMMENDED READINGS:

- R. C. Gonzalez & R. E. Woods - Digital Image Processing, Addison Wesley, 1993.
- A. K. Jain - Fundamentals of Digital Image Processing, PHI
- K. R. Castleman - Digital Image Processing, PHI 1996
- W. K. Pratt - Digital Image Processing, John Wiley Interscience, 1991

COURSE ASSESSMENT DETAILS:

Internal assessment: Class tests, Assignments, Laboratory tests

External assessment: End Semester Examination

COURSE CODE**: CSC3066** COURSE NAME**: DATA MINING AND WAREHOUSING** COURSE TYPE**: ELECTIVE/OPEN** NUMBER OF CREDITS**: 6**	L-T-P**: 4-1-1** CONTACT HOURS/WEEK**: 7** TOTAL MARKS**: 100** (INTERNAL**: 60,** EXTERNAL**: 40**) NATURE**: GRADED**

COURSE OBJECTIVES:

1. To introduce students the basic concepts of Data Warehouse and techniques and applications of Data Mining.
2. To develop skills for designing and implementing systems for data mining to solve practical problems in a variety of disciplines.
3. To provide students the experience of doing independent study and research.

COURSE PREREQUISITE:

- Programming knowledge of C, C++.
- Basic knowledge of Mathematics-Statistics.
- Basic concepts of Database.

COURSE OUTCOMES:

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

- explain the components and architecture of data warehouse architecture
- Illustrate different data mining techniques such as association rule mining, clustering and classification.
- Analyze different data mining algorithms such as K-means, DBSCAN, FR-tree growth, A priori, CURE, BIRC, ROCK, CART, C4.5 etc.
- Analyze the uses of developing areas-web mining, text mining and sequential data mining.

COURSE CONTENT:

Unit No & Name	Components of the Unit	No of contact hours	Marks
UNIT-I: **Introduction to Data Mining**	• **Basic Concepts:** Data Mining, kinds of patterns that can be mined, Data Mining versus Database systems, Data preparation, cleaning and visualization. • **Data Warehousing:** Differences between database systems and Data Warehouse, Data Warehouse architecture and its components, Warehouse versus Data Mining (OLTP & OLAP), OLAP tools, Data cubes, Multidimensional Data.	20	30

UNIT-II: **Data Mining Techniques**	• **Association Rules:** What is an association rule? Mining association rules, frequent sets and border sets, algorithms for mining association rules – A priori algorithm, Pincer-search algorithm, Border algorithm, FP-tree growth algorithm, generalized association rule, association rule with item constraints. • **Clustering:** Hierarchical versus Partitional clustering, types of data in clustering, Partitional algorithms – k-means, k-mediods, PAM, CLARA, CLARANS. Density based clustering algorithm – DBSCAN. Hierarchical algorithms – BIRCH, CURE. Categorical clustering algorithms – ROCK, CACTUS. • **Decision Trees :** Introduction, tree construction principle, decision tree generation algorithms – CART, ID3, C4.5 • **Other techniques for Data Mining** : Concepts of Genetic algorithms, Artificial Neural Network and Rough sets and their application in the domain of data mining. Introduction to Web Mining, Text Mining, Temporal data mining.	70	70
	TOTAL	90	100

TEXTBOOKS/ RECOMMENDED READINGS:

- Puzari K.; *Data Mining Techniques*; University Press
- Han J., Kamber M.; *Data Mining Concepts and Techniques*; India Morgan Kaufmann & Harcourt
- Soman K. P., Diwakar S., Ajay V.; (2008); *Insight into Data Mining*: *Theory and Practice* ; P.H.I (Eastern Economy Edition
- Jain K. and Dukes R. C.; *Algorithms for Clustering Data*; Prentice-Hall
- Cios K., Pedrycz W., Swiniarski R; (1998); *Data Mining : Methods of Knowledge Discovery*; Boston Kluwer Academic Publishers, ,

COURSE ASSESSMENT DETAILS:

Internal assessment: Class tests, Assignments, Laboratory tests, Seminar

External assessment: End Semester Examination

COURSE CODE**: CSC3076** COURSE NAME**: WEB PROGRAMMING AND TECHNOLOGIES** COURSE TYPE**: ELECTIVE/OPEN** NUMBER OF CREDITS**: 6**	L-T-P**: 4-1-1** CONTACT HOURS/WEEK**: 7** TOTAL MARKS**: 100** (INTERNAL**: 60,** EXTERNAL**: 40**) NATURE**: GRADED**

COURSE OBJECTIVES:

1. To provide the basic concepts of Internet and different Internet services, such as, telnet, e-mail, FTP, etc.
2. To enable students to design basic web page using HTML, JavaScript, Ajax, CSS.
3. To provide basic knowledge of markup languages, their structures, elements, content models, DTD, attributes, etc.
4. To introduce the basics of XML such as, how to display XML contents, how to use user-defined tags in web pages, use of XSL, etc.
5. To give brief introduction to some server-side scripting language, such as PHP, JSP, ASP, etc.

COURSE PREREQUISITE:

- Basics of programming languages and database management system

COURSE OUTCOMES:

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

- Analyze a web page and identify its elements and attributes.
- Create interactive web applications using AJAX. Differentiate between client side web technologies and server side web technologies
- Apply languages like HTML, DHTML, CSS, XML, JavaScript, VBScript, ASP, PHP, JSP, Servlet and protocols in the workings of the web and web applications
- Analyze a web project and identify its elements and attributes and build customize web sites and web applications

COURSE CONTENT:

Unit No & Name	Components of the Unit	No of contact hours	Marks
UNIT-I: **Internet basics**	• History and basic idea of Internet. • Different types of Internet services, such as, telnet, e-mail, ftp, WWW, etc.	5	8
UNIT-II: **Web page design**	• Designing web pages with HTML. • Use of different tags, hyper-links, URLs, tables, text formatting, graphics & multimedia, image-map, frames and forms in web pages.	8	10

	• Use of Cascading Style Sheet in web pages.		
UNIT-III: **Creating interactive and dynamic web pages with JavaScript**	• Basics of JavaScript overview, such as, constants, variables, operators, expressions & statements, • Creating user-defined & built-in functions, • Client-side form validation using properties and methods of built-in objects.	8	8
UNIT-IV: **Markup language basics**	• Basics of Standard Generalized Markup Language (SGML) such as structures, elements, Content models, DTD, attributes, entities.	10	10
UNIT-V: **Extensible Markup Language (XML):**	• Introduction to using user-defined tags in web pages, • Displaying XML contents, • Creating XML DTDs, • Use of XSL.	12	8
UNIT-VI: **Web Browsers**	• Functions and working principle of web browsers, • Plug-ins & helper applications, • Conceptual architecture of some typical web browsers.	5	10
UNIT-VII: **Introduction to Client/Server Computing**	• Basics of client-server computing, • Types of Client/Server systems, such as, middleware, 2-tier/3-tier/4-tier/ N-tier systems, • Fat Clients versus Fat Servers.	10	12
UNIT-VIII: **Web Servers**	• Web services and web server functionality, • Web server composition and registration, • Basics of HTTP, IP address, DNS & ports, • Conceptual architecture of some typical web servers.	8	8
UNIT-IX**: Server-side scripting**	• overview of CGI, ASP, and JSP, • Server side scripting using PHP, • Web database connectivity- introduction to ODBC, • PHP with database connectivity.	8	10
UNIT-X**: Exposure to Advanced Web Technologies**	• Distributed Object based models- DCOM, CORBA, EJB, • Web services and Related Technologies, such as, ISAPI, SOAP, UDDI, WSDL, • Other Advanced Web Technologies- AJAX, ISAPI, .NET.	8	8
UNIT-XI: **Web Security**	• Basics of Firewalls, such as, definition and their	8	8

	uses, • Types of Firewalls, such as, Network layer firewalls and application layer firewalls, • Proxy servers.		
	Total:	90	100

TEXTBOOKS/ RECOMMENDED READINGS:

- Oliver, Dick, Sams; *Teach Yourself Html 4 in 24 Hours*; Techmedia.
- Ashbacher, Charles, Sams; *Teach Yourself XML in 24 Hours*; Techmedia.
- Phil B., M. Michael, Sams; *Teach Yourself JavaScript in 24 Hours*; Techmedia.
- Julie C. M., Sams; *Teach Yourself PHP in 24 Hours*, Techmedia.
- Lehnert, Wendy. G.; *Web 101: Making the Net for you*; Pearson Education.
- Robert W. S.; *World Wide Web Programming*; Pearson Education.

COURSE ASSESSMENT DETAILS:

Internal assessment: Class tests, Assignments, Laboratory tests, Seminar

External assessment: End Semester Examination

SEMESTER NAME: FOURTH SEMESTER

COURSE CODE**: CSC4016**	L-T-P**: 4-1-1**
COURSE NAME**: PROGRAMMING LANGUAGES**	CONTACT HOURS/WEEK**: 7**
COURSE TYPE**: Core**	TOTAL MARKS**: 100** (INTERNAL**: 60,** EXTERNAL**: 40)**
NUMBER OF CREDITS**: 6**	NATURE**: GRADED**

COURSE OBJECTIVES:

1. To introduce the major programming paradigms, and the principles and techniques involved in design and implementation of modern programming languages.
2. To introduce notations to describe syntax and semantics of programming languages.
3. To analyze and explain behavior of simple programs in various programming paradigms using concepts such as binding, scope, control structures, subprograms and parameter passing mechanisms.
4. To introduce the concepts of concurrency control and exception handling

COURSE PREREQUISITE:

- CSC 1016, CSC 2056

COURSE OUTCOMES:

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

- Differentiate between different types of programming paradigms.
- Analyze semantic issues associated with function implementations, including variable binding, scoping rules, parameter passing, and exception handling.
- Implement functional programming (LISP), logical programming (PROLOG), Object Oriented Programming (C++/Java)

COURSE CONTENT:

Unit No & Name	Components of the Unit	No of contact hours	Marks
UNIT-I: **Programming Language concepts**	• Factors influencing the evolution of programming languages - influence of architecture and operating system, implementation methods. Development in programming methodology, desirable features and design issues. Language processors. Syntax, semantics and Virtual Computers, Binding and Binding time.	18	20

UNIT-II: **Imperative Programming Languages**	• Statements, data types, subprograms, sequence control, data control, dynamic allocation using pointers, operating and programming environment, Subprogram activation- parameter passing methods, scope rules for names. Nested procedures. Syntax and translation.	18	20
UNIT-III: **Object Oriented Languages**	• Data abstraction: object oriented thinking, class, grouping of data and operations, constructors and destructors, templates. • Inheritance: Extending a class, casting up the hierarchy, single and multiple inheritances, virtual base class. • Polymorphism: Compile time polymorphism, operator and function overloading, static binding, run-time polymorphism, virtual functions, pure virtual functions, abstract class, dynamic binding. • Exception handling. • [As OOP has been covered in semester 1 in depth, a quick review of the above concepts will be made.]	18	20
UNIT-IV: **Functional Programming Languages**	• Principles of functional programming. Types-values, bindings and functions, environment and scope, recursive functions, polymorphic functions, type variables. • Lists and programming with lists (LISP). • Functional programming in C++.	18	20
UNIT-V: **Logic Programming Languages**	• Review of Predicate Logic. Logic as a language for problem solving. Facts, rules, queries and deductions, sentence structure. General structure and computational behavior of logic programs. Unification algorithm. Procedural interpretation of Logic. Algorithmic view of logic program execution. A brief introduction to PROLOG.	18	20
	Total:	90	100

TEXTBOOKS/ RECOMMENDED READINGS:

- T.W. Pratt and M. V. Zelkowitz: Programming Languages: Design and Implementation; PHI.
- Ravi Sathi, Programming Languages, Concepts and Constructs, Pearson Education, Asia, LPE
- B. Stroustrup, The C++ Programming Language, Addison Wesley Publishing Company, 1995.
- W. Lloyd, Foundations of Logic Programming, Springer 1984.
- Carlo Ghezzi, Mehdi Jazayeri, Programming Language Concepts, J. Wiley & sons.
- E. Horowitz : Fundamentals of Programming Languages; Galgotia Publications Pvt Ltd.
- K. C. Louden; Programming Languages-Principles and Practice; Thompson (2 nd Indian Edition);

COURSE ASSESSMENT DETAILS:

Internal assessment: Class tests, Assignments, Laboratory tests, Seminar

External assessment: End Semester Examination

COURSE CODE**: CSC4026**	L-T-P**: 0-0-4**
COURSE NAME**: PROJECT**	CONTACT HOURS/WEEK**: 8**
COURSE TYPE**: CORE**	TOTAL MARKS**: 100** (INTERNAL**: 0,** EXTERNAL**: 100**)
NUMBER OF CREDITS**: 8**	NATURE**: GRADED**

COURSE OBJECTIVES:

1. The objective of the project is to train the student to independently search, identify and study real-life important topics in CS/IT,
2. To develop skills among students in a particular field of CS/IT; and to expose students to the world of technology, innovation, and research.

COURSE OUTCOMES:

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

- Search, identify, study and work on real-life applications of CS/IT independently.
- Implement their thoughts and ideas to develop new innovative solutions.
- Write their findings and analysis in the form of a dissertation
- Develop new systems whiting a time bound.

COURSE CONTENT:

Each student will be assigned some project work at the starting of the sixth semester. Each student (or group of at most 2 students) is expected to take a unique problem under the guidance/supervision of a faculty member of the department. The problem should be such that the students get a chance to explore one or two technologies in depth and grab good command over those technologies after successful completion of the project. Repetition of the problems already attempted by students of the previous years should not be encouraged unless the problem has exceptionally great research importance and scope. Application problems, if found interesting and arisen at the demand of a particular situation, may also be assigned; but typical information management systems with just two or three simple database tables and/or data-entry forms are to be discouraged. The project may be done in other Institutes/Organizations with prior permission from the concerned department of the College and in this case also one project supervisor should have to be from the concerned department in the College. The work will have to be submitted in the form of a dissertation. Project presentation and evaluation will have to be done as per the regulation of PG course semester system of G.U. with choice based credit and grading system.

COURSE ASSESSMENT DETAILS:

Internal assessment: seminars, presentations, viva, project implementation

COURSE CODE**: CSC4036**	L-T-P**: 4-1-1**
COURSE NAME**: EMBEDDED SYSTEM**	CONTACT HOURS/WEEK**: 7**
COURSE TYPE**: ELECTIVE**	TOTAL MARKS**: 100** (INTERNAL**: 60,** EXTERNAL**: 40**)
NUMBER OF CREDITS**: 6**	NATURE**: GRADED**

COURSE OBJECTIVES:

1. Provide students the concepts of various processors used in embedded system
2. Familiarize students with the concepts of I/O programming and memory management in embedded systems
3. Train students to design new embedded applications

COURSE PREREQUISITE:

- Basic knowledge of C/C++ programming
- Knowledge of Computer Organization and Architecture (CSC 1026)

COURSE OUTCOMES:

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

- Analyze different embedded processor architecture such as 8085, 8051, 8086.
- Distinguish different Concurrency control and Scheduling methodologies used in embedded technology.
- Explain different I/O programming.
- Design of systems on RTOS based embedded software in developing a complex embedded system product.

COURSE CONTENT:

Unit No & Name	Components of the Unit	No of contact hours	Marks
UNIT-I: **Introduction**	• Definition of embedded system, Why is it special? • Types of processor used in embedded systems, what are the other peculiarities?	8	10
UNIT-II: **Processors for embedded systems**	• 8 bit processors 8085, 8051 and PIC 18FXX: - Architecture and instruction set. (Already covered in microprocessor) 16 bit: - 8086 32 bit: - 80386 architecture and instruction set, ARM based processor architecture and instruction set.	8	10
UNIT-III: **I/O programming**	• Operating systems for embedded systems, Real time operating systems Issues • Synchronization, transfer rate and latency. Polled I/O issues. Interrupt driven I/O. ISR. Response time interrupt controller. Software interrupts and	14	15

	exceptions. Buffering of data and queuing of interrupt requests.		
UNIT-IV: **Concurrency control**	• Foreground/Background systems, Thread state and serialization, latency, prevention of interrupt overruns. Concurrent execution of threads, context switch, non-preemptive multitasking, preemptive multitasking. Critical sections:- disabling interrupts, disabling ask switch, spin lock, mutex and semaphore.	15	17
UNIT-V: **Scheduling in Embedded Systems**	• Conventional scheduling, deadline driven scheduling, rate monotonic scheduling, deadlock, watchdog timer.	13	15
UNIT-VI: **Memory Management**	• Static allocation, dynamic allocation. Recursion and dynamic allocation. shared memory, reentrant functions.	14	15
UNIT-VII: **Boot up and System initialization**	• 80x86 microprocessor with a C compiler (suited for RTOS) and uC/OS RTOS may be used for practical. • Some real embedded application shall be taken up for practical.	18	18
	Total:	90	100

TEXTBOOKS/ RECOMMENDED READINGS:

- Daniel W Lewis; *Fundamentals of Embedded Software*; Pearson Education
- David E. Simon; *An Embedded Software Primer*; Pearson Education

COURSE ASSESSMENT DETAILS:

Internal assessment: Class tests, Assignments, Laboratory tests, Seminar

External assessment: End Semester Examination

COURSE CODE**: CSC4046** COURSE NAME**: ARTIFICIAL INTELLEGENCE** COURSE TYPE**: ELECTIVE/OPEN** NUMBER OF CREDITS**: 6**	L-T-P**: 4-1-1** CONTACT HOURS/WEEK**: 7** TOTAL MARKS**: 100** (INTERNAL**: 60,** EXTERNAL**: 40**) NATURE**: GRADED**

COURSE OBJECTIVES:

1. The objective of the course is to present an overview of artificial intelligence (AI) principles and approaches
2. Develop a basic understanding of the building blocks of AI in terms of intelligent agents: Search, Knowledge representation, inference, logic, and learning.

COURSE PREREQUISITE:

- Discrete Mathematics, basics of C/C++ programming

COURSE OUTCOMES:

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

- Analyze important historical and current trends addressing artificial intelligence.
- Identify forward and backward recovery techniques.
- Implement logic programming concepts in AI.
- Explain the components of expert system

COURSE CONTENT:

Unit No & Name	Components of the Unit	No of contact hours	Marks
UNIT-I: **Historical foundation of AI**	• Historical foundation of AI. AI application areas. AI problem, Underlying assumptions, AI techniques, Level of models, success criteria. Problem as a state space search, Production Systems, Problem characteristics, PS characteristics, Design issues of search programs.	20	15
UNIT-II: **Heuristic Search Techniques**	• Generate and test, Hill Climbing, Best-First Search, Problem reduction	15	15
UNIT-III: **Knowledge representation and Mapping**	• Approaches, Issues. Predicate logic. Representing simple facts in logic, Instance and isa relationship, Computable function and predicity, Resolution, Natural Deduction.	15	20
UNIT-IV: **Knowledge representation using rules**	• Procedural vs declarative, logic programmes, Forward vs backward recovery, matching.	20	25

	Nonmonotonic reasoning and logic. Implementation: Depth first abd breath first search.		
UNIT-V: **Knowledge representation using rules**	• Introduction to statistical reasoning. Probability and bays theorem, Fuzzy logic concept. Concept of weak slot and filter, and strong slot and filter structure. Fundamental of Natural Language Processing: Syntactic processing, semantic analysis. Concept of Expert Systems: Representation using domain knowledge, Expert System shell, knowledge acquisition.	20	25
	Total:	90	100

TEXTBOOKS/ RECOMMENDED READINGS:

- Artificial Intelligence : E. Rich & K. Knight : Tata McGraw Hill.
- Artificial Intelligence: Structures and Strategies for Complex Problem solving: George Luger, Pearson Education.
- Principles of Artificial Intelligence: Nils J Nisson: Narosa

COURSE ASSESSMENT DETAILS:

Internal assessment: Class tests, Assignments, Laboratory tests, Seminar

External assessment: End Semester Examination

COURSE CODE**: CSC4056**	L-T-P**: 4-0-2**
COURSE NAME**: SPEECH PROCESSING**	CONTACT HOURS/WEEK**: 8**
COURSE TYPE**: ELECTIVE/OPEN**	TOTAL MARKS**: 100** (INTERNAL**: 60,** EXTERNAL**: 40**)
NUMBER OF CREDITS**: 6**	NATURE**: GRADED**

COURSE OBJECTIVES:

1. To provide students the concepts of basic models for speech production
2. To familiarize students to develop time and frequency domain techniques for estimating speech parameters
3. To introduce students a predictive technique for speech compression
4. To provide students the process of speech recognition, synthesis and speaker identification.

COURSE PREREQUISITE:

- Programming experience in C, C++

COURSE OUTCOMES:

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

- Explain basic characteristics of speech signal in relation to production and hearing of speech by humans.
- Analyze different algorithms of speech analysis common to many applications.
- Solve practical aspects of speech algorithm's implementation.
- Design a simple system for speech processing (speech activity detector, recognizer of limited number of isolated words), including its implementation into application programs.

COURSE CONTENT:

Unit No & Name	Components of the Unit	No of contact hours	Marks
UNIT-I: **Digital Signal Processing**	• Introduction: signals, systems and signal Processing, Frequency in Continuous Time & Discrete Time Signals. Analog to Digital & Digital to Analog Conversion. Discrete Time Signals & Systems: Discrete Time Signals, Discrete Time Systems, Discrete Time Systems described by Difference equations, Correlation of Discrete Time Signals.	12	15
UNIT-II: **Fundamentals of speech signal**	• History of speech recognition research, The Speech Signal: Speech production mechanism, Classification of speech, sounds, nature of speech signal, models of speech production. • Speech signal processing: purpose of speech processing, digital models for speech signal, Digital processing of speech signals, Significance, short time analysis.	12	15

UNIT-III: **Time domain methods for speech processing**	• Time domain parameters of speech, methods for extracting the parameters, Zero crossings, Auto correlation function, pitch estimation.	12	12
UNIT-IV: **Frequency domain methods for speech processing**	• Short time Fourier analysis, filter bank analysis, spectrographic analysis, Formant extraction, pitch extraction, Analysis - synthesis systems. Homomorphic Signal Processing	12	12
UNIT-V: **Linear predictive coding of speech**	• Formulation of linear prediction problem in time domain, solution of normal equations, Interpretation of linear prediction in auto correlation and spectral domains.	12	12
UNIT-VI: **Speech analysis**	• Cepstral analysis of speech, formant and pitch estimation, • Mel frequency cepstrum computation, Applications of speech processing - Speech recognition, Speech synthesis and speaker verification.	10	12
UNIT-VII: **Automatic speech recognition**	• Basic pattern recognition approaches, Parametric representation of speech, Evaluating the similarity of speech patterns, • Isolated digit Recognition System, Continuous digit Recognition System. Vector quantization, speech coding	10	12
UNIT-VIII: **HIDDEN MARKOV model for speech recognition**	• Introduction to Hidden Markov Model (HMM), • Types of HMM, Hidden Markov Model (HMM) for speech recognition, • Viterbi algorithm, Training and testing using HMMs, Adapting to variability in speech (DTW), Language models. Example of speech recognition project.	10	10
	Total:	90	100

TEXTBOOKS/ RECOMMENDED READINGS:

- L. Rabiner and B.-H. Juang, Fundamentals of Speech Recognition, Prentice Hall, 1995, ISBN 0-13-015157-2
- L. R. Rabiner and R. W. Schafer, Digital Processing of Speech Signals, Prentice-Hall, 1978, ISBN 0-13-213603-1.
- J.L Flanagan : Speech Analysis Synthesis and Perception - 2nd Edition - Sprenger Vertag, 1972.

- I.H.Witten : Principles of Computer Speech , Academic press, 1983.
- Speech Communications: Human & Machine - Douglas O'Shaughnessy, 2nd ed., IEEE Press.
- Discrete Time Speech Signal Processing: Principles and Practice - Thomas F. Quateri 1st ed., PE.
- Speech & Audio Signal Processing- Ben Gold & Nelson Morgan, 1 ed., Wiley.
- Speech Recognition - Claudio Becchetti and Lucio Prina Ricotti, Wiley

COURSE ASSESSMENT DETAILS:

Internal assessment: Class tests, Assignments, Laboratory tests, Seminar

External assessment: End Semester Examination

COURSE CODE**: CSC4066** COURSE NAME**: APPLIED GRAPH THEORY AND ALGORITHMS** COURSE TYPE**: ELECTIVE** NUMBER OF CREDITS**: 6**	L-T-P**: 4-2-0** CONTACT HOURS/WEEK**: 6** TOTAL MARKS**: 100** (INTERNAL**: 60,** EXTERNAL**: 40**) NATURE**: GRADED**

COURSE OBJECTIVES:

1. To introduce the students with a number of real life applications that can be treated and solved as graph theoretic problems.
2. To introduce the students with algorithms for graph theoretic problems.
3. To acquaint the students with domain dependent different representations of graphs.

COURSE PREREQUISITE:

- Basic knowledge of Programming, Discrete Mathematics and Graph Theory.

COURSE OUTCOMES:

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

- Analyze different shortest path problems
- Implement different graph matching problems
- Solve different graph colouring algorithms
- Explain different modeling of physical networks

COURSE CONTENT:

Unit No & Name	Components of the Unit	No of contact hours	Marks
UNIT-I: **Shortest path problems (SP)**	• Various versions of the SP problem. • Algorithms for single source SP problem. • Characterization and presence of SP, SP tree • Ford's labeling method and its correctness • Labeling and Scanning method - efficient scanning orders. Topological order for a cyclic networks. • Shortest-first search for non-negative network (Dijkstra), BFS search for several networks and its analysis, All-pair shortest path problem - Floyd's algorithm and its analysis.	25	20
UNIT-II: **Flows in Networks**	• Basic concepts, Max flow-min cut Theorem. • Ford and Fulkerson's augmenting path method. • The Edmonds-Karp algorithm to solve the maximum flow problem. • Integrality theorem - Maximum capacity	20	20

	augmentation and its analysis - Augmentation by blocking flows - Dinic's algorithm-analysis of number of blocking steps for general and unit networks.		
UNIT-III **Matching Problems**	• Basic concepts. Bipartite matching and network flows. Hall's marriage theorem. • Non-bipartite matching-basic concepts, Edmonds- Blossom shrinking algorithm and its analysis.	10	15
UNIT-IV **Planarity and Graph Isomorphism**	• Review of basic results about planarity. • Kuratowski's theorem • Polynomial algorithm for testing of planarity and applications. • Graph Isomorphism and its importance. Backtracking algorithm for general graphs. Isomorphism problem and its complexity. Isomorphism complete problems, polynomial time algorithm for planar graph isomorphism problem, Group theoretic methods and graph isomorphism problem.	15	20
UNIT-V **Graph Coloring**.	• Map and vertex coloring problem. • 6,5 and 4-colour theorems for planar graphs, coloring graphs on compact surfaces, chromatic number.	10	12
UNIT-VI **Physical Networks**	• Modeling physical networks, component equations. • Kirchoff's laws, dual networks. Fundamental cycle and cutest equations. • Matrix form of the network equations, state equations	10	13
	Total	90	100

TEXTBOOKS/ RECOMMENDED READINGS:

- Chartrand. G. and Ollermann.O.R, (1993), *Applied and Algorithmic Graph Theory*, Mc-Graw Hill.
- Tarjan T.E., (1983), *Data structures and Network Algorithms*, Siam.Society for Industrial and Applied Mathematics.
- Horowitz E. and Sahani S., (2nd Edition), *Fundamentals of Computer Algorithms*, Galgotia.
- Deo N., *Graph Theory with Applications to Engineering and Computer Science*, PHI.

COURSE ASSESSMENT DETAILS:

Internal assessment: Two/ Three mid semester examinations will be conducted. 60% of this evaluation will be added to the total marks for this course.

External assessment: End Semester Examination will be of 100 marks covering the entire course and the exam duration will be 3 hours. 40% of the mark obtained will be added to the total marks for this course.

COURSE CODE**: CSC4076** COURSE NAME**: SYSTEM ADMINISTRATION AND NETWORKING** COURSE TYPE**: ELECTIVE** NUMBER OF CREDITS**: 6**	L-T-P**: 4-2-0** CONTACT HOURS/WEEK**: 6** TOTAL MARKS**: 100** (INTERNAL**: 60,** EXTERNAL**: 40**) NATURE**: GRADED**

COURSE OBJECTIVES:

1. To teach the basics of Linux operating system, such as file handling, different file types, file ownership, process concepts, Linux kernel etc.
2. To teach system handling using shell programs and Linux commands.
3. To provide different ways of client-server communication, such as NFS, NIS, telnet, ssh, ftp etc.
4. To introduce the basics of internet such as, different classes of IP addresses, DNS etc.
5. To discuss recent network security issues and the remedies for those issues.

COURSE PREREQUISITE:

- Basics of OS, any programming language.

COURSE OUTCOMES:

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

- Explain the basic concepts of Linux OS such as file system, file hierarchy, processes, distributions, disk partitions
- Perform various basic commands in Linux for setting user and group ownerships of files and directories, access permissions; commands related to process, system monitoring and logging, file system handling and network configurations.
- Configure network systems, the resolver library to arrange TCP/IP services, set up IP-address, network masks, configure inet daemon etc.
- Explain the use and configure DNS, NFS, NIS, telnet, send mail etc.

COURSE CONTENT:

Unit No & Name	Components of the Unit	No of contact hours	Marks
UNIT-I: **Introduction on Linux File System**	• Major components of the Linux operating systems. • File system, setting user and group ownership of files and directories and access permissions. • Basic commands for starting and stopping processes. • Basic process attributes and their role in access control.	12	15

	• Mounting and un-mounting file systems and partitions.		
UNIT-II: **Linux Kernel**	• Linux kernel program, starting and stopping a Linux system. • Setting up user and group accounts on single machines. • The basics of backup and restore procedures.	12	15
UNIT-III**: Process management in Linux**	• Linux system monitoring and logging. • Examining the list of running processes on the system and understand the data presented there. • Monitoring memory usage and disk space usage on the system. • Customizing system log configuration.	8	10
UNIT-IV: **Basics of Networking**	• The rules of governing IP address classes and netmasks. • Configuring the resolver library to arrange for TCP/IP name service • Bringing interfaces up and down, and set their IP addresses and netmasks • Setting the default route in the kernel routing table. • Understanding the significance of the /etc/services file and well-known port numbers, • Configuring the inet daemon, Using telnet to contact servers directly, using the ping command to test network connectivity, • netstat command to examine kernel tables pertaining to networking, • traceroute command to discover network paths, tcpdump to examine all network traffic. • Methods used to bring interfaces up and down.	18	20
UNIT-V: **Network Configuration**	• Basics of configuring and using the Domain Name Service, sendmail • The Network Information System, Network File System: Structure and function of the Domain Name Service (DNS) • Setting up a Linux machine to function as a DNS server, Configuring and using sendmail, • Setting up an NIS domain with an NIS master server and NIS clients.	15	15
UNIT-VI: **Network Security**	• Basic network security issues and solutions.	5	7
UNIT-VII: **Configuration of NFS Client/Server**	• Setting up a Linux machine to act as an NFS server	12	12

	• Setting up a Linux machine to act as an NFS client		
UNIT-VIII: **Backup in Linux**	• Incremental back up. Monthly back. Mail server setup	8	6
	Total:	90	100

TEXTBOOKS/ RECOMMENDED READINGS:

- Proffitt B., Red Hat Linux, PHI
- Richard Stevens, W., UNIX Network Programming- Vol-I and Vol-II, PHI
- IBM series , Introduction to System Administration, PHI

COURSE ASSESSMENT DETAILS:

Internal assessment: Class tests, Assignments, Laboratory tests, Seminar

External assessment: End Semester Examination

COURSE CODE**: CSC4086** COURSE NAME**: WIRELESS COMMUNICATION AND NETWORKS** COURSE TYPE**: ELECTIVE** NUMBER OF CREDITS**: 6**	L-T-P**: 4-2-0** CONTACT HOURS/WEEK**: 6** TOTAL MARKS**: 100** (INTERNAL**: 60,** EXTERNAL**: 40**) NATURE**: GRADED**

COURSE OBJECTIVES:

1. To provide students the concepts of basic components of wireless communication systems
2. To give the familiarity of multiple access techniques of wireless communication technologies
3. To enable students the implement different wireless communication protocols

COURSE PREREQUISITE:

- Concepts of Data Communication and Computer Networks (CSC2016)

COURSE OUTCOMES:

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

- Distinguish different modulation techniques such AM, FM, BPSK, QPSK, QAM OFDM, FHSS, DSSS
- Explain the IEEE 802 protocol Architecture
- Analyze the concept of Mobility Management - handoff and location management
- Explain different multiple access techniques for wireless communication such as FDMA, TDMA etc.

COURSE CONTENT:

Unit No & Name	Components of the Unit	No of contact hours	Marks
UNIT-I: **WIRELESS COMMUNICATIONS AND SYSTEM FUNDAMENTALS**	• Introduction to wireless communications systems, examples, comparisons & trends. • Cellular concepts-frequency reuse, Cell splitting sectoring, repeaters, microcell concepts, strategies, interference & system capacity. • Modulation techniques: AM, FM, BPSK, QPSK, QAM OFDM, FHSS, and DSSS. • Spectrum allocation policy and scarcity of radio spectrum, capacity of cellular systems. Channel Assignment problem, Mobility Management - handoff and location	12	13

	management. Software Defined Radio and Cognitive Radio.		
UNIT-II: **MULTIPLE ACCESS TECHNIQUES FOR WIRELESS COMMUNICATION**	• FDMA, TDMA, SSMA (FHMA/CDMA/Hybrid techniques), SDMA technique (as applicable to wireless communications). • Packet radio access-protocols, CSMA protocols, reservation protocols, capture effect in packet radio	16	18
UNIT-III: **WIRELESS NETWORKING**	• Introduction, differences in wireless & fixed telephone networks, traffic routing in wireless networks –circuit switching, packet switching X.25 protocol. • Wireless data services – cellular digital packet data (CDPD), advanced radio data information systems, RAM mobile data (RMD). Common channel signaling (CCS),ISDN-Broad band ISDN & ATM, Signaling System no .7(SS7)-protocols, network services part, user part, signaling traffic, services and • performance	18	20
UNIT-IV: **WIRELESS LAN TECHNOLOGY**	• Infrared LANs, Spread spectrum LANs, Narrow bank microwave LANs, IEEE 802 protocol Architecture, IEEE802 architecture and services, 802.11 medium access control, 802.11 physical layer. 802.16 WiMAX. • Other wireless technologies: GSM, GPRS, 2.5G, 3G, 4G, WLL	16	18
UNIT-V: **802.11**	• 802.11 MAC, DCF and PCF. 802.11 framing in detail, WEP, Authentication, 802.1 management operations 802.11 in linux (nl80211). Handover-intra and inter BSS. QoS.	16	18
UNIT-VI: **MOBILE IP AND WIRELESS APPLICATION PROTOCOL**	• Mobile IP Operation of mobile IP, Co-located address, Registration, Tunneling • WAP Architecture, overview, WML scripts, WAP service, WAP session protocol, wireless transaction, Wireless datagram protocol.	12	13
	Total:	90	100

TEXTBOOKS/ RECOMMENDED READINGS:

- Theodore, Rappaport S.; *Wireless Communications, Principles*, *Practice*;
- Matthew S Gast; *802.11 Wireless Networks*;
- Stallings W.; *Wireless Communication and Networking*;
- Feher K. ; Wireless Digital Communications;
- Tse D. & Vishwanath P. ; *Fundamentals of Wireless Communication*; Cambridge University Press

COURSE ASSESSMENT DETAILS:

Internal assessment: Class tests, Assignments, Lab tests

External assessment: End Semester Examination

COURSE CODE**: CSC4096** COURSE NAME**: QUEUING THEORY AND OPERATIONS RESEARCH** COURSE TYPE**: ELECTIVE** NUMBER OF CREDITS**: 6**	L-T-P**: 4-1-1** CONTACT HOURS/WEEK**: 7** TOTAL MARKS**: 100** (INTERNAL**: 60,** EXTERNAL**: 40**) NATURE**: GRADED**

COURSE OBJECTIVES:

1. To provide students with basic skills and knowledge of Queing Theory and operations research and its application to find solution for a problem
2. To understand mathematical models used in Operations Research and its uses.

COURSE PREREQUISITE:

- Discrete Mathematics, Graph Theory (CSC4066), basic programming knowledge

COURSE OUTCOMES:

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

- Analyze linear programming techniques using Charma's method of penalties and the two-phase algorithm
- Distinguish Revised simplex method and the simplex method
- Solve the problem of reduction of the game problem into a linear programming problem
- Illustrate dynamic programming and integer programming

COURSE CONTENT:

Unit No & Name	Components of the Unit	No of contact hours	Marks
UNIT-I: **Linear Programming Techniques**	• The simplex algorithm, Charma's method of penalties, the two-phase algorithm, problems of degeneracy and cycling.	8	10
UNIT-II: **Duality in Linear Programming**	• The duality theorem, Revised simplex algorithm. Revised simplex method versus the simplex method. • Sensitivity Analysis, changes in the requirement vector, the cost vector and the coefficient matrix.	12	14
UNIT-III: **Parametric Programming**	• parametrization of the cost vector of the requirement vector. • Theory of Games: 2-person zero sum game; Reduction of the game problem into a linear programming problem.	12	14

	• The Transportation problem: various algorithms such as the algorithm of stepping stones. Vogel's method.		
UNIT-IV: **Non-Linear Programming**	• constrained minima and maxima, Necessary and sufficient condition for maxima and minima; The Kuhn – Tucker principle; Quadratic Programming. • Queuing Theory: the Exponential Distribution; queue disciplines much as M/M/1, M/M/C, M/EK/1, M/G/1, etc.	12	12
UNIT-V: **Simulation**	• Event type simulation. Monte Carl'o Techniques, simulation techniques applied to queues	12	12
UNIT-VI: **Dynamic Programming**	• the recursion approach, computation procedures using Calculus • Geometric Programming; Generalization using Kuhn-Tucker Principle.	12	12
UNIT-VII: **Integer Programming**	• Integer linear programming in 2 dimensions, General ILP and MILP problems	12	12
UNIT-VIII: **Selected hard optimization problems**	• Review of NP hardness, TSP, Max clique problem, Multiprocessor scheduling problem.	10	12
	Total:	90	100

TEXTBOOKS/ RECOMMENDED READINGS:

- Linear Programming – S.L. Gass
- Optimization methods – K.V Mittal and G.Mohan
- Stochastic Processes – J. Medhi
- Operations Research – K.Swarup , P.K. Gupta and M. Mohan.
- Operations Research – by H. Taha.

COURSE ASSESSMENT DETAILS:

Internal assessment: Class tests, Assignments, Lab tests

External assessment: End Semester Examination